

Ivica Kartelo

Visual Basic i ASP



ŠKOLA E-92 SPLIT
Split 2003.

Ružici, Ivani, Martini.

Autor:
Ivica Kartelo, dipl. ing.

Naslov:
Visual Basic, ASP, Web aplikacije

Urednik:
Ivica Kartelo

Nakladnik:
Škola E-92 Split
Mosečka 105, 21000 Split

Fax/Tel.: 021 569-777, 021 569-333, E-mail: skola-e-92@st.tel.hr
web: <http://www.e92.hr>

Za nakladnika:
Ružica Kartelo-Marušić, dipl. iur.

Tisak:
Kartular Split

Grafičko uređenje korica
ArtMobile Split

Priprema sloga za tisak i grafičko uređenje:
Ivica Kartelo

Lektor:
Ivan J. Bošković, prof.

U ovoj knjizi pojavljuje se nekoliko zaštićenih znakova i naziva čija je uporaba ograničena zakonom. Popis tih znakova i naziva kao i njihovih vlasnika nalazi se kod nakladnika.

© Ivica Kartelo

I. izdanje / Naklada: digitalni tisak.

Split, rujan 2003.

Sva imena firma, osoba, proizvoda i usluga koja se pojavljuju u knjizi i u primjerima su izmišljena.

Autor i Nakladnik ne snose odgovornost za posljedice korištenja ove knjige, niti za možebitne pogreške nastale u procesu njezina stvaranja.

ISBN 953 - 6245 - 41-8

Zahvala

Zahvaljujem se dipl. ing. Ivanu Jurasu, mr. ecc. Mirku Vukušiću i prof. Siniši Paroviću iz Netmedije na svesrdnoj profesionalnoj pomoći u programiranju.

Predgovor

U jednoj od svojih knjiga Bill Gates kaže kako se u predviđanju budućih događaja najbolje držati ovog pravila: ako entuzijast kaže da će za prihvaćanje nekog projekta trebati pet godina, pomnožite to s dva, a ono što kaže pesimist, podijelite s dva i blizu ste istine.

Ja spadam u entuzijaste, pa nije čudno što sam učenje na daljinu pomoću Interneta pokrenuo još 1996. godine, a prvih 50 učenika imao tek 2003. godine. Zato, kada spomenem neke godine u budućnosti, pomnožite to s dva!

Poslovanje preko Interneta i kod nas će imati veliku ulogu, i to vrlo brzo. Prije svega zato što imamo jedan veliki svjetski izvozni *brand*: turizam. Svima koji se bave turizmom Internet nudi gotovo nevjerojatno smanjenje troškova poslovanja. Automatizacija rezervacija i plaćanja svih mogućih usluga, od kreveta do raftinga i istraživanja na biciklima. To je dosadašnjim "na ruke" praćenjem naprosto neisplativo, jer treba više uposlenih nego gostiju.

Ne povećava se dobit samo direktnom prodajom preko Interneta. I ostale firme kod nas, od najmanje do najveće, i s minimalnim investicijama u Intranet smanjit će troškove poslovanja i tako povećati dobit. Do jučer nam je bilo najvažnije imati web stranice na Web-u. Danas moramo imati i podatke i obavljati transakcije (manipulirati podacima u bazi podataka).

Grafičkim web dizajnerima, web kreatorima i autorima sadržaja pridružili su se i Internet programeri. Došli su posljednji, a u sljedećim bi godinama mogli biti najbrojnija radna snaga na Internetu. Ova je knjiga namijenjena svima koji se premišljaju ući ili ne ući još i na ovo područje. U knjizi su i cjelovite profesionalne web aplikacije na kojima ćete mnogo naučiti i... jabuka je zagrižena. Dao sam sve od sebe. Knjiga pruža veliki panoramski pogled na temu, ali je istovremeno obuhvatila i podosta detalja. Siguran sam, nakon ove knjige imat ćete jasnu sliku o zanimanju Internet programer. Steći ćete programersko iskustvo u izgradnji dviju najčešćih web aplikacija danas. To su web aplikacije E-trgovina i On-line učilište.

U Splitu, 4. rujna 2003.

Ivica Kartelo

Sadržaj

<i>Zahvala</i>	4
Predgovor	5
Sadržaj	6
Uvod	13
<i>Cilj knjige i gdje skinuti programski kôd</i>	13
<i>Kome je namijenjena ova knjiga</i>	13
<i>Što ćete naučiti u ovoj knjizi</i>	13
<i>Što nema u ovoj knjizi</i>	14
<i>Koji mi je software potreban za rad po ovoj knjizi</i>	14
ASP	15
Zaključak.....	16
Microsoft Access.....	16
SQL Server.....	18
<i>Gdje je zapinjalo</i>	18
Tutorial Windows XP Professional IIS 5.1	18
Primjer 1 iz Microsoftovog Tutoriala: zašto u VBScriptu moramo pisati brojeve ugrađenih konstanti, a u Visual Basic-u to ne moramo.....	18
Instalacija IIS 5.1 na Windows XP Professional.....	18
Pokretanje IIS.....	18
Kreiranje virtualne mape.....	19
Primjeri ASP stranica se nalaze u iissamples.....	20
Primjer 2 iz Microsoftovog Tutoriala: zašto primjer s bazom Authors.mdb radi, a primjer s bazom guestbook.mdb ne radi	24
Lekcija 1 ASP počnimo s primjerom	27
<i>Izrada komponente</i>	27
DLL komponenta	28
Registracija DLL komponente	29

Izmjene u komponenti.....	29
Izrada HTML dokumenta.....	29
Izrada ASP stranice.....	30
Fizička mapa i virtualna mapa	32
Windows NT 4 i Option Pack 4	32
Windows XP i IIS 5,1	32
A sada, provjerimo u radu HTML, ASP i komponentu	33
Windows NT 4 i Option Pack 4	33
Windows XP i IIS 5,1	34
Analiza.....	35
Moja osobna učinkovitost do ljeta 2003.	41
Lekcija 2 ASP	43
Panoramski pogled na ASP.....	43
ActiveX.....	44
COM.....	45
IIS.....	46
World Wide Web	47
Inetinfo.exe.....	48
ISAPI (Internet Server Application Programming Interface).....	48
Active Server Pages	48
Koji thread imaju naše COM komponente.....	50
ASP.....	52
Izrada ASP stranice	53
Programiranje pomoću skripta Procedura Function.....	54
Analiza	56
Lokalizacija	57
Lekcija 3 – Ugrađeni ASP objekti	60
ASP direktive.....	60
Web aplikacija.....	61
Built-in ASP Objects	63
Objekt Application.....	63
Objekt Request.....	63

Objekt Response	64
Objekt Server	64
Objekt Session.....	64
Objekt ObjectContext	65
Objekt ASPError	65
<i>Primjeri uporabe ugrađenih objekata.....</i>	<i>65</i>
Primjer 1: Metoda Server.HTMLEncode.....	65
Primjer 2: Metoda Get i kolekcija Request.QueryString	66
Analiza ASP stranice QueryString_VBScript .asp.....	66
Primjer 3: Metoda Post i kolekcija Request.Form	68
Analiza	68
Primjer 4: Properties Session.SessionID.....	69
Analiza	69
Primjer 5: Kolekcije Request.Cookies i Response.Cookies	70
Analiza	71
Lekcija 4 COM komponente.....	74
<i>Granularity komponente</i>	<i>74</i>
<i>Kreiranje instancije objekata koji postoje u komponentama.....</i>	<i>75</i>
<i>Ugrađeni ASP objekti.....</i>	<i>76</i>
<i>Život (scope) objekta</i>	<i>76</i>
<i>Objekt koji živi koliko i jedna posjeta (session).....</i>	<i>77</i>
<i>Objekt koji živi koliko i jedna aplikacija.....</i>	<i>78</i>
Lekcija 5 Varijable.....	80
<i>Varijable životnog vijeka Session i Application.....</i>	<i>81</i>
Session	81
Application.....	83
Konstante	84
Prijenos Arrays-a u proceduru	85
Rad s kolekcijama	85
Razlika između VB i ASP kolekcija	87
Cookie	87
Lekcija 6 ADO – Knjiga gostiju.....	88

<i>Spajanje na izvor podataka.....</i>	88
Analiza ASP stranice <i>guestbook1.asp.....</i>	88
Gledanje poruka u knjizi gostiju. Analiza ASP stranice <i>GuestBookList1.asp.....</i>	94
Lekcija 7 ADO – Unos, brisanje, editiranje, ažuriranje.....	97
Analiza stranice <i>radbazom.asp.....</i>	97
<i>Execute metoda objekta Connection</i>	110
<i>ADO objekt Recordset.....</i>	111
<i>ADO objekt Command</i>	113
Lekcija 8 Transaction	115
<i>Kratka povijest aplikacija klijent – poslužitelj.....</i>	115
TCP/IP.....	117
COM.....	118
Web aplikacija u tri sloja.....	119
<i>Vratimo se transakciji</i>	121
ObjectContext	122
Lekcija 9 Profesionalni primjer E-knjižara Kartelo ASP stranice	125
<i>ekartelo.asp</i>	125
<i>registracija.asp.....</i>	138
<i>provjeraregistracije.asp.....</i>	148
<i>PrikaziKnjige.asp.....</i>	149
<i>pogleduKosaricu.asp.....</i>	153
<i>OdabirKategorija.asp</i>	157
<i>Global.asa</i>	159
<i>glavni.css.....</i>	159
<i>error.inc</i>	161
<i>dodajKnjigu.asp</i>	161
<i>DetaljiNarudzbe.asp.....</i>	163
<i>azurirajKosaricu.asp.....</i>	169

<i>abort.asp</i>	171
Lekcija 10 Profesionalni primjer E-knjižara Kartelo COM Visual Basic 6 komponente	173
<i>modZajednicki (modZajednicki.bas)</i>	173
<i>eKarteloASPPomocna.dll</i>	178
Class modul (objekt): NovaKolekcija (NovaKolekcija.cls).....	178
<i>eKarteloSQLKupci.dll</i>	180
Class modul (objekt): SQLKupci (SQLKupci.cls)	180
<i>eKarteloASPKupci.dll</i>	188
Class modul (objekt): Kupci (Kupci.css).....	188
<i>eKarteloSQLKatalog.dll</i>	193
Class modul (objekt): SQLKatalog (SQLKatalog.cls).....	193
<i>eKarteloASPKatalog.dll</i>	199
Class modul Katalog (Katalog.cls)	199
<i>eKarteloSQLKosarice.dll</i>	204
Class modul (objekt): SQLKosarice (SQLKosarice.cls).....	204
<i>eKarteloASPKosarice.dll</i>	215
Class modul (objekt) Kosarice (Kosarice.cls).....	215
<i>eKarteloASPObavljanjeNarudzbe.dll</i>	222
Class modul (objekt): Narudzba (Narudzba.cls).....	222
Lekcija 11 E-knjižara Kartelo u radu	228
<i>Od čega se sastoji e-knjižara</i>	228
<i>Koji sam software koristio za izradu i rad</i>	229
<i>e-knjižare</i>	229
<i>Prezentacijski, biznis i podatkovni slojevi web aplikacije e-knjižara</i>	229
<i>Instalacija e-knjižare</i>	230
Download	230
IIS 5.1	231
Svojstva web aplikacije.....	232
Component Services(IIS 5.1 Windows XP Professionl).....	233
Prijava baze u DSN	235
Provjerite e-knjižaru Kartelo.....	236

Greške i kako ih ukloniti	241
Zadatak za vježbu.....	242
Lekcija 12 E-knjižara Kartelo analiza ASP kôda	244
<i>index.asp</i>	244
<i>ekartelo.asp</i>	244
Poziv komponente	248
<i>Nastavimo s analizom kôda na ASP stranici ekartelo.asp</i>	257
<i>registracija.asp</i>	261
Lekcija 13 Analiza kôda komponenti.....	265
<i>Class forma Kupci</i>	265
Metoda <i>ProvjeriKupca</i>	268
Metoda <i>eKarteloSQLKupci.SQLKupci.OdaberiKupca</i>	270
<i>Vraćamo se metodi eKarteloASPKupci.Kupci.ProvjeriKupca točno tamo gdje smo je napustili</i>	275
Najprije analiza mogućih vraćenih podataka	275
Nastavljamo s kôdom metode <i>Provjeri kupca</i>	278
Evo nas konačno nazad u ASP kodu stranice <i>ekartelo.asp</i>	281
<i>Sažetak analize web aplikacije E-knjižara Kartelo</i>	283
Lekcija 14 Virtualno učilište Kartelo	287
<i>Instalacija aplikacije</i>	287
<i>Prijavite Access bazu u DSN</i>	289
<i>Pokrenite eLearning Kartelo</i>	290
<i>index.asp</i>	291
<i>Global.asa</i>	296
<i>izvjesce.asp</i>	297
<i>Kreniiodustani.asp</i>	301
<i>novi_ucenik.asp</i>	308
<i>promjena_podataka.asp</i>	315
<i>putlekcija.asp</i>	323

<i>razred.asp</i>	324
<i>razredi.asp</i>	339
<i>test.asp</i>	343
<i>testrezultat.asp</i>	350
<i>ucenik_izbornik.asp</i>	358
<i>zadatak.asp</i>	363
Literatura	369

Uvod

Cilj knjige i gdje skinuti programski kôd

Ovom sam knjigom želio dati poticaj svima kojima se po glavi vrti ideja o programiranju za Web. Jedini pravi način uvida u cjelovitu problematiku je analiza i učenje na živim profesionalnim web aplikacijama: E-trgovinai i Online učilištu.

Sav programski kôd možete skinuti na URL: www.e92.hr/knjigaasp.zip

Kome je namijenjena ova knjiga

Ova se knjiga rađa četiri godine, pa nije čudno što pokriva tehnologiju od NT 4 i IIS 4, preko Windowsa 2000 i IIS 5.0 do Windowsa XP i IIS 5.1. Cijelo vrijeme rada na knjizi red očima sam imao svoje učenike iz zanimanja Internet programer, koje držim preko Interneta u ustanovi Kartelo.

Do ove knjige moji su učenici "prešli" sljedeće moje knjige: Access, Visual Basic 6, Visual Basic baze, Visual Basic objekti.

Što ćete naučiti u ovoj knjizi

U ovoj ćete knjizi naučiti:

1. Administrirati Internet Information Services (IIS) za web aplikacije.
2. Administrirati Component Services (Microsoft Transaction Server, COM+) za web aplikacije.
3. Component Object Model (COM) okruženje.
4. Active Server Components.

5. Manipulirati s bazom podataka iz bilo kojeg Web browser-a.
6. Praviti komponente i registrirati ih na operacijskom sustavu Windows.
7. Napraviti cjelovitu profesionalnu web aplikaciju e-knjižara.
8. Napraviti web aplikaciju u tri sloja: prezentacijskom, biznis sloju i sloju izvora podataka.
9. Zašto je najbolje rješenje koristiti komponente za srednji sloj ili biznis logiku.
10. Kako napraviti komponente i ASP stranice koje će raditi u COM i Microsoft Transaction Server okruženju.
11. Koristiti ugrađene ASP objekte.
12. Koristiti ActiveX Data Objects (ADO) i ASP komponentu za rad s bazama podataka.
13. Koristiti VBScript i Visual Basic za manipulaciju varijablama, objektima i posebno kolekcijama.

Što nema u ovoj knjizi

U knjizi nema referentnih tablica s popisom i parametrima svih ugrađenih ASP objekata, kolekcija, metoda, properties-a. Nema referentnih tablica i objašnjenja svih ključnih riječi VBScripta i VisualBasic-a.

Koji mi je software potreban za rad po ovoj knjizi

Za učenje, razvoj i minimalne potrebe manje firme nije vam potreban skupi poslužiteljski software.

Za mnoge primjere već vjerojatno imate sve potrebno na svom računalu, makar ste još na Windowsima 95, Windows 2000 Professional ili Windows XP Professional.

Sve praktične web aplikacije, koje ćemo raditi u ovoj knjizi, mogu se podijeliti u tri sljedeće kategorije:

Potreban samo ASP (klijent verzije IIS-a su već na vašem računalu ili na CD-ROM-u operacijskog sustava).

Potreban ASP i bilo koja verzija Visual Basic-a u kojoj možemo kreirati ActiveX DLL.

Potreban ASP, Visual Basic i Microsoft Access (koji najvjerojatnije već imate na svom računalu).

ASP

Kad kažemo da je na svom računalu potrebno imati ASP, to znači da je potrebno imati Microsoft Internet Information Server, s kojim u naše računalo dolazi i ASP. Evo više detalja o verzijama IIS.

IIS verzija	Operacijski sustav
IIS 4	Windows NT 4.0
Personal Web Server	Windows NT 4.0 Workstation, Windows 95
IIS 5	Windows 2000 Server sve verzije
IIS 5 (klijent verzija)	Windows 2000 Workstation
IIS 5.1	Windows XP Professional
IIS 6	Windows Server 2003 (sve verzije)

Za gornje verzije IIS vrijede sljedeća pravila:

1. Svaka verzija IIS namijenjena je točno određenom operacijskom sustavu. To znači da se IIS 6 ne može instalirati na Windows XP ili 2000, niti IIS 5 može biti instaliran na NT 4.
2. Navedeno je posljedica tijesne integriranosti IIS-a u specifični operacijski sustav.

3. IIS 5.1 nije podržan niti isporučen na Windows XP Home Edition. IIS 5.1 se ne može skinuti s Interneta, kupiti posebno ili naručiti na CD-ROM-u. IIS 5.1 je isporučen i podržan na Windows XP Professional, a ugrađuje se preko Control Panel/Add/Remove Programs/Add Remove Windows Components. IIS 5.1 je klijent verzija namijenjena za učenje i razvoj, a u rijetkim slučajevima i za vrlo mali Intranet. Ima ugrađena ograničenja i nije projektiran za raditi kao poslužitelj.
4. Klijent verzije IIS su PWS, IIS 5.1 i IIS 5.0 na Windows 2000 Professional. Ove su verzije namijenjene za učenje i razvoj, i ograničene su na podršku samo jednom web site-u, i to do samo 10 istovremenih posjeta. Nema mogućnosti promjena na Registry, niti postoje nadogradnje.
5. S izuzetkom IIS 4, sve verzije IIS se isporučuju na CD-ROM-u operacijskog sustava i ne mogu se skinuti s Microsoftovog web site-a. IIS 4 možete skinuti s URL:
<http://www.microsoft.com/ntserver/nts/downloads/recommended/NT4OptPk/default.asp>
U istom NT 4 Option Pack-u se nalazi i PWS za Windows 95.

Zaključak

Zahvaljujući klijent verzijama IIS-a, mi na svom računalu već imamo ASP. Budući da je Microsoft Access postao uobičajen član Office-a, i taj program već imamo na računalu. Visual Basic 6 Working Model Edition je edukacijska verzija u kojoj se mogu kreirati ActiveX DLL komponente, a isporučivala se i uz neke inozemne knjige.

Za primjere web aplikacija, riješene samo pomoću ASP-a ili samo pomoću ASP-a i Accessa, vi imate sav potreban software.

U ovoj sam knjizi sve primjere radio u Visual Basic-u 6, dok sam bez znatnijih razlika eksperimentirao sa svim verzijama IIS-a, osim IIS 6, kao i sa svim verzijama Microsoft Access-a.

Microsoft Access

Za web aplikacije s relacijskom bazom podataka koristio sam Microsoft Access iz sljedećih razloga:

Od 1993. godine, kada je napravljena prva verzija, u Školi E-92 na Accessu podučavamo relacijske baze podataka, i na tu sam temu napisao više knjiga,

svi ga već imamo na svom računalu,

Access je izuzetno pogodan za učenje SQL programskog jezika. Upiti se kreiraju u prozoru Query By Example (QBE), grafički, a Access u pozadini ispisuje SQL kôd. Taj SQL kôd možemo kopirati (preslikati) na bilo koje mjesto naše aplikacije. Također ga možemo i razumjeti kada smo razumjeli grafički prikaz istoga.

Tko god zna koristiti Access, bez problema će moći koristiti i SQL Server. To je veoma važno jer uvijek naglašavam kako Access nije alat tipa "server" i kao takav nije za "Nettop", nego za "Desktop" aplikacije. Jedno je učiti ili napraviti aplikacije za manje zahtjevne Intranete, a posve drugo izići profesionalno na otvoreni, svakom u svijetu dostupni Internet.

Ima još jedan razlog, koji je to postao tek nakon izlaska Windows-a 2000. Više nije potrebno bazu u Access-u prijavljivati u sustav DNS. To znači da možemo našu web aplikaciju s Access bazom objaviti s našim web site-om kao da se radi isključivo s HTML-om. Svi koji su se koristili mojim knjigama FrontPage, mogli su vidjeti da je verzija 2000 dobila Lekciju 10 u kojoj se pomoću Wizard-a kreiraju web stranice za komunikaciju s Access bazom.

Pojeftinjenje ADSL usluge na svega 200 kn mjesečno za 1 Gb prometa, svima nam je približilo mogućnost posjedovanja Web poslužitelja u svojoj školi, stanu i sl. Budući da profesionalni ISP ne dopuštaju ugradnju našeg programskog kompajliranog kôda (DLL) na njihova host računala, školama i drugim institucijama pružena je izvanredna mogućnost da uz mala ulaganja dođu do svog poslužitelja na kojem mogu vježbati po želji, i iz svoga stana objavljivati svoje web aplikacije.

ADSL je stalna, 24-satna veza s Internetom koja se kao takva naplaćuje prema ostvarenom prometu, tj. količini bajtova. Do nedavno se nije mogla koristiti za osobne poslužitelje kojima se može pristupiti izvana, jer je u tijeku 24 sata nekoliko puta mijenjala svoju Internet adresu (IP). To znači da se naš URL, preko

kojega nam pristupaju vanjski posjetitelji, također mora svaki put mijenjati. Pronađen je način da URL ostaje isti bez obzira što se IP mijenja, bez dodatnih ulaganja, pa je otvorena mogućnost i najmanjim firmama i školama da sebi priušte web hosting kojem će pristupati i iznutra, kao Intranetu, i izvana, kao Internetu. Novom organizacijom poslovanja, koja uključuje Internet, firme mogu uštedjeti mnogo, a mogućnosti učenja na daljinu nisu više ograničene. I u poslovanju i u učenju jedino je ograničenje naša mašta.

SQL Server

Prijelaz na SQL Server s Access-a je bezbolan. I SQL Server nudi grafičko sučelje za kreiranje SQL upita, a izrada tablica je istovjetna onoj u Access-u. Osim nekoliko manjih razlika u terminologiji tipova polja, poznavatelj Access-a će se u okruženju SQL Servera osjećati kao kod kuće.

U profesionalnoj izradi web aplikacija s bazama treba zaboraviti na Access, i svoje baze prebaciti na SQL Server. SQL Server se obvezno i dalje prijavljuje na DSN sustav na poslužitelju. Ukoliko smo web aplikaciju napravili s Access bazom, a istu prijavili također na sustav DSN, na primjer pod imenom eKartelo, promjena s Access-a na SQL Server bazu najčešće ne zahtijeva nikakav zahvat u programski kôd web aplikacije. Dovoljno se držati istog naziva eKartelo i za novu bazu u SQL, a nazivi polja i tablica u SQL bazi su ostali isti kao u Access bazi.

Gdje je zapinjalo

Tutorial Windows XP Professional IIS 5.1

Primjer 1 iz Microsoftovog Tutoriala: zašto u VBScriptu moramo pisati brojeve ugrađenih konstanti, a u Visual Basic-u to ne moramo.

Instalacija IIS 5.1 na Windows XP Professional

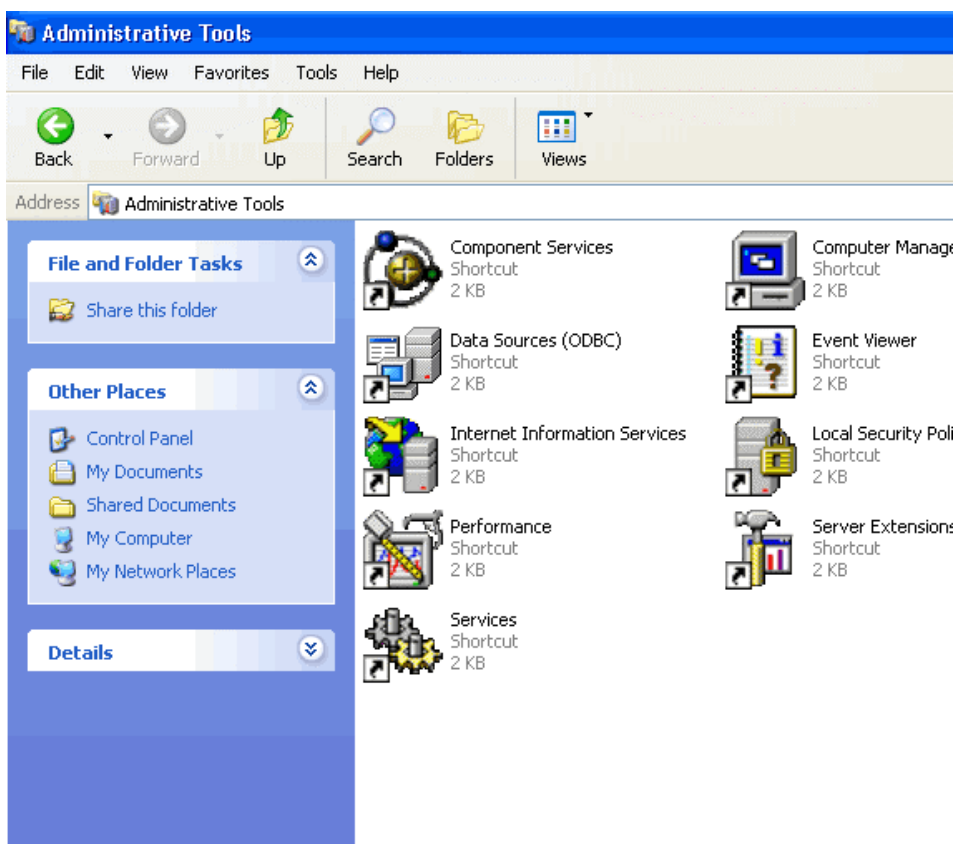
Na CD-u Windows XP Professional (ne i Home verzije) nalazi se IIS 5.1. Instalirajte IIS 5.1 naredbama:

1. Stavite CD Windows XP Professional u pogon.
2. Start/Control Panel/Add or Remove Programs

Nakon instalacije IIS 5.1, pokrenite IIS 5.1 ovako:

Pokretanje IIS

1. Start/Control Panel/Administrative Tools i otvorit će se ovaj prozor:



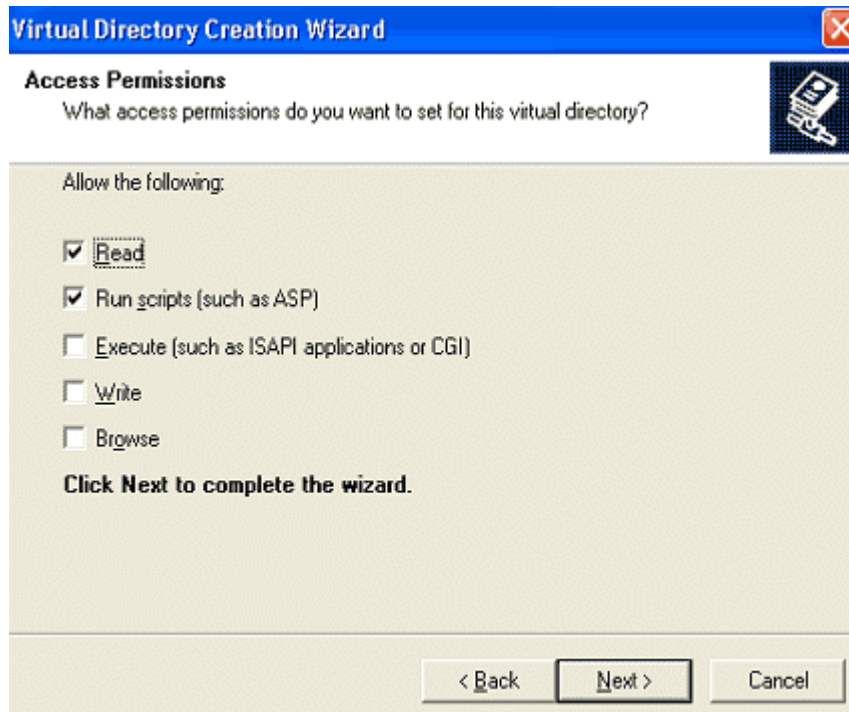
2. Pritisnite tipku miša dva puta na Internet Information Services i otvorit će se ovaj prozor:



3. Razgranajte (local computer).

Kreiranje virtualne mape

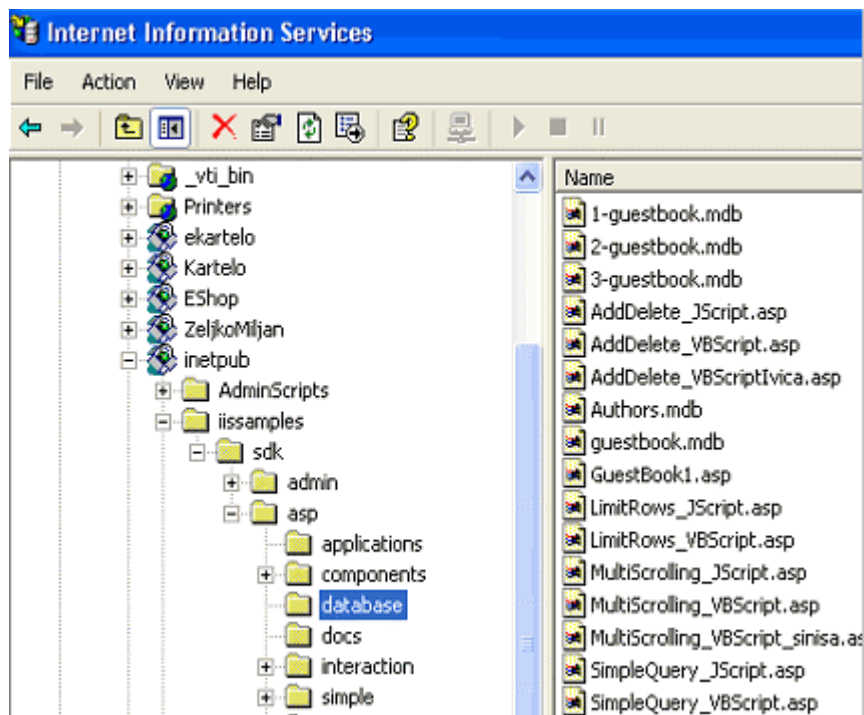
1. Pritisnite desnu tipku miša na Default Web Site, odaberite New, pa Virtual Directory, Next.
2. U polje Alias tipkajte ime virtualnog direktorija, na primjer **inetpub**, kao što se naziva i fizička mapa u kojoj su Microsoftovi ASP primjeri za vježbu.
3. Next. Browse. Označite mapu Inetpub koja se nalazi na disku C:\.
4. Next, zatim uključite ove opcije:



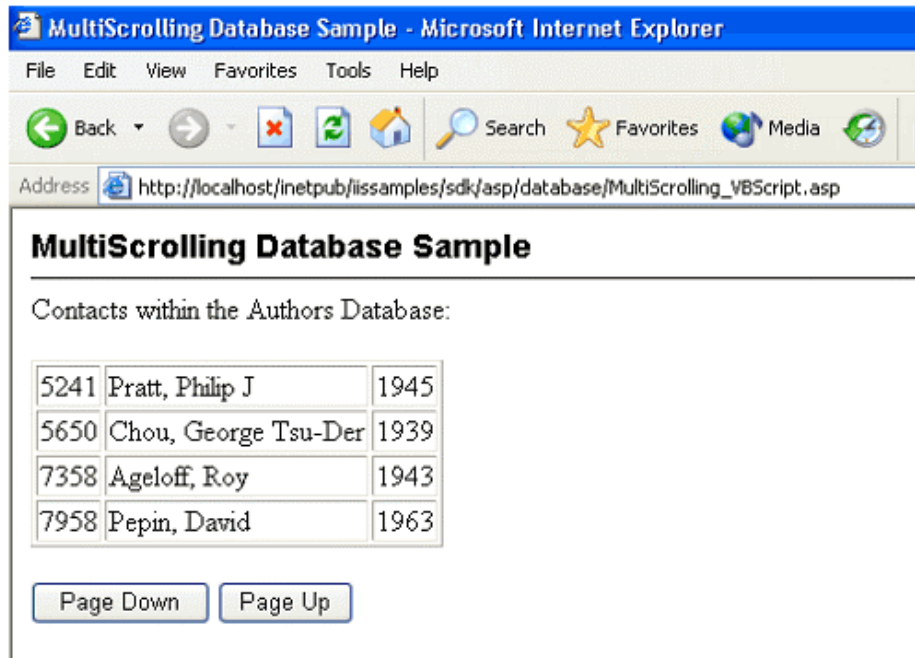
5. Next. Finish. Tako ste dobili virtualnu mapu u kojoj je cijela struktura podmapa.

Primjeri ASP stranica se nalaze u iissamples

1. Razgranajte strukturu mapa unutar virtualnog prostora Inetpub i otvorite mapu database čiji je put:
Inetpub/ inetpub/iissamples/sdk/asp/database/



2. i u mapi database otvorite ASP stranicu MultiScrolling_VBScript.asp, tako što ćete na istu pritisnuti desnu tipku miša i odabrati naredbu Browse.
3. Stranica MultiScrolling_VBScript.asp će se otvoriti u Internet Exploreru...



4. ...i vi ćete je provjeriti; sve će vjerojatno raditi kako treba.
5. U polju Address piše virtualna adresa:
http://localhost/inetpub/iissamples/sdk/asp/database/MultiScrolling_VBScript.asp
i zato ASP stranica radi ispravno.
6. Onda ćete se vratiti u prozor IIS i otvoriti kôd iste stranice i učiti na primjeru.

Došli smo do mjesta na koje vam želim skrenuti pozornost. To je početak ASP stranice i ovaj kôd:

```
<!--METADATA TYPE="typelib"
```

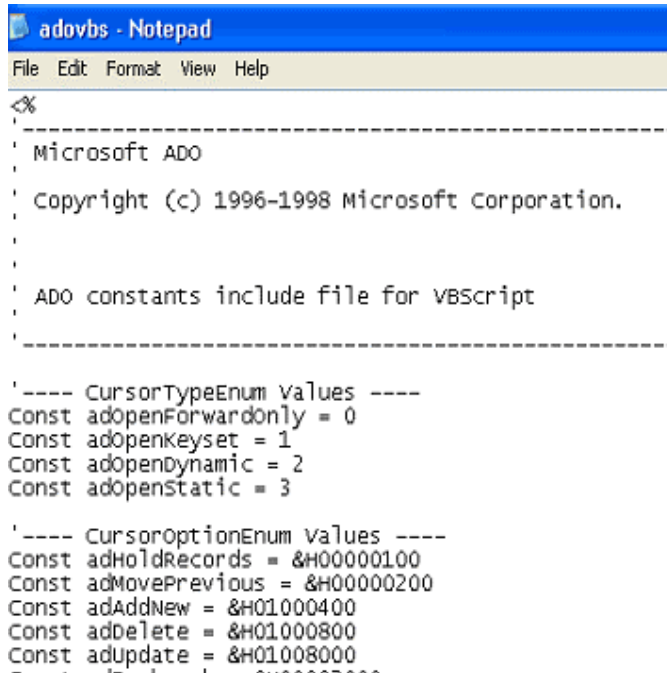
```
uuid="00000206-0000-0010-8000-00AA006D2EA" -->
```

To ćete obrisati, pohraniti i provjeriti stranicu. Ne radi! Vratite to, ona opet radi. Ali sve stranice do sada su vam radile bez toga. Zašto?

Odgovor: ADO komponenta ASP-a koristi posebne tekst datoteke u kojima definira konstante koje moramo znati kada u skriptu manipuliramo podacima iz baze

podataka. Naredba METADATA je slična naredbi #include i poziva stranicu čiji je uuid broj jednak

```
uuid="00000206-0000-0010-8000-00AA006D2EA4"
```



```
adovbs - Notepad
File Edit Format View Help
<%
-----
' Microsoft ADO
' Copyright (c) 1996-1998 Microsoft Corporation.
'
' ADO constants include file for VBScript
-----

'---- CursorTypeEnum Values ----
Const adOpenForwardOnly = 0
Const adOpenKeyset = 1
Const adOpenDynamic = 2
Const adOpenStatic = 3

'---- CursorOptionEnum Values ----
Const adHoldRecords = &H00000100
Const adMovePrevious = &H00000200
Const adAddNew = &H01000400
Const adDelete = &H01000800
Const adUpdate = &H01008000
```

Takve stranice imaju nastavak INC i otvaraju se u Notepad-u, ali ne uvijek automatski, nego morate primijeniti naredbu Open with.

Sve stranice s INC nastavkom koje se nalaze na vašem računalu, u ASP okruženju možete pronaći pomoću Start/Search/Tipkajte **ado*.inc**

Ta se stranica nalazi na IIS 5.1. To ne znači da se nalazi i na računalu vašeg kupca. Zato je bolje ne biti ovisan o njoj. Radi te činjenice...

...dođite do retka...

```
oRs.Open "SELECT * FROM Authors order by author", oConn, adOpenStatic
```

... i obrišite konstantu 'adOpenStatic' i umjesto nje tipkajte 3...

```
oRs.Open "SELECT * FROM Authors order by author", oConn, 3
```

... sada obrišite...

```
<!--METADATA TYPE="typelib"
```

```
uuid="00000206-0000-0010-8000-00AA006D2EA4" -->
```

...pohranite i provjerite. Radi u redu!

Naučili smo u Visual Basic-u koristiti konstante pomoću njihovih imena, ali u ASP-u se moramo učiti koristiti brojeve umjesto imena. Visual Basic je alat koji će uvijek iza scene za nas tipkati broj, dok je nama lakše zapamtiti i koristiti naziv konstante. Prilikom kompajliranja potreban je broj, i VB će kompajlirati broj.

VBScript nema tako kompaktnu logistiku. On je potpuno ogoljen i jako ovisan o okolini koja ga čeka na IIS-u, konkretnije u okolini ASP. Kad ASP procesuirá (kompajlira) redak...

```
oRs.Open "SELECT * FROM Authors order by author", oConn, adOpenStatic
```

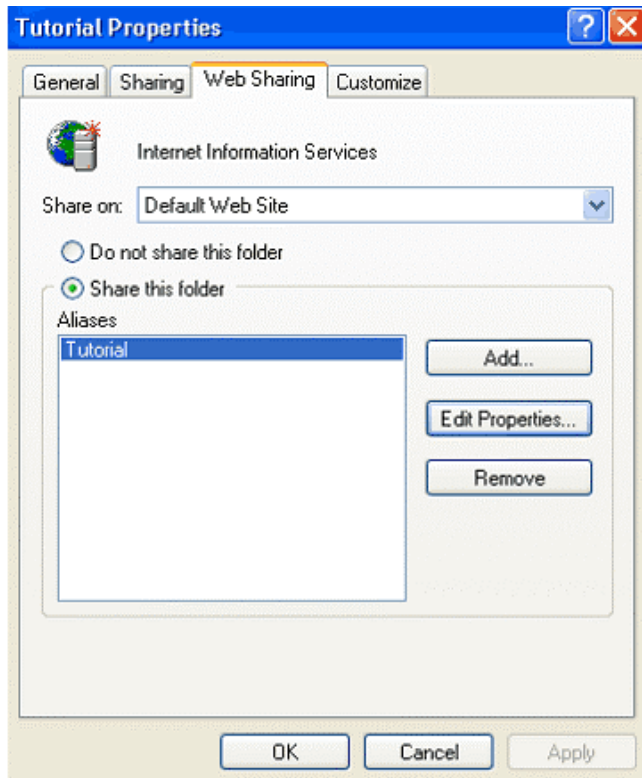
...i naiđe na ime konstante 'adOpenStatic', on će pozvati dokument 'adovbs' kako bi doznao broj i kompajlirao broj.

Primjer 2 iz Microsoftovog Tutoriala: zašto primjer s bazom Authors.mdb radi, a primjer s bazom guestbook.mdb ne radi

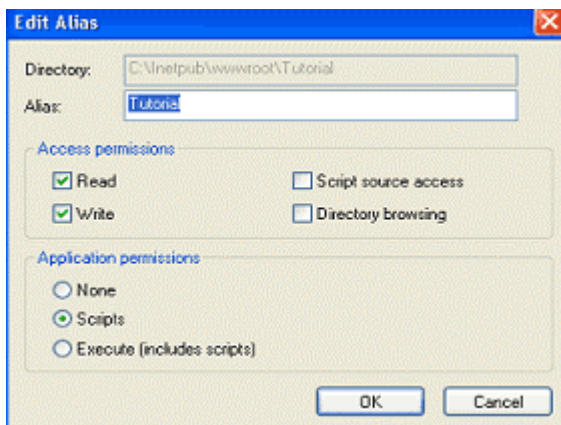
U mapi 'database' je primjer s bazom 'Authors.mdb'. Radi sve: Add, Delete, Update, Edit.

U mapi Tutorial je primjer s bazom 'guestbook.mdb'. Na bazu se možete spojiti, možete je pregledavati pomoću SQL naredbe Select..., ali ne možete joj dodati nove podatke. Kôd pada na retku gdje ste primijenili SQL naredbu Insert.

Uzrok su "permissions" ili dozvole pristupa mapama. U Windows Explorer-u pritisnite desnu tipku miša na mapu Tutorial, Properties, Web Sharing pa uključite Share this folder:



Odaberite Edit Properties i uključite ove opcije:



Ako nije uključena "permission" Write, ne možemo unositi podatke u baze koje se nalaze u takvim mapama. Zanimljivo je i to da smo dozvole pristupa mapama odredili u Windows Explorer-u, a ne izravno u IIS.

Obratite pozornost na gornji prozor: iako smo u Windows Explorer-u, u prozoru se navodi i virtualno ime mape (Alias) uz fizički put te mape (Directory).

Lekcija 1 ASP

počnimo s primjerom

Počet ćemo s primjerom kako bi nam u nastavku bilo lakše s teorijom. Ukoliko vama više odgovara obrnuto, najprije idite na *Lekciju 2*, pa onda na *Lekciju 1*.

Primjer je rađen na svim Windowsima i odgovarajućim IIS i VB 6.

Vlasnik lanca trgovina pored redovne plaće svoje trgovce nagrađuje s 4% od ukupne prodaje. Kako bi trgovac bio zainteresiran za prodaju što više robe svakom kupcu, formula za nagradu glasi: $\text{ukupni iznos} * 4\% + ((\text{Ukupni iznos prodaje}) / (\text{Broj izdanih računa})) * 1\%$.

Vlasnik će tu formulu ugraditi u komponentu na svom web site-u. Trgovci će preko web stranica u svakom trenutku moći izračunati svoje nagrade. Ukoliko vodstvo promijeni pravila nagrađivanja, te će promjene ugraditi u komponentu i one će biti u funkciji na Web-u i svima dostupne.

Izrada komponente

Ovu komponentu, kao i sve u ovoj knjizi, napraviti ćemo u Visual Basic-u 6. Komponentu s imenom **Nagrada.dll** pohranite u pripremljenu mapu na disku vašeg računala, npr. **C:\KnjigaASP\Lekcija1**. Također, u istu mapu pohranite i Visual Basic Project1 i formu Class1 pod imenima navedenim dolje. Tako ćete sve svoje uratke imati na jednom mjestu. Za web aplikaciju sasvim je svejedno gdje ste pohranili DLL komponentu. To je važno vama radi bolje preglednosti. "Sirovi" radovi Visual Basic Project i forma Class su samo stepenice na putu kreiranja DLL komponente, i samo je komponenta važna za našu web aplikaciju.

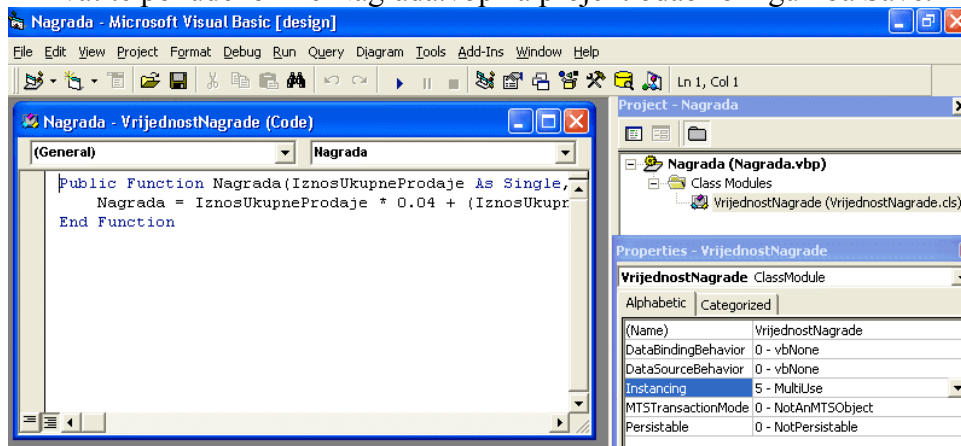
1. Otvorite Visual Basic.
2. U dijalogu New Project odaberite ActiveX DLL, pa gumb Open.
3. U prozoru Project1-Class1 tipkajte ovaj kôd:

```
Public Function Nagrada(IznosUkupneProdaje As Single,
BrojRacuna As Integer) As Single
```

```
Nagrada = IznosUkupneProdaje * 0.04 +
(IznosUkupneProdaje / BrojRacuna) * 0.01
```

```
End Function
```

4. Pritisnite desnu tipku miša na Project1 u Project Explorer-u, pa u skraćenom izborniku odaberite Project1 Properties.
5. U dijalogu Project Properties, u polju Project Name tipkajte ime projekta **Nagrada**.
6. OK.
7. Pritisnite tipku miša na Class1 u Project Explorer-u, pa u prozoru Properties tipkajte **VrijednostNagrade** umjesto Class1 za svojstvo Name.
8. Pohranite projekt naredbom File/Save Project.
9. Prihvatite ponuđeno ime VrijednostNagrade.cls za class modul odabirom gumba Save.
10. Prihvatite ponuđeno ime Nagrada.vbp za projekt odabirom gumba Save.



DLL komponenta

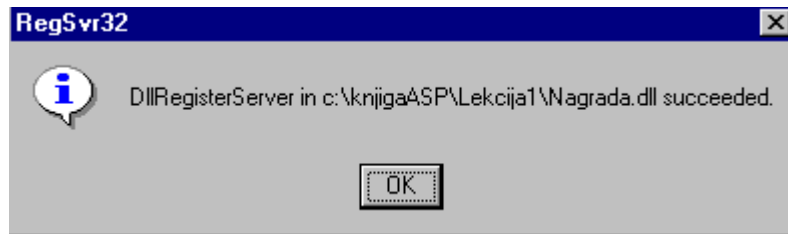
Sada pohranimo DLL komponentu.

1. File/Make Nagrada.dll.
2. Prihvatite ime Nagrada u dijalogu Make Project odabirom gumba Save.

Iako se nakon ovoga dogodila i registracija komponente, u svojoj praksi naučio sam da je i te kako dobra navika registrirati komponentu još jednom ručno.

Registracija DLL komponente

1. Start/Run.
2. U dijalog Run tipkajte cijeli put vaše DLL komponente:
`regsvr32 c:\knjigaASP\Lekcija1\Nagrada.dll`
3. Ukoliko je sve u redu, slijedi dijalog-poruka o uspješnoj registraciji komponente:



4. OK.

Izmjene u komponenti

Ukoliko želite napraviti promjene na komponenti, najprije morate poništiti registraciju iste:

1. Start/Run.
2. U dijalog Run tipkajte cijeli put vaše DLL komponente:
`regsvr32 /u c:\knjigaASP\Lekcija1\Nagrada.dll`
3. Ukoliko je sve u redu, slijedi dijalog-poruka o uspješno poništenoj registraciji komponente.
4. Sada možete obrisati komponentu Nagrada.dll i otvoriti Visual Basic projekt Nagrada.vbp, potom napraviti promjene na projektu, pohraniti ih i na kraju pohraniti novu komponentu Nagrada.dll i registrirati je.

Izrada HTML dokumenta

1. Otvorite svoj HTML editor (najjednostavniji editor je Notepad) i tipkajte ovaj HTML kôd:

<HTML>

<HEAD>

```

<TITLE>Komponenta i ASP</TITLE>

</HEAD>

<BODY>

<FORM ACTION="izracunajnagradu.asp" METHOD=POST>

<BR>Iznos ukupne prodaje: <INPUT TYPE="TEXT"
NAME="iznosukupneprodaje" VALUE="" SIZE=30>

<BR>Broj ukupno izdanih racuna: <INPUT TYPE="TEXT"
NAME="brojukupnoizdanihracuna" VALUE="" SIZE=20>

<P><INPUT TYPE="SUBMIT" NAME="" VALUE="Nagrada iznosi">

<INPUT TYPE="RESET" NAME="" VALUE="Obrisi sve">

</FORM>

</BODY>

</HTML>

```

2. Pohranite HTML dokument pod imenom **nagrada.htm**, u mapu
C:\KnjigaASP\Lekcija1\

Izrada ASP stranice

U istom editoru, u kojem ste upravo kreirali HTML stranicu, možete kreirati i sljedeću ASP stranicu.

```

<HTML>

<HEAD>

<TITLE>Komponenta i ASP</TITLE>

</HEAD>

<BODY>

<%

```

```

'Uzimanje vrijednosti iz forme

ukupnaProdaja = Request.Form("iznosukupneprodaje")

ukupnoRacuna = Request.Form("brojukupnoizdanihracuna")

%>

Ukupna prodaja: <% = FormatCurrency(ukupnaProdaja) %>
<BR>

Ukupno racuna: <% = ukupnoRacuna %> <P>

<%

'Kreiranje instance nase nove komponente Nagrada.dll

Set iznosNagrade =
Server.CreateObject("Nagrada.VrijednostNagrade")

%>

<%

'Dobijanje rezultata i prikazivanje istog na web
stranici

iznos = iznosNagrade.Nagrada(csng(ukupnaProdaja),
cint(ukupnoRacuna))

Response.Write("Nagrada iznosi: " +
FormatCurrency(iznos))

%>

</BODY>

</HTML>

```

1. Pohranite ovaj ASP dokument pod imenom **izracunajnagradu.asp** u istu mapu u koju ste pohranili i gornji HTML dokument nagrada.htm (u mapu C:\KnjigaASP\Lekcija1\).

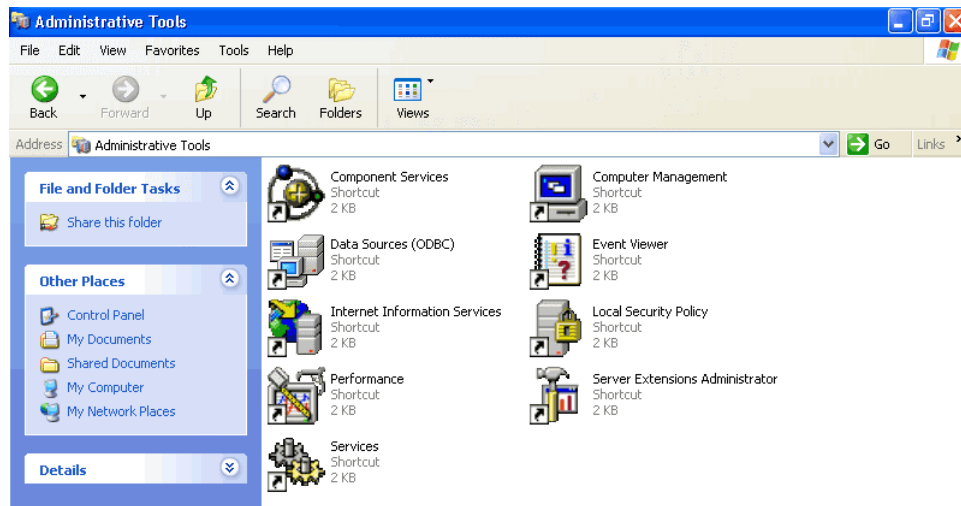
Fizička mapa i virtualna mapa

Windows NT 4 i Option Pack 4

1. Start/Programs/Windows NT 4.0 Option Pack/Microsoft Internet Information Server/Internet Service Manager.
2. Razgranajte stablo na lijevoj strani prozora dok ne ugledate Default Web Site.
3. Pritisnite desnu tipku miša na Default Web Site, pa u skraćenom izborniku odaberite New, Virtual Directory.
4. Tipkajte naziv virtualne mape, npr. **Kartelo**, pa odaberite Next.
5. Odaberite Browse i pritisnite tipku miša na fizičku mapu C:\KnjigaASP\
6. Ja sam u Windows Explorer-u otvorio mapu KnjigaASP na C:\ i u njoj podmapu Lekcija1. Sve što sam napravio u poglavlju Lekcija 1, pohranit ću u mapu Lekcija1. Sada kad je trebalo virtualnoj mapi pridružiti fizičku mapu, ja sam joj pridružio mapu KnjigaASP, iako to nije direktna mapa u kojoj su HTML i ASP dokumenti. Mapa KnjigaASP je 'roditelj' mapi Lekcija1.
7. Next.
8. Prihvatite predzadane opcije i odaberite Finish.

Windows XP i IIS 5,1

1. Start/Control Panel/Administrative Tools.

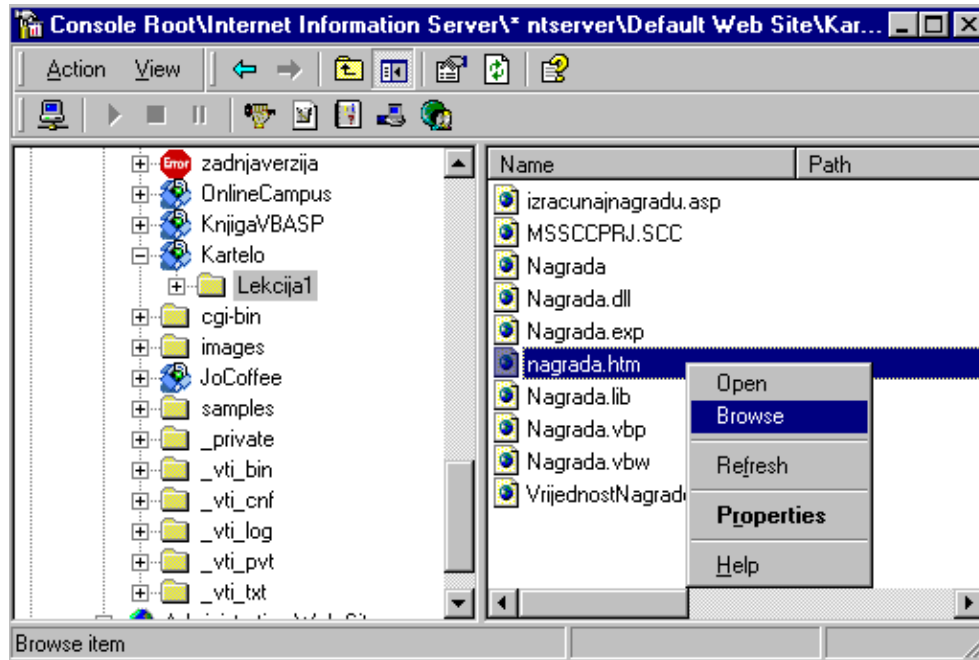


2. Odaberite Internet Information Services.
3. Dalje je sve isto kao za NT 4 i IIS 4,0.

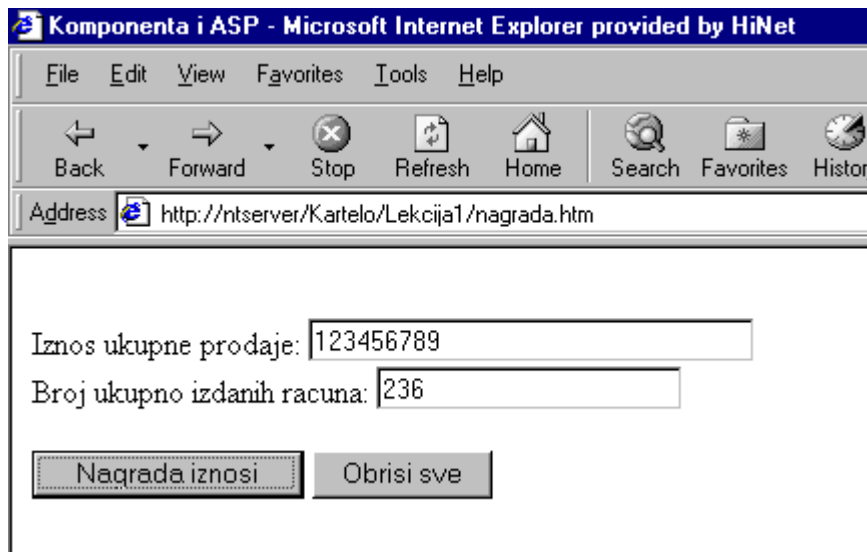
A sada, provjerimo u radu HTML, ASP i komponentu

Windows NT 4 i Option Pack 4

1. Start/Programs/Windows NT 4.0 Option Pack/Microsoft Internet Information Server/Internet Service Manager.
2. Otvorite mapu Lekcija1, pritisnite desnu tipku miša na HTML dokument nagrada.htm.

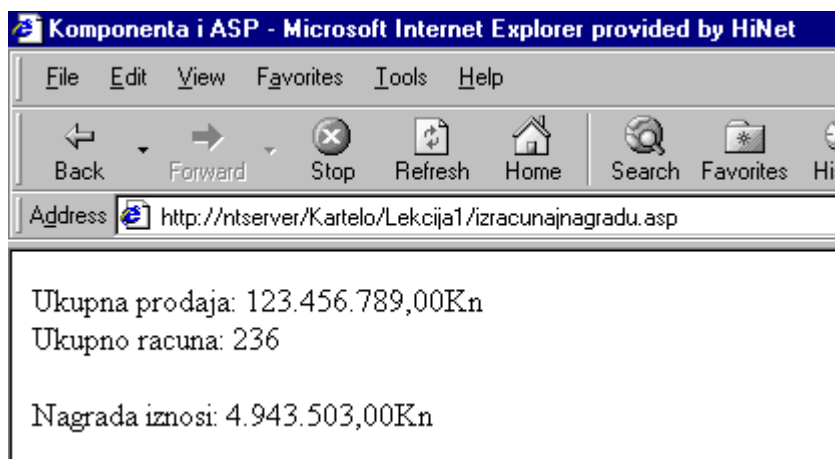


3. U skraćenom izborniku odaberite Browse.
4. U Web browser-u ispunite formu kao na slici:



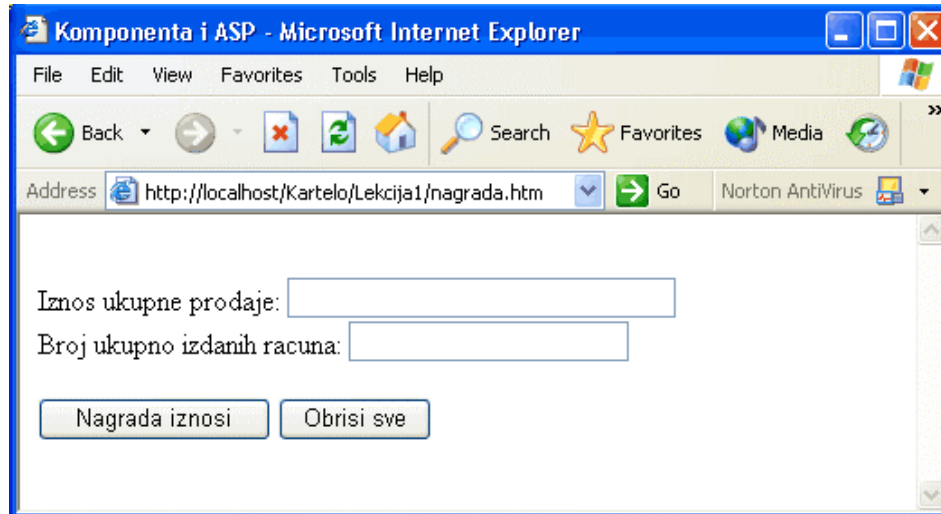
Obratite pozornost na URL u polju Address:
<http://ntserver/Kartelo/Lekcija1/nagrada.htm> Adresa počinje s http, što znači da radim preko lokalnog Web Server-a.

5. Pritisnite gumb 'Nagrada iznosi'.



Windows XP i IIS 5,1

1. Start/Control Panel/Administrative Tools/Internet Information Services.
2. Dalje je kao i na NT 4.



Ovdje je URL: <http://localhost/Kartelo/Lekcija1/nagrada.htm>

Analiza

Cijela web aplikacija se sastoji od HTML dokumenta, ASP dokumenta i komponente. Svi naši učenici u programu Internet programer na putu do ove knjige su prošli već tri knjige o Visual Basic-u i naučili oblikovati ActiveX DLL komponente. Također znaju napraviti formu u HTML dokumentu, makar i pomoću FrontPage-a. U gornjem HTML dokumentu je novost:

```
ACTION="izracuna_jnagradu.asp" METHOD=POST
```

To znači: kad posjetitelj u formi pritisne gumb tipa Submit, prema poslužitelju će otići zahtjev za web stranicom `izracuna_jnagradu.asp`. Zahtjev će biti prosljeđen metodom POST, o čemu detaljnije u nastavku knjige.

I evo nas na web stranici koja ima nastavak `.asp`, a ne, kao što smo naučili, nastavak `.htm`.

Nakon pritiska na gumb Submit, prema IIS je krenuo zahtjev za stranicom `izracuna_jnagradu.asp`.

Radi nastavka `.asp`, IIS prosljeđuje zahtjev prema svom dijelu koji se naziva ASP. ASP procesira stranicu `izracuna_jnagradu.asp` ovako...

...ja sam ASP i od mene se uvijek očekuje da posjetitelju vratim HTML dokument. Ovaj posjetitelj je pozvao stranicu izracunajnagradu.asp koja je sada u mojim rukama. Da je stranica kojim slučajem imala nastavak .htm, onda i ne bi došla u moje ruke nego bi IIS odmah znao koja joj je fizička adresa i poslao bi njenu presliku posjetitelju. Sve stranice s nastavkom .asp dolaze najprije u moje ruke, i ja od njih moram napraviti stranice koje su "čisti" HTML kôd. Na primjeru ove stranice vidjet ćete kako ja to radim. Počinjem čitati stranicu odozgo prema dolje.

...ovo su HTML tagovi i njih ostavljam na stranici onakvima kakvi jesu...

```
<HTML>
<HEAD>
<TITLE>Komponenta i ASP</TITLE>
</HEAD>
<BODY>
```

...evo znakova kutne zagrade i postotka. To je znak početka nekog skript programskog jezika. Budući da ne piše kojega jezika, proslijedit ću taj dio svom interpreteru VBScript namještenom kao primarnom interpreteru. Interpreter će kompajlirati skript i dobiti rezultat...

```
<%
```

... apostrof znači da je riječ o komentaru, pa taj komentar neću prikazati na web stranici...

```
'Uzimanje vrijednosti iz forme
```

...evo prve programske naredbe (instrukcije). Ta naredba kreira varijablu ukupnaProdaja i pridjeljuje joj vrijednost koju je posjetitelj utipkao u formi u polje iznosukupneprodaje. Ta se vrijednost nalazi u memoriji računala-poslužitelja. Tamo ju je pohranila moja ugrađena komponenta Request, točnije objekt Form komponente Request...

```
ukupnaProdaja = Request.Form("iznosukupneprodaje")
```

...slijedi naredba kao i što je bila prethodna. Naredba koja kreira varijablu ukupnoRacuna i pridjeljuje joj vrijednost koju je posjetitelj utipkao u formi u

polje brojukupnoizdanihracuna. Ta se vrijednost nalazi u memoriji računala-poslužitelja. Tamo ju je pohranila moja ugrađena komponenta Request, točnije objekt Form komponente Request...

```
ukupnoRacuna = Request.Form("brojukupnoizdanihracuna")
```

...i evo opet znakova postotka i kutne zagrade samo poredanih obrnuto nego na početku ove sekcije. To za mene znači – ne treba mi više VBScript interpreter...

```
%>
```

...i ovo što slijedi uzimam "zdravo za gotovo" i ugrađujem u web stranicu, jer to pripada HTML-u...

Ukupna prodaja:

... u HTML je ukopan (in-line) VBScript koji ponovo prosljeđujem svom VBScript interpreteru, koji će taj skript kompajlirati i razumjeti. Samo ću vrijednost varijable ukupnaProdaja formatirati u format Currency...

```
<% = FormatCurrency(ukupnaProdaja) %>
```

...uzimam HTML kao takav...

```
<BR>
```

Ukupno racuna:

...interpreter VBScript kompajlira ovaj VBScript kôd...

```
<% = ukupnoRacuna %>
```

...pa opet slijedi HTML koji ostavljam takvim kakav jest...

```
<P>
```

...interpreter VBScript kompajlira kôd VBScript i memorija poslužitelja postaje bogatija za instanciju objekta VrijednostNagrade. Instancija objekta je referencija prema objektu. **Referenciju** prema objektu je kreirala metoda CreateObject. Vlasnik metode je moja ugrađena komponenta Server...

```
<%
```

```
'Kreiranje instance naše nove komponente Nagrada.dll
```

```
Set iznosNagrada =  
Server.CreateObject("Nagrada.VrijednostNagrada")
```

... na ovom mjestu u memoriji imam objekt iznosNagrada. Taj objekt je produkt naše osobne class forme VrijednostNagrada. Ta class forma je dio Visual Basic projekta Nagrada.vbp. Projekt Nagrada.vbp smo kompajlirali u komponentu Nagrada.dll. Class forma u komponenti predstavlja objekt. Gornji VBScript kôd kreira instanciju upravo tog objekta VrijednostNagrada.

Slijedi komentar...

```
'Dobijanje rezultata i prikazivanje istog na web  
stranici
```

...pa slijedi oblikovanje nove varijable iznos. U istoj naredbi varijabli iznos dodijeljena je i vrijednost. Koja vrijednost? Ova vrijednost: = rezultat metode Nagrada. Pogledajte tu metodu u class formi VrijednostNagrada.

```
Public Function Nagrada(IznosUkupneProdaje As Single, BrojRacuna  
As Integer) As Single
```

```
    Nagrada = IznosUkupneProdaje * 0.04 + (IznosUkupneProdaje /  
BrojRacuna) * 0.01
```

```
End Function
```

Metoda Nagrada ima dva ulazna parametra IznosUkupneProdaje As Single, BrojRacuna As Integer. Ti ulazni parametri postaju sudionici matematičke operacije. Rezultat te matematičke operacije metoda vraća onom tko ju je pozvao. U ovoj VBScript naredbi...

```
iznos = iznosNagrada.Nagrada(csng(ukupnaProdaja),  
cint(ukupnoRacuna))
```

...događa se sljedeće:

1. Kreira se varijabla iznos.
2. Poziva se metoda Nagrada.
3. Metodi Nagrada se šalju vrijednosti koje je posjetitelj tipkao u formi. Svaki posjetitelj na svoj način unosi brojeve. Netko koristi točku za decimalni zarez, a netko koristi zarez. Metoda Nagrada, međutim, traži jedan parametar tipa Single, a drugi tipa Integer, i to napisane po pravilima koja vrijede na

poslužitelju na kojem se metoda nalazi (odabrane opcije Regional Settings na poslužitelju).

Zato je potrebno izvršiti CSng i CInt konverzije podataka koje je unio posjetitelj u formu.

CSng je funkcija koja izvršava konverziju bilo kojih podataka u podatke tipa Variant podtip Single i internacionalno je prepoznatljiva. Bez obzira što je korisnik unio u formu, funkcija CSng će to uvijek konvertirati u oblik koji je zadan opcijom Regional Settings na poslužitelju. Ako ulazna vrijednost leži izvan granica podtipa Single, na zaslonu ćemo imati poruku o grešci.

CInt također razlikuje internacionalne načine pisanja brojeva i radi pravilnu konverziju. Ova funkcija konvertira bilo koji tip podataka u tip Integer. Ukoliko ulazna veličina leži izvan granica tipa Integer, sustav će nam javiti grešku.

4. Te lokalizirane vrijednosti su ulazni parametri za metodu Nagrada.
5. Metoda Nagrada napravi proračun s tim vrijednostima.
6. Metoda Nagrada pohrani u varijabli `iznos` rezultat tog proračuna.

Jedna jedina naredba napisana u VBScriptu na ASP stranici toliko posla učini!!! Eto zašto je ASP omiljen kod programera. Jedan redak kôda u ASP okruženju zamijenio je na desetke redaka kôda u prethodnim okruženjima. Microsoftovi projektanti žele napraviti okruženje koje će, što je moguće više, smanjiti broj redaka ostalim programerima. Kako to postižu? To postižu tako što odmiču programere od operacijskog sustava. Sloj po sloj, ali programer je sve dalje od Windowsa. O tome će biti riječi u nastavku, ali je ovdje zgodno, uz svjež primjer, podsjetiti se sloja ActiveX na kojem leži sloj Components, pa slijedi sloj Scripts, a posljednji je sloj ASP. ASP je vrh piramide. Trenutno smo vrlo, vrlo daleko od Windowsa.

ASP nastavlja procesuiranje naše ASP stranice...

...slijedi naredba u kojoj na scenu stupaju ugrađena ASP komponenta Response i njena metoda Write...

```
Response.Write("Nagrada iznosi: " +  
FormatCurrency(iznos))
```

```
%>
```

...rezultat koje će biti ispis nagrade u formatu Currency.

Slijede HTML tagovi koje ostavljam onakvima kakvi jesu...

```
</BODY>
```

```
</HTML>
```

Kraj ASP procesuiranja, rezultat kojega je ovaj HTML dokument:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Komponenta i ASP</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Ukupna prodaja: $123,456.56 <BR>
```

```
Ukupno racuna: 3215 <P>
```

```
Nagrada iznosi: $4,938.65
```

```
</BODY>
```

```
</HTML>
```

Taj HTML dokument ASP ispostavlja IIS-u, koji ga potom prosljeđuje posjetitelju. Obratite pozornost na polje Address u Web browser-u:

<http://localhost/Kartelo/Lekcija1/izracunajnagradu.asp>

Na zaslону smo dobili stranicu koju smo tražili, `izracunajnagradu.asp`, ali priznat ćete, puno se smanjila. Sada je ta stranica čisti HTML. U njoj nema VBScripta i znakova postotka. Ona duža varijanta stranice `izracunajnagradu.asp` ostala je na poslužitelju. Na njoj su VBScripti daleko od očiju posjetitelja. Ona čeka novi poziv kad će joj novi posjetitelj preko forme proslijediti nove brojeve. A ona će tom posjetitelju također vratiti druge brojeve nego prethodnom. Znači, na svaki poziv odgovara drugim sadržajem, drugim brojkama. Iz ovoga zaključujemo dvije stvari:

1. ASP proizvodi, bez pomoći vlasnika site-a, dinamički sadržaj ili DHTML stranice.

2. ASP kreira interaktivnost, automatski, bez pomoći vlasnika site-a.

A to dvoje, dinamički sadržaj i interaktivnost i jest Internet.

Napomena: Razlikujmo DHTML postignut pomoću skript programiranja na klijent strani od DHTML postignutog programiranjem na strani poslužitelja! Pod skraćenicom DHTML uobičajeno je misliti na ono prvo. To su HTML dokumenti koji na klijent strani mogu mijenjati sadržaj i izgled i nakon što se prekine veza s poslužiteljem. Kada je o ASP-u riječ, onda je pravi izraz *dinamički kreiran HTML* umjesto samo *dinamički HTML*.

Međutim, u svojim web aplikacijama, kao što ćete sretati i u ovoj knjizi, koristit ćete na istoj ASP stranici skript za ASP (za stranu poslužitelja), ali i skript za klijent stranu. ASP će jako dobro razlikovati što je za njega, a što nije. Ono što nije za njega vratit će klijentu. Tako će ASP vratiti klijentu HTML stranicu koja u sebi sadrži, pored HTML tagova, još i skript-klijent-strane. To dalje znači da ASP ne vraća uvijek čisti HTML. Može on vratiti HTML i JavaScript ili VBScript, ali će znati zašto je to učinio. Sada znate i vi.

Radi svega navedenoga, a skraćenica ima ionako puno, ostajemo samo pri jednom izrazu, DHTML, ali kreator web aplikacije veoma dobro zna razliku koja može postojati.

Moja osobna učinkovitost do ljeta 2003.

Vratio bih se na ona dva najvažnija pojma u Internet tehnologiji: dinamički kreiran sadržaj i interaktivnost. Od 1996. do danas te dvije najvažnije stvari na Internetu postižem ručno. Evo samo jedan primjer:

1. Kupac online naruči download knjigu.
2. Ja ručno pretipkavam kupca u Access aplikaciju, u tablicu Kupci.
3. Ručno pravim predračun u istom Access-u.
4. .rtf format tog predračuna stavljam u privitak e-maila.
5. Upisujem nekoliko riječi u e-mail.
6. Send.

7. Nakon par dana stiže potvrda o uplati, i ja ručno e-mailom javljam URL na kojoj kupac može skinuti moju knjigu u PDF formatu.

Odlično, zar ne!

A još 1998. godine sam doživio sljedeće:

1. Subota poslije podne, napolju kiša, zima, a ja želim svoj web site obogatiti besplatnim download čuvarima ekrana (Screen Savers).
2. Tipkam u Altavistu "Screen Saver" i nalazim firmu koja prodaje program za izradu Screen Savera za 19 USD.
3. Isprobam trial verziju. Odlično. Međutim, ne mogu se riješiti teksta "ovo je napravljeno ilegalnom kopijom...". Kako ću poklanjati posjetiteljima Screen Savere s takvim tekstom.
4. Klik, klik, podaci s moje American Express kartice su unešeni, Send. Ne sjećam se točno koliko je vremena, u sekundama, prošlo, ali sam vrlo brzo na zaslonu ugledao: "Hvala, vaša kartica je ispravna, evo brojeva koje morate utipkati na to i to mjesto i onaj tekst će nestati iz vaših čuvara ekrana."
5. Postupio sam po uputama i za pola sata napravio sve Screen Savere koji su mi ležali na srcu, a isti su još na mom site-u www.e92.hr link download.

A moj prijatelj Joža kaže kako je moja učinkovitost ohooooooooo visoko u odnosu na naš prosjek. Misli, naravno, na sve nas, i one koji još nisu na Internetu.

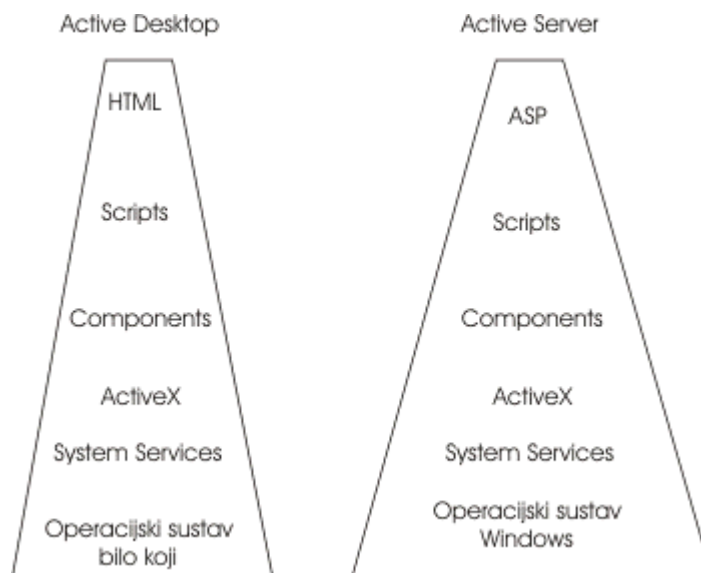
Sada znate zašto sam žarko prionuo na učenje programiranja za Internet. Da mogu sebi napraviti e-knjižaru koju su ti momci imali još davne 1998. godine. A koliko toga moram isprogramirati u svom e-learningu, vi, moji online učenici, najbolji ste mi svjedoci.

Lekcija 2 ASP

Iako je riječ o ASP, ipak počnimo od njegovog roditelja IIS. Ili još više povećajmo kut gledanja kako bi nam cijela slika bila puno jasnija i kako bismo mogli vidjeti "kud plovi ovaj brod". Premda su u predviđanjima tehnologije duljim od sedmicu dana unaprijed svi suzdržani.

Panoramski pogled na ASP

Ova se knjiga u cijelosti odnosi na Internet, a Internet ima dvije strane: poslužitelja i klijenta. Evo smjera kojim se kreće Microsoftova tehnologija:



Dinamika znači stalne promjene, stalna aktivnost. Kako bi naglasio sve veću ulogu tehnologije, konkretno klijenta i poslužitelja u našim aktivnostima, Microsoft je svoj smjer razvoja nazvao aktivnom platformom, koju čine aktivni klijent i aktivni poslužitelj. Danas je taj razvoj po okomici dosegao razinu kao na slici gore.

I kao što smo vidjeli u Lekciji 1, neusporediva je razlika programirati na razini operacijskog sustava i danas, na tako visokoj razini kao što je ASP tehnologija. U primjeru smo vidjeli sljedeće:

ActiveX DLL komponentu smo napravili pomoću Visual Basic-a i znanja koje smo već imali otprije.

Nije nam bilo potrebno nikakvo znanje izvan okvira Visual Basic-a. Nismo morali programirati sučelje i komunikaciju s okolinom za našu komponentu. Sve je to umjesto nas isprogramirao Microsoft i naša je komponenta postala sukladna budućoj okolini samim činom kompajliranja, tj. našom naredbom File/Make Nagrada.dll.

ActiveX

Kad su krenuli s idejom aktivne platforme, 1996. godine se na tragu te ideje kreira tehnologija ActiveX. Aktivna platforma ima aktivnog klijenta i aktivnog poslužitelja, pa se tehnologija ActiveX odnosi i na klijenta i na poslužitelja. U tehnologiju ActiveX ulaze sljedeće tehnologije:

ActiveX Scripting – omogućava programiranje u script jezicima na strani klijenta i na strani poslužitelja. Cilj je proizvodnja dinamičkih sadržaja web stranica na strani klijenta ili na strani poslužitelja.

ActiveX Controls – omogućava dinamički download klijent komponenti koje su potrebne za podršku web stranicama na klijent strani.

ActiveX Documents – omogućava browser-ima podršku ne-HTML dokumenata.

ActiveX Server Components – omogućava sučelju Web poslužitelja rad s ostalim komponent-softwareom na poslužitelju.

COM

COM je skraćena za ideju na kojoj Microsoft temelji svoj daljnji prosperitet. Component Object Model je put k snižavanju troškova proizvodnje, održavanja i re-projektiranja web aplikacija.

Kako?

Microsoft je učio od automobilske industrije, ali i od tržišnih uvjeta općenito:

Automobilska je industrija počela s velikim firmama koje su sve dijelove proizvodile same. Tada je životni vijek jednog modela automobila bio 20 godina i više. Zašto danas te iste firme samo sastavljaju automobile, a dijelove proizvode na tisuće njihovih malih dobavljača?

Kad sebi postavite dobro pitanje, odmah ste sebi i odgovorili. Velikoj se firmi ne isplati svaki dan mijenjati proizvod. Njoj odgovara statika – dvadeset godina isti model automobila i prodaja istog modela u milijunima. Ona financijski ne može izdržati česte promjene. Maloj obiteljskoj firmi se isplati mijenjati proizvod svaki dan. Ona je to u stanju a da ne propadne, nego da normalno funkcionira.

Trend je na tržištu što češće mijenjati modele, a i civilizacija je na toj razini da su promjene svakodnevne. Banke svaki dan moraju izbaciti nove pakete kredita, industrija nove proizvode, softverske kuće nove aplikacije. Tko ne može ući u tu trku – ispada!

Knjigu koju držite u ruci tiskati će mali nakladnik, Škola E-92? Zašto neće neki veliki nakladnik? Zato što se njemu ne isplati jer će se ova knjiga u sljedećih godinu dana prodati tek u nekih 50 primjeraka, i to po najnižoj mogućoj cijeni. Koliko bi dugo živio kit kad bi dnevno jeo samo jednu srdelu? a ima puno vrsta u moru koje jako dobro žive s jednom srdelom dnevno.

Automobilska je industrija podijelila malim proizvođačima kriterije standarda koje moraju zadovoljiti, a sve ostalo je njihova briga.

Isto je učinio i Microsoft. Velike web aplikacije se grade od softverskih komponenti i softverskog okruženja. U kojem su programskom alatu napravljene

komponente ili dijelovi okruženja nije važno. Važno je da se sučelja tih komponenti i tog okruženja razumiju i mogu komunicirati bilo da su na istom računalu, bilo da su na različitim računalima širom Interneta. Microsoft je vlasnik standarda za sučelja i dijelove prepušta otvorenom javnom dobru, a svatko od nas može proizvoditi komponente koje zadovoljavaju te standarde. I mi smo proizveli već u prvoj lekciji jednu jako korisnu i iskoristivu komponentu. O ostalim prednostima COM filozofije čitajte u nastavku.

Ima dosta firma u svijetu, i u Hrvatskoj, koje žive od proizvodnje komponenti. Najbrojnije su one koje broje tek nekoliko zaposlenih. Zahvaljujući Internetu svatko u svijetu od takvih firma može naručiti komponentu. To se i događa. U takvih firma komponente naručuju i web kreatori, koji tek što su otvorili svoj kućni obrt, i General Motors, u istom danu.

U Lekciji 1 smo napravili web aplikaciju od komponente i okruženja. Komponenta je kompajlirana softverska cjelina, a okruženje je puno toga, ali recimo ono najuže - ASP. Da bismo zadovoljili COM standarde, bilo je dovoljno odabrati ActiveX DLL na početku izrade projekta u Visual Basic-u. Standardi su nakon toga briga alata u kojem radimo, u ovom slučaju Visual Basic-a. On će sve učiniti za našu komponentu, neprimjetno iza scene. Dat će joj sve što je potrebno da radi u ASP okruženju.

IIS

Internet Information Services, ili skraćeno IIS, zaštićeni je naziv Microsoftovog softvera namijenjenog računalima poslužiteljima. Međutim, budući da poslužitelji imaju svakim danom sve više posla i odgovornosti, tako im raste i količina kôda, pa onda i zahtjevi za resursima ne samo našeg računala, nego i našeg džepa. Takva je situacija na jednoj strani.

Na drugoj, klijent strani, sjedimo mi koji učimo ili smo već zaposleni, pa projektiramo, razvijamo, kodiramo. Nama treba što više znanja, a što manje troškova.

Microsoft je mislio na to, pa je tako već od verzije Windows 95 svima nama osigurao besplatnu klijent inačicu IIS-a (pogledajte: 'Koji mi je software potreban za rad po ovoj knjizi'). Te klijent inačice IIS-a mogu poslužiti veoma dobro i za manje Intranete.

World Wide Web

Upravo smo svjedoci kako se i naš poslovni svijet sve više počinje zanimati za poslovanje preko WWW. Zapadni je svijet prošao kroz pravu euforiju investiranja u Web, a i danas je taj segment još uvijek hit, samo što više vrijednosni papiri ne rastu tako nerazumno kao do sada. Zato će biti i manje bankrota.

Microsoft izlazi kasno na Internet scenu. Tek 1996. godine izlaskom Windowsa NT 4 i IIS 4, ASP 2 postaje konkurentna tehnologija za izradu dinamičkog sadržaja na poslužitelju.

Prva generacija Web aplikacija je sastavljena od statičkih web stranica nastavka .htm/.html. Ukoliko smo na stranicama imali forme preko kojih su posjetitelji mogli slati podatke u naše baze, ili dobivati dinamički sadržaj, trebalo je na poslužitelju osigurati programe koji su zadovoljavali CGI (Common Gateway Interface) protokole i standarde. CGI programi su se radili uglavnom u programskom jeziku C ili skriptnom jeziku Perl. Programer je trebao ispisati i nekoliko desetaka instrukcija, za što mu je danas potrebna samo jedna.

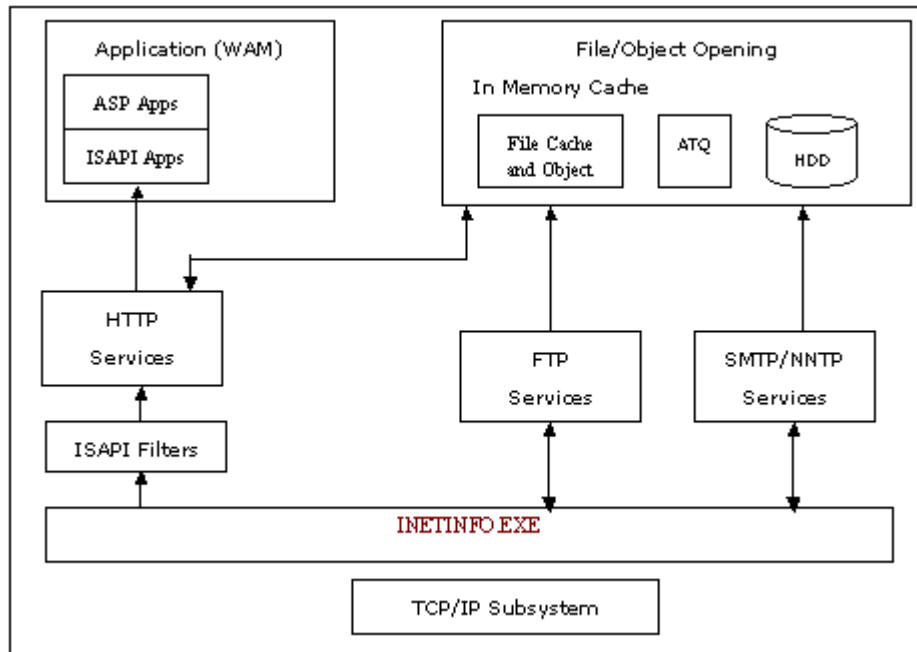
CGI tehnologiji je trebalo i više vremena za izvršavanje zahtjeva, jer se CGI program nije učitavao jednom i ostajao u radu kao današnja DLL komponenta, nego je svaki HTTP poziv značio i ponovno učitanje (i ponovno čekanje) CGI ekstenzije. To se ubrzo, s porastom korisnika Interneta, pokazalo velikim nedostatkom.

Drugu generaciju web aplikacija obilježava izlazak Web browser-a koji podržavaju download softverskih komponenti i novih skriptnih jezika kojima se programira i strana klijenta i strana poslužitelja.

CGI ekstenzije su zamijenjene ISAPI (Internet Server Application Programming Interface) ekstenzijama. ISAPI ekstenzije su DLL komponente. Microsoft je odmah osigurao ISAPI Wizarda u alatu Visual C++. Programer bi kroz nekoliko dijaloga imao gotov skelet ISAPI aplikacije.

Princip rada je isti kao DLL-ova u Windowsima. Kad prva web aplikacija pozove neku ISAPI DLL komponentu, ona se učita i radi u istom procesu u kojem i IIS, i nakon obavljenog poziva odmah je spremna za nove pozive. To CGI nije bio.

Arhitektura IIS-a 5.0:



Inetinfo.exe

je osnovni program IIS-a koji nudi mnogo Internet usluga (HTTP, FTP, SMTP, NNTP). Inetinfo.exe primi zahtjev (request) od TCP/IP i prosljedi ga prema odgovarajućoj usluzi. Internet usluga (servis) interpretira zahtjev, izvrši ga i rezultat vrati klijentu. Pošto sve ove Internet usluge "voze istom cestom", tj. rade unutar istog programskog procesa (inetinfo.exe), u stanju su dijeliti zajedničke resurse kao što su privremeno pohranjeni podaci (cached data) iz HTTP zahtjeva, informacije o računu i dnevnik (log file) podataka.

ISAPI (Internet Server Application Programming Interface)

Postoje dva tipa ISAPI DLL: ISAPI filtri i ISAPI ekstenzije. ISAPI filtri su DLL koji rade u procesu inetinfo.exe i filtriraju podatke prema i od poslužitelja. Ovi filtri se mogu koristiti za monitoring tko je logiran na naš poslužitelj, za osobnu enkripciju, osobno sažimanje podataka ili za dodatna provjeravanja korisnika. Ukoliko se aktiviraju svi ovi filtri, promet na IIS-u se znatno usporava jer svi zahtjevi moraju proći kroz njih.

ISAPI ekstenzije su također DLL-ovi koje HTTP poslužitelj može pozvati. pokretač poziva je Web browser, tj. web aplikacija.

Active Server Pages

ASP je implementiran kao ISAPI ekstenzija DLL i ne radi u istom *thread*-u s inetinfo.exe. Radi u posebnom threadu koji se odvija u posebnom, odvojenom procesu. Tako u IIS 5.1 postoje dva *procesa*. U jednom procesu je inetinfo.exe i svi DLL-ovi tako isprogramirani. U drugom procesu je ASP i ostali DLL-ovi tako isprogramirani. Taj drugi, posebni thread (i proces) u kojem je ASP naziva se *ATQ/MTS thread pool* (Asynchronous Thread Queue).

Objašnjenje pojmova *thread* i *proces*. Svaki je program struktura programskog kôda. Osnovni dio te strukture je instrukcija ili naredba. Program se izvršava tako da njegove naredbe čekaju svoj red u RAM-u kada će ih pozvati procesor i izvršiti. Rad programa je programski *proces*. Pod programskim *procesom* se smatra jedan izolirani virtualni prostor nastao dinamički. Ovdje je važna riječ *izolirani*, što znači da događanja unutar procesa neće imati utjecaja na rad ostalih aplikacija.

OS Windows upravlja procesorom – šalje ga malo jednoj, malo drugoj aplikaciji. Ti Windowsovi "djelići" se nazivaju *thread*. Proizlazi: *proces* je pojam vezan za aplikaciju sa znakom jednakosti. *Thread* je pojam vezan za Windows. Međutim, programer koji kreira program odlučuje hoće li dopustiti da se njegov program obavi u više *threadova* ili neće. Jasno, veliki programi poput Worda i Excela su svi *multi-thread*, pa zato Excel u pozadini može računati dok ja sporo tipkam u Wordu u prednjem planu. Mogućnost Windowsa da radi u thread-ovima dala mu je epitet *multi-task* ili *višezadaćnog OS-a*. Više aplikacija se za naša osjetila odvija istovremeno pa ne mogu primijetiti te *threadove*. A sad znamo da Windows postiže tu *višezadaćnost* tako da se svakom programu posveti u puno kratkih vremenskih odsječaka, *threadova*.

Kad programer projektira ISAPI DLL program, on donosi odluku i o tome hoće li taj program, kad bude pozvan, biti procesuiran u istom *threadu* kao i osnovni program inetinfo.exe ili će se za njega formirati posebni *thread*.

Kad poslužitelj (inetinfo.exe) dobije zahtjev za ASP stranicom, on zahtjev prosljeđuje ASP/MTS *thread poolu* na procesuiranje.

Cijeli ISAPI management i ASP kôd su stavljeni u komponentu nazvanu WAM (Web Application Manager - komponenta WAM.DLL). Svi ISAPI DLL-ovi su skupina aplikacija. Tu skupinu aplikacija koje pripadaju istoj tehnologiji, a pružaju različite usluge kad ih se pozove, možemo smatrati posebnim tehnološkim slojem u IIS – *slojem uslužnih aplikacija*. Sve informacije koje ulaze i izlaze iz tog sloja prolaze kroz WAM.

Po default-u, sve aplikacije u IIS 4 (NT 4) rade "in-process", a u IIS 5.0 *pooled out-of-process*.

IIS In-Process aplikacije

znači da sve aplikacije (DLL, EXE) u IIS-u rade u istom procesu u kojem i inetinfo.exe. Ako padne jedna aplikacija, pada cijeli Web poslužitelj. To je nedostatak in-processa. Prednost je što takav rad znači brži odziv poslužitelja i manje zahtjeve za resursima hardvera. Međutim, mi možemo odrediti da aplikacija radi i u svom posebnom procesu i tako povećati sigurnost cijelog poslužitelja da ne padne ako padne aplikacija. Vidi sljedeći odlomak.

IIS Out-of-Process aplikacije

znači da svaka aplikacija radi u svom izoliranom procesu. ASP joj se pridružuje također u svom posebnom procesu. Pad aplikacije nema utjecaja na ostale aplikacije. Instalacija i ponovna instalacija aplikacije ne traži i resetiranje cijelog poslužitelja. Nedostatak ovog rada su veći zahtjevi za resursima procesora.

U IIS4 pritisnemo desnu tipku miša na virtualnu mapu aplikacije, odaberemo naredbu Properties i u dijalogu uključimo opciju Run in separate memory space. U IIS5.1 bi izabrali Application protection: High (isolated).

IIS Pooled Out-of-Process aplikacije

je nova mogućnost verzije IIS 5.0. Riječ je o kompromisu. Niti su sve aplikacije u istom procesu, niti je svaka aplikacija u posebnom procesu. Postoje samo dva procesa. Sve one aplikacije koje nisu u procesu inetinfo.exe, rade u jednom zajedničkom posebnom procesu kreiranom od aplikacijskog paketa MTS/COM+.

Greške u tom posebnom "pooled" procesu neće srušiti Web poslužitelja. Pošto su samo dva procesa, nisu značajno povećani zahtjevi nad resursima. Međutim, ako padne jedna pooled aplikacija, past će sve pooled aplikacije budući da su sudionici istog procesa.

To znači, sve ugrađene ASP komponente rade u posebnom procesu. Objekti u tim komponentama su stalno pozivani iz skripta ASP stranica. Svaki poziv je jedan *thread* nad tim objektom, u toj komponenti. Između dva *threada* objekti se ne iskrcavaju iz memorije (iz procesa) jer bi to usporilo odziv. Takav način rada je nazvan *pooled-thread*.

Koji thread imaju naše COM komponente

Virtualno, dinamičko okruženje MTS (Microsoft Transaction Server) koje postoji na našem IIS vodi brigu o *threadovima* naših COM/MTS komponenti. Da bi naša komponenta bila COM sukladna, dovoljno je u VB novom projektu odabrati ActiveX DLL. VB će nakon toga automatski pridodati našoj komponenti potrebne VB ASP "library". Da bi naša komponenta bila MTS sukladna, moramo sami ručno dodati Microsoft Transaction library (Project/References).

Kada kreiramo COM komponentu, moramo joj odrediti i koji će tip *threada* imati u radu. Taj će podatak dospjeti u atribut ThreadingModel, a nakon registracije komponente dospjet će u Registry. Tamo će ga naći MTS. Postoje četiri mogućnosti.

Single-Threaded komponenta

Ukoliko komponenti ne odredimo atribut ThreadingModel, ista će po default-u biti Single-Threaded komponenta. Kad komponentu pozove prvi put neki ASP skript, ona će biti učitana u ASP okruženje među ostale ISAPI DLL. U IIS 5.1 će to biti i posebni izolirani proces ATS pooled-thread. U jednom procesu su sve ASP komponente. To znači da se u jednom trenutku može izvršavati samo jedna komponenta, točnije samo jedan njen objekt. Ostali objekti čekaju svoj red. Ovakve komponente smanjuju mogućnosti skalabilnosti web aplikacije, jer kad su objekti u jednom procesu, preostaje im samo "strpljivo" čekanje svoga reda. Tada nema mogućnosti da dobije pažnju procesora "preko reda", bez obzira što bi to bilo bolje za web aplikaciju.

Apartment-Threaded komponente

Ove komponente mogu biti odrađene i "preko reda", kad treba, pa su zato bolje za skalabilnost. Mogu to zato što se izvršavaju u svom posebnom *threadu* unutar pooled-threada. I to odjednom cijela komponenta u jednom svom izoliranom threadu (apartmanu). Web aplikacija bolje radi s apartment nego single komponentama, pogotovo kada se u toj aplikaciji u isto vrijeme zahtijeva izvršenje više objekata iz više komponenti.

Free-Threaded komponente

MTS kreira za svaku web aplikaciju jedan MTA apartment. MTA znači multi-threaded apartment. MTA apartment se nalazi kao poseban apartman unutar posebnog procesa pooled-threada. MTS u tom apartmanu kreira sve objekte iz komponenti koje su free-threaded. Zato se kreira proxy za vezu objekta u MTA i

njegove komponente koja je ostala izvan MTA. Taj proxy usporava cijelu aplikaciju.

Both-Threaded komponente

Ove komponente imaju osobine i apartment-threaded i free-threaded. Prednost je u tome što bez obzira koji se thread od ova dva odabere u procesu, objekt će uvijek biti kreiran u istom threadu u kojem je i njegova komponenta, i neće se formirati proxy veza, nego će poziv objekta uvijek biti direktni brzi poziv.

ASP

Active Server Pages je Microsoftov zaštitni naziv za tehnologiju na Web poslužitelju (IIS) koja će kreirati dinamički sadržaj web stranica i slati te stranice klijentu. U čemu je njegova najveća snaga i prestiž? U ovome:

1. Koristi sve skripte, bez obzira na proizvođača. Važno je da proizvođač skripta napravi script koji podržava COM specifikaciju. Interpreter skripta ugradimo u ASP na našem računalu, i to je sve.
2. Svaki skript koji koristimo u ASP zahvaljujući COM podršci je obogaćen sučeljem preko kojega može komunicirati s vanjskim COM sukladnim softverskim komponentama, uključujući one koji su isporučeni s ASP, one koje nudi operacijski sustav Windows i one koje smo mi kreirali ili netko treći.
3. COM sukladne komponente su aplikacije napravljene u bilo kojem programskom jeziku koji podržava COM specifikaciju.
4. Na kraju, sva ta različitost skripti, komponenti kao i njihova fizička rasutost po istom ili različitim računalima, uvijek šalje klijentu čisti HTML. Znači i klijent može biti bilo koji, jer svaki na svijetu podržava HTML.

Kad smo već kod "rasutosti" jedne web aplikacije po Internetu (ili Intranetu), spomenimo i izraz *distribuirane aplikacije*. Svaka web aplikacija je distribuirana, jer svaki klijent koji uputi zahtjev prema poslužitelju postaje dio web aplikacije. Distribuirana web aplikacija mora komunicirati s klijentom i isporučiti mu ono što taj klijent podržava. ASP isporučuje klijentima HTML.

Na osnovu navedene teorije i primjera u Lekciji 1, imamo sljedeću situaciju kada je o ASP-u riječ:

Iako je ASP Microsoftova tehnologija, glavninu programerskog posla možemo i dalje raditi u našim omiljenim programskim alatima koji ne moraju biti Microsoftovi, od skripta pa do programskog jezika za komponente.

Već u našem prvom primjeru u Lekciji 1, naš VBScript na ASP stranici je komunicirao s ugrađenim ASP komponentama Response, Request, Server i našom komponentom Nagrada.dll. Rezultat te ASP stranice je na kraju ispao čisti HTML. HTML je poslan klijentu i zato je sasvim svejedno je li klijent Internet Explorer, Netscape ili Opera ili bilo koji drugi. Svi klijenti znaju "parse" (složiti na ekranu) HTML dokument.

Izrada ASP stranice

ASP stranice su HTML stranice. Dovoljno je našem HTML dokumentu obrisati nastavak .htm i tipkati nastavak .asp pa da ga procesuiru ASP. Ukoliko na takvoj ASP stranici nema skripta za ASP, ona će biti poslana klijentu nepromijenjena.

ASP stranici umjesto .htm ili .html, dakle, dodajemo nastavak .asp i kreiramo je pomoću HTML tagova, skript programiranja za poslužitelja i običnog teksta. Stranice s .asp nastavkom i stranice s .htm ili .html nastavkom su tekstualne stranice. Za klijent stranu je potpuno isto svejedno traži li stranicu s nastavkom .asp ili s nastavkom .htm.

Klijentu će se uvijek vratiti čisti HTML dokument bez skripta. Na poslužitelju su fizički pohranjene i .htm i .asp stranice. Tu su one različite, iako obje tekstualne, .htm nema skripta za poslužitelja, a .asp stranica ima skript za poslužitelja.

Kada klijent pozove .htm stranicu, on poziva njenu virtualnu adresu. IIS zna vezu između virtualne i fizičke adrese i vraća klijentu presliku nepromijenjene originalne .htm stranice. Kada klijent pozove .asp stranicu, IIS prosljeđuje stranicu ASP-u. ASP procesuiru skripte koji su za njega, a rezultat ugradi u web stranicu koju će vratiti klijentu.

Samo pozivanjem .asp stranice iz Web browser-a preko njene virtualne adrese dat će rezultat.

Virtualna se adresa izgradi ovako. ASP stranice smo pohranili u fizičku mapu na disku. Pomoću Wizarda u IIS kreiramo virtualnu mapu. Jedan od dijaloga će nas pitati za fizički put mape nad kojom kreiramo ovu virtualnu mapu. I tu vezu je IIS zapamtio. Moramo virtualnoj mapi uključiti opciju Script or Execute permission.

ASP stranicu možemo pisati u bilo kojem tekst procesoru, počevši od Notepad-a. Puno je bolje to raditi u tekst procesorima koji nam pomažu automatskim bojanjem ključnih izraza ili u sofisticiranom Microsoft® Visual InterDev™, koji je član Visual Studia.

Programiranje pomoću skripta Procedura Function

ASP se isporučuje s ugrađenim interpreterima za VBScript i Jscript. VBScript je pod grupa Visual Basic-a. Ponekad ćete i zaboraviti što tipkate jer su izrazi i sintaksa slični. Možete koristiti i bilo koji skript jezik ako ste za njega ugradili interpreter. Na istoj *.asp* stranici možemo koristiti i različite skripte.

U ovoj će knjizi koristiti VBScript i tako ostati u poznatom okruženju porodice Visual Basic.

Skript se odvaja od ostalog teksta u HTML dokumentu pomoću delimitera `<% i %>`.

U primjeru Nagrada, u Lekciji 1, koristili smo komponentu za izračunavanje nagrade zaposlenika. Napišimo sada ASP stranicu koja će izračunati nagradu bez pomoći komponente.

Tipkajte najprije HTML kôd stranice na kojoj je forma. Nazovite ga `nagrada_sub.htm`. Sve je isto kao u primjeru u prvoj lekciji, osim što ovdje forma poziva drugu *.asp* stranicu:

```
<HTML><HEAD><TITLE>Procedure i ASP</TITLE>
</HEAD>
<BODY>
<FORM action=izracunajnagradu_sub.asp method=post>
<BR>Iznos ukupne prodaje: <INPUT
```

```
size=30 name=iznosukupneprodaje> <BR>Broj ukupno  
izdanih racuna: <INPUT  
  
name=brojukupnoizdanihracuna>  
  
<P><INPUT type=submit value="Nagrada iznosi" name="">  
<INPUT type=reset value="Obrisi sve" name="">  
</FORM></P></BODY></HTML>
```

A sada tipkajte ASP stranicu izracunajnagradu_sub.asp

```
<HTML>  
  
<HEAD>  
  
<TITLE>Procedura i ASP</TITLE>  
  
</HEAD>  
  
<BODY>  
  
<%  
dim v1,v2  
Function Izracunaj(v1,v2)  
Izracunaj=v1*0.04+(v1/v2)*0.01  
End function  
  
'Uzimanje vrijednosti iz forme  
ukupnaProdaja = Request.Form("iznosukupneprodaje")  
ukupnoRacuna = Request.Form("brojukupnoizdanihracuna")  
  
>  
  
Ukupna prodaja: <% = FormatCurrency(ukupnaProdaja) %>  
<BR>  
  
Ukupno racuna: <% = ukupnoRacuna %> <br>  
  
zarada: <% = izracunaj(ukupnaprodaja,ukupnoracuna)  
><P>
```

```

<%
'Kreiranje instance nase nove komponente Nagrada.dll

'Set iznosNagrade =
Server.CreateObject("Nagrada.VrijednostNagrade")

'Dobijanje rezultata i prikazivanje istog na web
stranici

'iznos = iznosNagrade.Nagrada(csng(ukupnaProdaja),
cint(ukupnoRacuna))

'Response.Write("Nagrada iznosi: " +
FormatCurrency(iznos))

%>

</BODY>

</HTML>

```

Analiza

Ovdje se susrećemo s dvije novosti. Deklarirali smo varijable...

```
dim v1,v2
```

...i napisali proceduru Function koja je zamijenila komponentu. Na isti se način kreira i procedura Sub. Razlika je samo u početnoj i zadnjoj instrukciji: *sub....end sub* i *function....end function*. U VBScriptu je svejedno koristimo li mala ili velika slova, kao i koristimo li razmaknice ili ne. Kako vas je volja. S razmaknicama je preglednije, ali sve je to osobna stvar programera.

Na primjer, možemo pisati bez razmaknica, ovako...

```
Izracunaj=v1*0.04+(v1/v2)*0.01
```

... ili s razmaknicama, ovako..

```
Izracunaj = v1 * 0.04 + (v1/v2) * 0.01
```

Nastavljamo s analizom kôda...

```
Function Izracunaj(v1,v2)
```



```
Izracunaj=v1*0.04+(v1/v2)*0.01
```

```
End function
```

Kod deklaracije varijabli nema onog nastavka ... As Integer... i slično o tipu varijable. U VBScriptu su sve varijable uvijek *variant*. Moguće deklaracije su *Dim*, *Public* i *Private*.

Na mjestu na kojem smo deklarirali varijable, odredili smo njihov vijek: globalno, tj. dok je stranica aktivna. Mogli smo ih deklarirati i unutar funkcije pa bi im vijek bio koliki je i vijek funkcije.

Vrijednosti koje se iz forme prenesu u ove varijable v1 i v2 su prošle "preko ruku" više varijabli ByVal (preko vrijednosti varijabli), a ne ByRef (preko referencija varijabli). Vrijednosti su uvijek iste, ali imena varijabli kroz koje prolaze te vrijednosti ne moraju biti ista.

VBScript na Internetu:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsstmsub.asp>

Lokalizacija

U gornjem se primjeru pojavljuju nelokalizirani formati valute. Evo kako se lokaliziraju formati pomoću ASP.

Primjer ASP stranice s različitim lokalizmima. Otvorite IIS, zatim otvorite ASP stranicu čija je virtualna adresa: <http://localhost/Kartelo/Lekcija2/intform1.asp>

```
<%@ Language=VBScript %>

<html>

<head>

<title>Internacionalni formati</title>

</head>

<body>

<font face="Arial Unicode MS">
```

<H3>Hvala što ste kupovali kod nas. Tiskajte svoj
račun.</H3>

<%

'Kreirajmo varijable.

dim saveLocale

dim totalBill

'Zadajmo varijable.

saveLocale = GetLocale

totalBill = CCur(85.50)

'Za svaki lokal formatiraj datum i valutu.

SetLocale("fr")

Response.Write"Ovako pišu u
Francuskoj:
"

Response.Write FormatDateTime(Date, 1) & "
"

Response.Write FormatCurrency(totalBill) & "
"

SetLocale("de")

Response.Write"Ovako pišu u
Njemačkoj:
"

Response.Write FormatDateTime(Date, 1) & "
"

Response.Write FormatCurrency(totalBill) & "
"

SetLocale("en-au")

Response.Write"Ovako pišu u Engleskoj i
Australiji:
"

Response.Write FormatDateTime(Date, 1)& "
"

Response.Write FormatCurrency(totalBill) & "
"

```

SetLocale("hr")

Response.Write"<B>Ovako pi&#353u u
Hrvatskoj:</B><BR>"

Response.Write FormatDateTime(Date, 1)& "<BR>"

Response.Write FormatCurrency(totalBill) & "<BR>"

'Vrati na lokalne formate

SetLocale(saveLocale)

%>

</font>

</body>

</html>

```

Napomena. Kada želimo da na našim web stranicama svi vide naša slova, i oni u Kini i oni u Americi, pišimo u HTML kôdu naših stranica sljedeće kôdove:

Slovo	Pisati u HTML	Slovo	Pisati u HTML	Slovo	Pisati u HTML
Ć	Ć	ć	ć	Č	Č
č	č	Đ	Đ	đ	đ
Š	Š	š	š	Ž	Ž
ž	ž				

Dekadski broj je iz Unicode standarda http://www.alanwood.net/unicode/latin_extended_a.html a u HTML ispred toga broja moramo dodati još znakove: &#

Ako pišete tehničke stranice, Unicode za grčka slova:

<http://www.alanwood.net/unicode/greek.html>

Ispred dekadskih brojeva koje nađete u tablici Unicode, u HTML-u još dodajte kao gore: &#

Lekcija 3 – Ugrađeni ASP objekti

ASP direktive

Postoje direktive koje određuju izlaznu vrijednost i one koje određuju procesuiranje.

Direktive za izlazne vrijednosti. U primjeru 'nagrada' u Lekciji 1 imali smo naredbe u kojima se izlazna vrijednost definirala pomoću objekta Response...

```
Response.Write("Nagrada iznosi: " +  
FormatCurrency(iznos))
```

...i pomoću direktive `<% = izraz %>`...

```
Ukupna prodaja: <% = FormatCurrency(ukupnaProdaja) %>  
<BR>
```

```
Ukupno racuna: <% = ukupnoRacuna %> <P>
```

Rezultat je isti s Response i s izlaznom direktivom, pa smo umjesto Response mogli napisati...

```
Nagrada iznosi <%= FormatCurrency(iznos) %>
```

Naredba koja definira procesuiranje

glasi, na primjer...

```
<%@ LANGUAGE=VBScript %>
```

Uvijek se piše na početku ASP dokumenta. Mora biti razmaknica između 'et' (@) i ključnog izraza direktive. Gornja naredba govori ASP-u koji će interpreter koristiti (VBScript). Znak 'et' govori ASP-u da je interpretacija skripta njegov posao, a ne posao klijenta. Ukoliko ima više naredbi, sve se pišu unutar istih delimitera:

```
<%@ LANGUAGE="JScript" CODEPAGE="932" %>
```

Sve direktive su direktive za ASP na poslužitelju, a moguće su ove direktive:

LANGUAGE – kojeg interpretera pozvati (VBScript, JavaScript, Jscript i sl.).

ENABLESESSIONSTATE - koristi li ASP stranica Session.

CODEPAGE – po kojoj kodnoj stranici dekodirati znakove na ASP stranici.

LCID – o kojem se lokalizmu radi.

TRANSACTION – ASP stranica radi unutar "transaction context" - a.

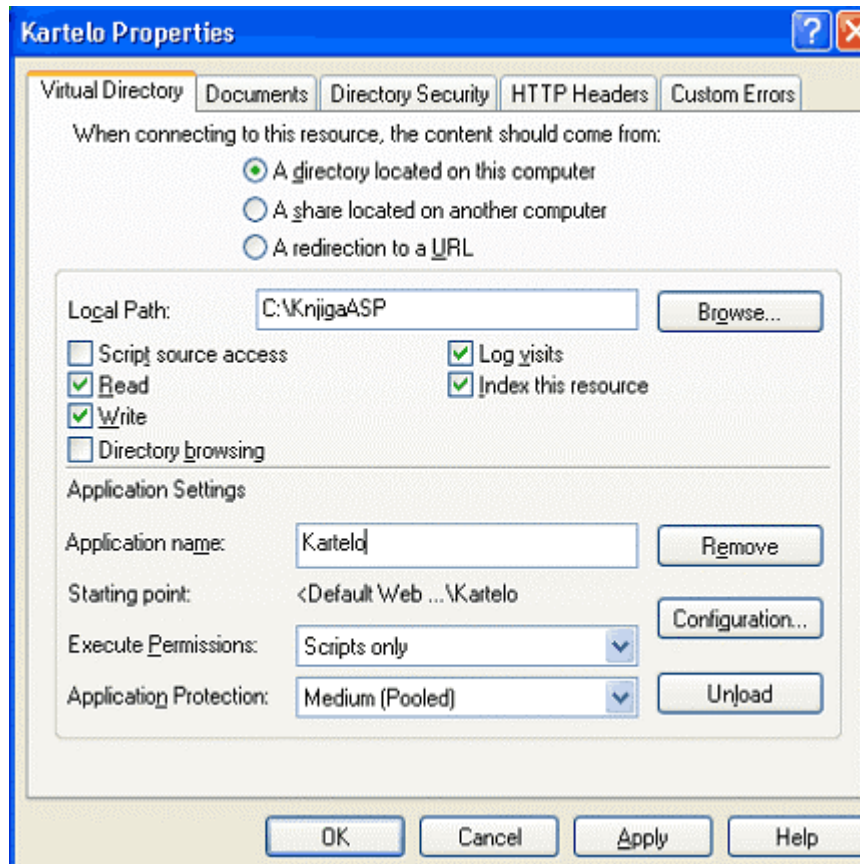
Web aplikacija

Počnimo od web site-a. Što je web site? Web site je virtualna mapa i sadržaj te virtualne mape. Virtualnu mapu kreira administrator IIS-a. Sada ste to vi i već ste u ovoj knjizi kreirali nekoliko virtualnih mapa. Unutar virtualne mape je sadržaj našeg web site-a. Svaka virtualna mapa je "alias" točno određene fizičke mape. Proces kreiranja virtualne mape Microsoft uspoređuje s kreiranjem imena jednog virtualnog prostora. Ukoliko u virtualnoj mapi imamo strukturu podmapa, sve su te podmape automatski postale i virtualne.

Da bi naša virtualna mapa postala i Web aplikacija, moramo "prostoru" dati još jedno ime, ime aplikacije, ovako:

1. Otvorite IIS.
2. Pritisnite desnu tipku miša na virtualnu mapu 'Kartelo', pa odaberite naredbu Properties u skraćenom izborniku.

3. U polje Application name tipkajte ime aplikacije po želji, na primjer Kartelo.



4. OK.

To je sve. Naša web aplikacija ima "root" (korijen) mapu Kartelo, koja je "alias" fizičke mape KnjigaASP na disku C. Sve naše web stranice koje ćemo pohranjivati u mapu Kartelo i njene podmape pripadaju web aplikaciji Kartelo. Postoji samo jedna ASP stranica za koju postoji točno pravilo gdje mora biti pohranjena i kako se mora zvati. To je ASP stranica Global.asa i mora biti pohranjena u korijen mapi. Nju ne moramo obvezno kreirati. Ali ako je kreiramo, ime i lokacija joj se znaju.

Naša web aplikacija komunicira s vanjskim svijetom, koji najčešće čine ugrađene ASP komponente, naše osobne komponente, baze podataka. Ti elementi nisu unutar mape Kartelo i mogu biti fizički bilo gdje na računalu ili mreži (znači mogu biti i u mapi Kartelo).

Built-in ASP Objects

ASP ima ugrađene objekte pomoću kojih 'hvata' podatke koje posjetitelji unose u forme, prati putovanje svakog posjetitelja, broji koliko posjetitelja ukupno trenutno ima na web site-u itd. Neke od tih objekata već smo koristili, a evo popisa i kratkog opisa svih:

Objekt Application

Kada želimo nešto pohraniti i koristiti na razni naše web aplikacije, onda ćemo posegnuti za ovim objektom. On će nam pomoći ukoliko želimo na našim stranicama imati podatak o tome koliko trenutno posjetitelja imamo.

Objekt Application sadrži:

Kolekcije: StaticObjects, Contents

Kolekcija Contents ima metode: Remove, RemoveAll

Metode: Lock, Unlock

Događaje: Application_OnEnd i Application_OnStart. Skripti za ove događaje se deklariraju u ASP stranici Global.asa.

Objekt Request

Objekt Request memorira sve informacije pristigle u HTTP zahtjevu, i naše ASP stranice mogu koristiti te informacije. Pomoću metoda GET i POST podaci koje posjetitelj unese u forme su dostupni parametri objektu Request. Cookies i certifikati također, kao i binarni podaci poslani poslužitelju, kao što je npr. "file uploads".

Objekt Request ima:

Kolekcije: ClientCertificate, Cookies, Form, QueryString, ServerVariables

Properties: TotalBytes

Metode: BinaryRead

Objekt Response

Ovaj objekt šalje informacije izravno browser-u, mijenja smjer navigacije (redirecting) i daje nove vrijednosti cookie-ma.

Objekt Response ima:

Kolekciju: Cookies

Properties: Buffer, CacheControl, Charset, CodePage, ContentType, Expires, ExpiresAbsolute, IsClientConnected, LCID, Pics, Status,

Metode: AddHeader, AppendToLog, BinaryWrite, Clear, End, Flush, Redirect, Write

Objekt Server

Objekt Server ćemo najviše koristiti za kreiranje instancije komponente COM (**Server.CreateObject**). Ovaj objekt ima:

Properties: ScriptTimeout

Metode: CreateObject, Execute, GetLastError, HTML Encode, MapPath, Transfer, URLEncode

Objekt Session

Objekt Session koristimo za pohranjivanje informacija o "session". Session je posjeta. Objekti koje smo kreirali u objektu Session žive toliko dugo koliko je posjetitelj u našoj web aplikaciji, plus vrijeme koje smo namjestili (predzadano 20 minuta). Jasno, u programiranju naše web aplikacije mi možemo upravljati sa životom tih objekata po našoj volji, kao i s vremenom mirovanja nakon kojega će objekt Session "*umrijeti*". Objekt sadrži:

Kolekcije: Contents, StaticObjects

Properties: CodePage, LCID, SessionID, Timeout

Metode: Abandon, Contents.Remove, Contents.RemoveAll

Događaji. Skripte za sljedeće događaje deklariramo na stranici Global.asa:
Session_OnEnd, Session_OnStart

ObjektObjectContext

Pomoću ovog objekta potvrđujemo ili poništavamo transakciju koju je pokrenuo neki ASP skript.

ObjektASPErrror

Objekt ASPError koristimo kada želimo uhvatiti jednu ASP grešku i korisniku vratiti sadržajni opis te greške.

Primjeri uporabe ugrađenih objekata

Primjer 1: Metoda Server.HTMLEncode

Online učitelj u svojim web stranicama mora prikazati retke poput ovoga:

Ukupna prodaja: <% = FormatCurrency(ukupnaProdaja) %>

To se postiže pomoću Server metode HTMLEncode, a sintaksa je:

Ukupna prodaja: <% = Server.HTMLEncode(" <% =
FormatCurrency(ukupnaProdaja) %\>")%>

Morali smo pisati %\> za dobiti pravilan rezultat. Da smo pisali %> na tom mjestu, ASP bi bio zbunjen jer mu nedostaje znak otvaranja <% , pa bi izbacio grešku.

Primjer se nalazi u mapi *Lekcija 3*, a virtualna mu je adresa:

Pozovite najprije: <http://localhost/Kartelo/Lekcija3/HTMLEncode.htm>

Nakon pritiska na gumb 'Nagrada iznosi' otvara se ASP stranica u kojoj koristim metodu HTMLEncode:

<http://localhost/Kartelo/Lekcija3/HTMLEncode.asp>

i na njoj je prikazano:

Ukupna prodaja: <% = FormatCurrency(ukupnaProdaja) %>

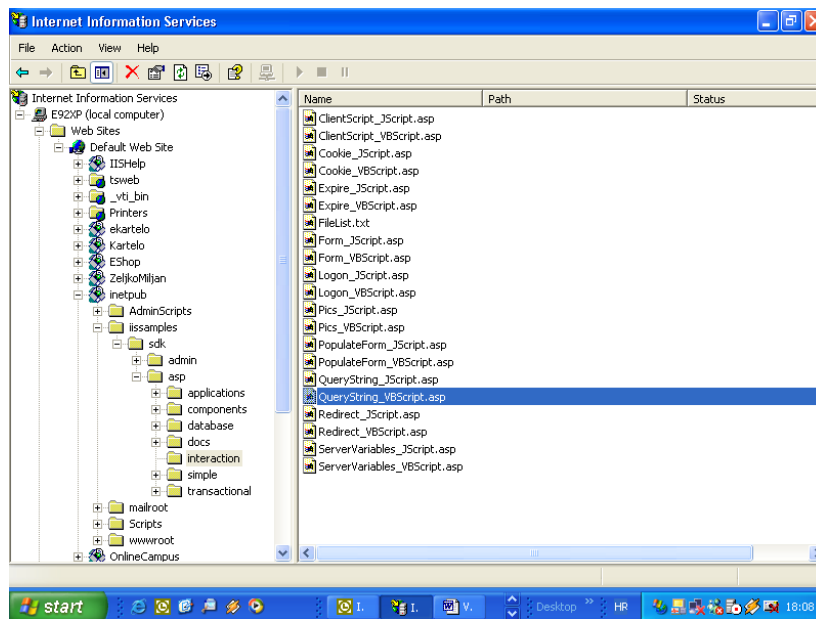
U Source pogledu vidite ovo:

Ukupna prodaja: <% = FormatCurrency(ukupnaProdaja) %>

Umjesto znaka < HTMLencode je napisao < što IE "parse" kao <. Isto je učinio za znak > (zamijenio sa %> što IE "parse" kao >)

Primjer 2: Metoda Get i kolekcija Request.QueryString

1. U prozoru IIS 5.1 (Windows XP Professional) otvorite mapu "interaction" koja se nalazi u strukturi podmapa mape "inetpub".



2. Pritisnite desnu tipku miša na ASP stranicu `QueryString_VBScript.asp`, pa odaberite naredbu Browse.
3. U IE se otvorila ova adresa:
http://localhost/inetpub/iissamples/sdk/asp/interaction/QueryString_VBScript.asp
4. Tipkajte svoje ime i prezime u polja First Name i Last Name. Pritisnite Submit i na dnu će se ispisati vaše ime i prezime.

Analiza ASP stranice `QueryString_VBScript.asp`

HTML kôd forme je...

```
<FORM NAME=Form1 METHOD=GET
ACTION="QueryString_VBScript.asp">

    <P>First Name: <INPUT TYPE=Text NAME=fname>

    <P>Last Name: <INPUT TYPE=Text NAME=lname></P>

    <INPUT TYPE=Submit VALUE="Submit">

</FORM>
```

... i iz tog HTML-a vidimo sljedeće. Metoda kojom će podaci biti otpremljeni poslužitelju je GET. Kada pritisnemo gumb Submit, atribut forme ACTION određuje što će se dogoditi. Događiti će se poziv iste ASP stranice na kojoj se nalazi i ova forma, ASP stranice `QueryString_VBScript.asp`.

Ispis na dnu web stranice na zaslonu klijenta je došao iz memorije na poslužitelju u kolekciji `Request.QueryString("fname")`, preko metode `Response.Write`.

```
<% Response.Write(Request.QueryString("fname"))%>
<BR>

<% Response.Write(Request.QueryString("lname"))%>
```

Pogledajmo sada polje Address u browser-u:

```
http://localhost/inetpub/iissamples/sdk/asp/interaction/
QueryString_VBScript.asp?fname=Ivica&lname=Kartelo
```

To je osnovna osobina metode GET. HTTP zahtjev je vidljiv u polju Address browser-a. Postoji i ograničenje na 256 znakova. Takav ispis često vidimo na tražilicama. Iza imena ASP stranice slijedi upitnik, pa ime kontrole u formi (`fname`). Dalje slijedi znak jednakosti, zatim ono što je tipkao posjetitelj, znak `&`, onda ime druge kontrole u formi, znak jednakosti, i na kraju ono što je posjetitelj unio u drugu kontrolu na formi. Takva je sintaksa razumljiva kolekciji `Request.QueryString`, koja je "uhvatila" samo ono što je posjetitelj unio, i to pohranila u memoriju poslužitelja.

Primjer 3: Metoda Post i kolekcija Request.Form

1. Iz prozora IIS otvorite ASP stranicu `Form_VBScript.asp` i u Address od IE će pisati URL:
http://localhost/inetpub/iissamples/sdk/asp/interaction/Form_VBScript.asp
2. Ispunite HTML formu i pritisnite Submit. Na dnu se, kao i u primjeru otprije, pojavljuje vaše ime i prezime.

Analiza

Ovaj put je u polju Address od browser-a ispis isti kao i prije pritiskanja gumba Submit:

http://localhost/inetpub/iissamples/sdk/asp/interaction/Form_VBScript.asp

Pogledajmo HTML stranice `Form_VBScript.asp`...

```
<FORM NAME=Form1 METHOD=Post
ACTION="Form_VBScript.asp">

    First Name: <INPUT TYPE=Text NAME=fname><P>

    Last Name: <INPUT TYPE=Text NAME=lname><P>

    <INPUT TYPE=Submit VALUE="Submit">

</FORM>

<HR>

<% Response.Write Request.form("fname")%> <BR>

<% Response.Write Request.form("lname")%> <BR>
```

Ovaj se primjer od prethodnoga razlikuje samo u metodi slanja HTTP zahtjeva poslužitelju. Sada je uporabljena metoda Post. Njene su osobine neograničen broj znakova i tajnost podataka. Naš se upit sada nije vidio u polju Address.

Preporučuje se odmah, ispod naredbe, prije početka skripta, navesti:

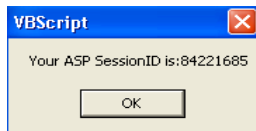
```
<% Option Explicit %>
```

Primjer 4: Properties Session.SessionID

1. Iz prozora IIS otvorite ASP stranicu ClientScript_VBScript.asp i u Address od IE će pisati URL:

http://localhost/inetpub/iissamples/sdk/asp/interaction/ClientScript_VBScript.asp

2. Pritisnite gumb i dobit ćete poruku na zaslonu:



Analiza

Nakon ASP naredbe ...

```
<%@ LANGUAGE = VBScript %>
```

...i naredbe koju ne bi trebalo zaboravljati...

```
<% Option Explicit %>
```

... slijedi definicija skripta za klijent stranu. Kôdirana je procedura Sub koja će se izvršiti na klijent strani. U proceduri se poziva VBScript funkcija MsgBox. U argumentu te funkcije se nalazi tekst poruke u kojem je i naredba

```
<%= Session.SessionID %>
```

Naredba ima za cilj umetnuti u tekst broj svojstva SessionID. Ovdje je zanimljivo da se ova naredba izvršava dinamički tek kada posjetitelj pritisne gumb i pokrene funkciju MsgBox.

```
<SCRIPT LANGUAGE = VBScript>
```

```
Sub Doit()
```

```
    MsgBox "Your ASP SessionID is:" & <%=  
Session.SessionID %>
```

```
End Sub
```

```
</SCRIPT>
```

Sav gore navedeni kôd učitat će se prije prvog HTML taga i tako biti spreman kada ga pozovemo pritiskom na gumb. Slijedi HTML kôd...

```
<HTML>

  <HEAD>

    <TITLE>Client-Side Scripting</TITLE>

  </HEAD>

  <BODY BGCOLOR="White" TOPMARGIN="10"
LEFTMARGIN="10">

    <FONT SIZE="4" FACE="ARIAL, HELVETICA">

    <B>Client-Side Scripting</B></FONT><P>
```

... a na kraju slijedi samo gumb, ne i forma. Pogledajte kako je tom gumbu pridružen klijent skript u kojem događaj ONCLICK poziva klijent funkciju Doit ...

```
    <INPUT TYPE=Button VALUE="Click Here"
ONCLICK=Doit>

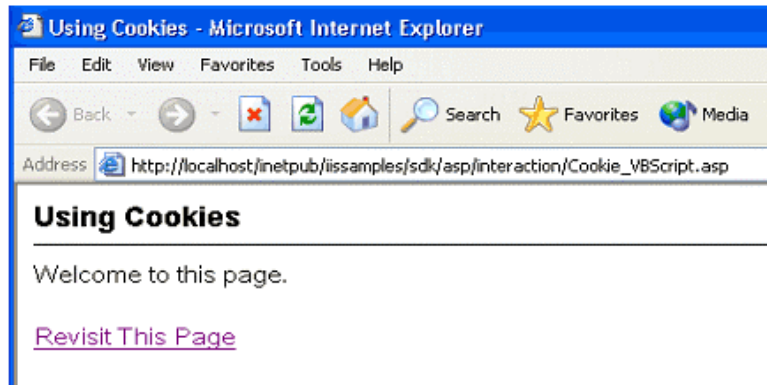
  </BODY>

</HTML>
```

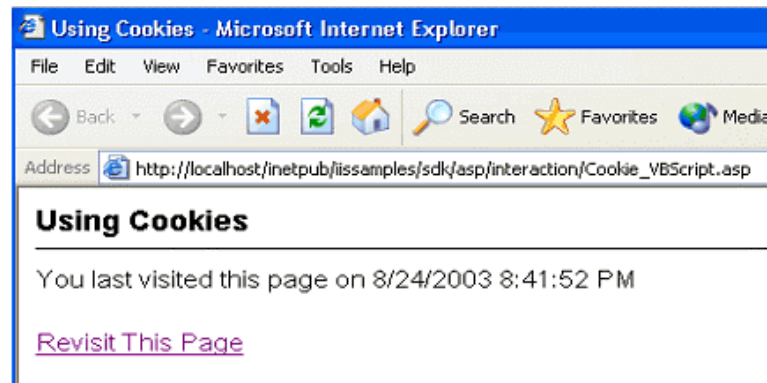
Primjer 5: Kolekcije Request.Cookies i Response.Cookies

3. Iz prozora IIS otvorite ASP stranicu `Cookie_VBScript.asp` i u Address od IE će pisati URL:

http://localhost/inetpub/iissamples/sdk/asp/interaction/Cookie_VBScript.asp



4. Pritisnite link Revisit This Page...



Analiza

```
<%@ LANGUAGE = VBScript %>
```

```
<% Option Explicit %>
```

Prije bilo kojih podataka u HTTP zaglavlju, objekt Response određuje da stranica neće biti pohranjena u "cache" memoriji na strani klijenta. Budući da su ASP stranice svaki put drukčije (ASP kreira dinamički sadržaj), nema ih smisla pohranjivati u cache memoriju. Svojstvo Expires objekta Response određuje koliko će dugo stranica biti u cache memoriji. Kada napišemo ...

```
<%
```

```
Response.Expires = 0
```


... naredili smo "ne memoriranje stranice u cache memoriji klijenta"; to znači da će na svaki poziv klijentu stići svježa stranica s poslužitelja. Deklariramo varijablu u kojoj ćemo sačuvati podatke iz kolekcije Request.Cookies...

```
Dim LastVisitCookie
```

... i pridodajemo joj vrijednost CookieVBScript iz kolekcije Request.Cookies...

```
LastVisitCookie = Request.Cookies("CookieVBScript")
```

Samo pomoću objekta Response možemo mijenjati vrijednost članova kolekcije Request.Cookies, kojem sada pridružujemo vrijednost datuma na klijent računalu...

```
Response.Cookies("CookieVBScript") =  
FormatDateTime(NOW)
```

```
%>
```

Dakle, u gornjem skriptu, prije nego je počelo učitavanje HTML tagova, objekt Response je promijenio kolekciju Request.Cookies i tako joj dodao funkciju datuma.

Počinja "parse" HTML dokumenta u klijentu...

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Using Cookies</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="White" TOPMARGIN="10"  
LEFTMARGIN="10">
```

```
<FONT SIZE="4" FACE="ARIAL, HELVETICA">
```

```
<B>Using Cookies</B></FONT><BR>
```

```
<HR SIZE="1" COLOR="#000000">
```

Slijedi If ... Else ... End If. Ako je varijabla prazna, ovom posjetitelju je to prva posjeta pa mu šaljem poruku dobrodošlice. Ako nije prazna, ispišimo mu datum i vrijeme zadnje posjete...

```
<%  
    If (LastVisitCookie = "") Then  
        Response.Write("Welcome to this page.")  
    Else  
        Response.Write("You last visited this page on "  
+ LastVisitCookie)  
    End If  
%>  
  
    <P><A HREF="Cookie_VBScript.asp">Revisit This  
Page</A>  
  
</BODY>  
  
</HTML>
```

Lekcija 4 COM komponente

Kada kažemo COM komponente, onda mislimo na programske datoteke koje podržavaju COM standarde. Uz ASP, COM komponente igraju najvažniju ulogu u razvoju web aplikacija. Ima zadataka koje i ne možemo isprogramirati u VBScriptu ili bilo kojem skriptu programskog jezika. Tada su rješenje COM komponente. Možemo ih kreirati u Visual Basic-u, C, C++, Javi i mnogim skript jezicima.

Komponente su "crne kutije" i nitko ne može vidjeti naš kôd, za razliku od ASP skript kôda. Komponente su kompajlirane pa se brže izvršavaju od ASP kôda, koji se treba, nakon svakog zahtjeva, uvijek iznova kompajlirati.

ASP dolazi s mnogo ugrađenih komponenti (najpoznatija i najznačajnija među njima je ADO), a Internet je pun trećih strana koje nude gotove komponente ili izradu novih prema vašim potrebama.

U ovoj knjizi gradimo detaljan profesionalni projekt, danas najčešći na Internetu, e-commerce (e-knjižara Kartelo), upravo s COM komponentama programiranim u Visual Basic-u 6.

COM komponente se mogu uvijek iznova koristiti, mijenjati, nadograđivati. Biznis logika, koja je uvijek između prezentacijskog sloja (HTML dokumenata) i sloja u kojem su pohranjeni podaci (baza podataka), najčešće se (i to je najbolje rješenje) programira u COM komponentama.

COM komponente ne mogu imati sučelje vidljivo klijentu, MsgBox i sl. Sučelje komponenti je vidljivo samo poslužitelju.

Granularity komponente

Svaka komponenta, bila ona općenito COM ili COM/MTS, ima svojstvo *granularity*. To se svojstvo odnosi na broj radnji koje obavlja jedna komponenta. Što komponenta obavlja manje radnji, to ima finiji *granularity* i obrnuto; više radnji, grublji *granularity*. Komponente koje obavljaju provjeru kupaca u našem primjeru su grublje, a komponente koje brinu o katalogu su finije.

Visual Basic projekt ima jednu ili više class formi. Jedna class forma jednako je jedan objekt. Na class formi može biti više metoda (Function procedura). Svaka metoda obavlja neku radnju. Puno metoda, komponenta ima grublji *granularity*, i obrnuto.

Što je bolje? Bolje je praviti komponente s finijim *granularity* jer ih je lakše debugirati, ponovo uporabiti, a i u radu brže obave svoj posao i oslobode zauzeća za druge threadove. E-knjižaru u našem primjeru je moguće napraviti i samo s jednom komponentom, i to može izgledati privlačno – cijela biznis logika na jednom mjestu, ali iz gornjega vidimo da je to lošija varijanta.

Kada se cijela biznis logika stavi u samo jednu komponentu, takav koncept distribuirane aplikacije nazivamo *komponentnocentričnim*. Kada koristimo više komponenti a vodilja nam je ergonomičnost kupca, onda je to *kupcocentričan* koncept. Naš primjer e-knjižare Kartelo je kupcocentrična web aplikacija.

Kreiranje instancije objekata koji postoje u komponentama

COM komponenta je programski kôd kompajliran u DLL ili EXE datoteku. Unutar komponente postoji jedan ili više *objekata*. Objekti su CLASS forme u VB projektu. Svaki objekt ima metode (programirane procedure) i properties (attribute). ASP skript koristi objekt u COM komponenti na sljedeći način:

1. Pomoću metode *Server.CreateObject* ili pomoću HTML taga <OBJECT> se najprije kreira instancija objekta. Za ovu naredbu je potrebno znati registrirano ime komponente (PROGID). Pomoću ASP skript jezika kreiramo instanciju objekta i dodijelimo joj ime. Tako je instancija objekta u ASP skriptu dobila svoje ime.

Primjer kreiranja instancije objekta komponente Ad Rotator čiji je PROGID MSWC.AdRotator. Naredba u VBScript-u glasi:

```
<% Set reklama = Server.CreateObject("MSWC.AdRotator") %>
```

Kreiranje instancije istog objekta pomoću HTML taga <OBJECT> glasi ovako:

```
<OBJECT RUNAT=Server ID=reklama  
PROGID="MSWC.AdRotator"></OBJECT>
```

ili pomoću registracijskog broja komponente CLASSID:

```
<OBJECT RUNAT=SERVER ID=MyAds  
CLASSID="Clsid:1621F7C0-60AC-11CF-9427-  
444553540000"></OBJECT>
```

Napomena: ASP podržava Windows Script Components, Microsoftovu scripting tehnologiju pomoću koje se također mogu kreirati komponente. Pogledajte Help u IIS.

Ugrađeni ASP objekti

ASP ima ugrađene objekte kao što su Request, Response i drugi objašnjeni ranije u knjizi. ASP skript u našim ASP stranicama ne mora kreirati instanciju tih objekata. Ti se objekti automatski kreiraju čim počne procesuiranje ASP stranice. Naš skript odmah može pristupiti metodama i properties-ima tih objekata. Na primjer, napisati nešto na web stranici pomoću metode Write ugrađenog objekta Response možemo ovako:

```
<% Response.Write "Dobar dan" %>
```

Na isti način se pristupa i properties-ima objekata:

```
<%= reklama.Border=0 %>
```

Život (scope) objekta

"Scope" ili "život" jednog objekta određuje koji će sve skript u našoj distribuiranoj aplikaciji moći koristiti taj objekt. Scope je vrijeme koje objekt provodi u RAM-u poslužitelja.

ASP stranica "živi" (u RAM-u poslužitelja) toliko koliko je potrebno ASP-u da je procesuirana. Predzadano je (default): Instancija objekta koju smo kreirali na ASP stranici živi koliko i ASP stranica. Svaka skript naredba na ASP stranici može

pristupiti objektu koji ima "scope" stranice. Objekt koji ima scope stranice "umire" s tom stranicom (kraj procesuiranja stranice).

Ovaj objekt živi koliko i ASP stranica na kojoj je kreiran:

```
<% Set reklama = Server.CreateObject("MSWC.AdRotator") %>
```

ili ovaj:

```
<OBJECT RUNAT=Server ID=reklama  
PROGID="MSWC.AdRotator"></OBJECT>
```

ili ovaj:

```
<OBJECT RUNAT=SERVER ID=MyAds  
CLASSID="Clsid:1621F7C0-60AC-11CF-9427-  
444553540000"></OBJECT>
```

U našim praktičnim primjerima profesionalnih web aplikacija u ovoj knjizi česti su slučajevi kreiranja objekata unutar petlje. Takav objekt je potreban samo dok se "vrti" petlja i šteta je da živi duže od toga. Rješenje je u "razaranju" objekta i prije kraja života stranice. Ova će naredba na kraju tijela petlje Set objReklama = Nothing nakon svakog kruga petlje razoriti objekt, pa ga tako *neće biti na životu* nakon zadnjeg kruga.

```
<%  
  Dim objReklama  
  For i = 0 To 100  
    Set objReklama =  
Server.CreateObject("MSWC.AdRotator")  
    .  
    .  
    .  
    Set objReklama = Nothing  
  Next  
%>
```

Objekt koji živi koliko i jedna posjeta (session)

Da bi neki objekt imao životni vijek jedne posjete (*session*), moramo ga pohraniti u ASP ugrađeni objekt Session. Kreiramo ga na Global.asa stranici pomoću HTML taga <OBJECT> ili na ASP stranici pomoću metode Server.CreateObject.

Primjer s HTML tagom <OBJECT>:

```
<OBJECT RUNAT=SERVER SCOPE=Session ID=MojPreglednik  
PROGID="MSWC.BrowserType">
```

```
</OBJECT>
```

Ovdje je kreirana objekt varijabla MojPreglednik, koja će živjeti koliko i posjeta, a u njoj je pohranjen objekt "BrowserType".

Dok traje posjeta, dok posjetitelj surfa od stranice do stranice u našoj web aplikaciji, skript na svakoj ASP stranici te aplikacije može dohvatiti instanciju objekta MojPreglednik ovako:

```
<%= If MyBrowser.browser = "IE" and MyBrowser.majorver  
>= 4 Then . . .%>
```

Kreiranje instancije pomoću metode Server.CreateObject na ASP stranici izgleda ovako:

```
<% Set Session("MojPreglednik") =  
Server.CreateObject("MSWC.BrowserType") %>
```

Ako želimo ispisati na web stranici neke podatke iz našeg objekta MojPreglednik, najprije pozovemo instanciju objekta, pa onda pomoću naredbe ubacimo podatke na stranicu:

To display browser information in a different .asp file, you first retrieve the instance of the **BrowserType** object stored in the **Session** object, and then call the **Browser** method to display the name of the browser:

```
<% Set MojPreglednik = Session("MojPreglednik") %>  
<%= MojPreglednik.browser %>
```

Session objekt koji ste deklarirali pomoću HTML taga <OBJECT>, činom deklaracije nije i instanciran. Kada je deklariran metodom Server.CreateObject, odmah je i instanciran. Zato je HTML tag objekt <OBJECT> povoljniji za pravljenje skalabilnih aplikacija (bolje upravljamo s resursima, tj. memorija se opterećuje što je više moguće samo onda kada je to i stvarno potrebno).

Kada koristimo objekt koji će živjeti koliko i posjeta, moramo znati *thread* model komponente iz koje nam dolazi objekt (vidi otprije u knjizi). Thread model komponente utječe na performanse i sigurnost objekta u MTS (Microsoft Transaction Server) okruženju. Model komponente Both-Thread će dati najbolje rezultate. Ako koristimo single-threaded ili apartment-threaded objekte, Web poslužitelj će "zaključati" session u jednom *threadu*. Free-threaded objekti ne "zaključavaju" session u jedan thread, ali su spori.

Objekt koji živi koliko i jedna aplikacija

Instancija *application-scope* objekta se kreira kad web aplikacija primi poziv od prvog posjetitelja, a "umire" nakon nekog zadanog vremena nakon što je otišao posljednji posjetitelj. Svi posjetitelji "vide" ovaj objekt. Ukoliko bismo željeli u svakom trenutku znati koliko posjetitelja ima naša web aplikacija, angažirali bi objekt koji živi koliko i web aplikacija. Sintaksa za kreiranje objekta Application je ista kao za objekt Session.

Lekcija 5

Varijable

O deklariranju varijabli pisano je ranije u knjizi. Sve što smo rekli o životu objekata vrijedi i za život varijabli. Varijable su nazivi lokacija u memoriji. U te lokacije pohranjujemo podatke, kao što su brojevi i stringovi. Varijable su tu da vršimo manipulaciju podacima. Kada budete pratili kôdove u primjerima u knjizi, najviše vremena potrošit će te baš na varijable i često se pitati zar je baš trebalo kreirati 15 varijabli za tako jednostavan zadatak. Potom ćete pokušati smanjiti broj varijabli, ali program neće raditi.

Tu je nedoumicu slikovito objasnio John Smiley u svojim knjigama. Lako nam je zamisliti pekare ispred velike peći. Kruh je i u pećnici A i u pećnici B. Kako pekar vrši zamjenu mjesta. Vršiti u tri koraka, ne u dva:

1. Izvadi kruh iz pećnice A i stavi ga na stol.
2. Izvadi kruh iz pećnice B i stavi ga u pećnicu A.
3. Kruh sa stola stavi u pećnicu B.

Isto se događa ako u programu želimo zamijeniti mjesta vrijednostima varijabli A i B. Neka je na memorijskoj lokaciji A trenutna vrijednost 8, a na memorijskoj lokaciji B neka je 12. Zamjenu ćemo napraviti također u tri koraka:

$C = A$ (znači $C = 8$)

$A = B$ (znači $A = 12$)

$B = C$ (znači $B = 8$)

Zašto nismo mogli obaviti zamjenu u jednom koraku (ili u dva koraka):

$A = B$ (znači "obriši trajno postojeću vrijednost u A, a to je 8, pa pohrani novu vrijednost, a to je 12").

Nakon gornje naredbe trajno je nestalo vrijednosti iz A, i naša namjera da tu vrijednost premjestimo u B je propala na samom početku.

Varijabla deklarirana u proceduri je dohvatljiva samo naredbama u toj proceduri, pa kažemo da ima "local scope", da je lokalna. Varijabla deklarirana izvan procedure je dohvatljiva svim naredbama na ASP stranici, pa kažemo da ima "global scope", da je globalna.

Vrijedi samo za VBScript: na početak obvezno stavljati `<% Option Explicit %>`. Gdje na početku stranice? Odmah nakon bilo koje direktive, a prije početka HTML tagova.

```
<% Option Explicit %>
<HTML>
<%
    Dim strIme
    Public lngJMBG
%>
.
.
```

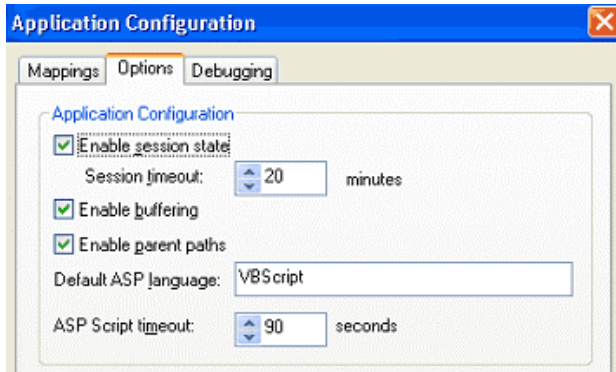
U primjeru e-knjižare sretat ćemo naredbu `#include` pomoću koje ASP stranice dijele međusobno zajednički kôd. U tom zajedničkom kôdu spominju se varijable koje postoje i na ASP stranicama. Samo kada na početku ASP stranice, odmah iza naredbe, napišemo `<% Option Explicit %>`, smijemo koristiti iste nazive varijabli na stranicama i u dijelovima koji su na stranicu umetnuti preko `#include`.

Ista imena mogu imati lokalna i globalna varijabla na ASP stranici. Nikad neće biti nesporazuma ako na vrhu stranice stoji `<% Option Explicit %>`.

Varijable životnog vijeka Session i Application

Session

Iako varijable dohvatljive na jednoj ASP stranici imaju globalni vijek, ta se globalnost ograničava samo na jednu ASP stranicu. U više od jedne stranice mogu se održati na životu samo varijable Session i Application. Život varijable Session se proteže preko svih ASP stranica u jednoj web aplikaciji koje je pozvao jedan posjetitelj u jednoj posjeti. Session je jedna posjeta. U IIS u dijalogu Properties (Configuration/Options) od web aplikacije postoji naredba Enable session state i naredba Session timeout. Kod mene je na slici dolje namješteno na 20 minuta:



To znači varijable Session će se *razoriti* 20 minuta nakon što je posjetitelj zadnji put nešto odabrao u web aplikaciji. Ukoliko je to bio neki dugi dokument pa se posjetitelj na njemu zadržao više od 20 minuta (ili je skoknuo do kuhinje i zadržao se), ako ponovno nakon isteka 20 minuta posjetitelj odabere neki link unutar web aplikacije, započinje nova session i varijable iz prethodne su razorene. To dalje znači, ako je posjetitelj odabrao vanjski link i napustio našu web aplikaciju, pa je prije isteka 20 minuta odabrao ponovo link u našoj web aplikaciji, session je neprekinuta i sve varijable Session su na životu.

Sintaksa za kreiranje varijabli koje će živjeti jednu 'session' je:

```
<%  
    Session("Ime") = "Ivica"  
    Session("Prezime") = "Kartelo"  
%>
```

Što smo učinili? Ugrađenom ASP Objektu Session smo ugradili dvije varijable i pridružili im vrijednosti.

Kada nam te vrijednosti zatrebaju na nekoj ASP stranici, iz objekta Session ćemo ih pozvati pomoću ASP naredbe ovako:

```
Dobro došli <%= Session("Ime") %>
```

Ako varijablu iz objekta session koristite više puta na jednoj ASP stranici, bolje su performanse programa ako kreiramo globalnu varijablu ovako:

```
<%  
    strIme = Session("Ime")  
    If strIme = "Ivica" Then  
    .  
    .  
%>
```

Varijabla je ostala živjeti u objektu Session, ali je na nekim stranicama kopirana u globalnu varijablu radi bržeg obavljanja posla na toj stranici.

Application

Varijable u objektu Application žive od prve do zadnje posjete (session). Varijabla se kreira tako što imenujete "ulaz" u objekt Application i pridodate mu vrijednost:

```
<% Application("DobroDosli") = "Dobro do&#352li u e-  
knji&#382aru Kartelo!" %>
```

Koristio sam ž umjesto slova ž i Š umjesto slova š.

Vrijednost varijable se umeće (poziva) u HTML dokument pomoću ASP naredbe ovako:

```
<%= Application("DobroDosli") %>
```

Ako se neka Application varijabla više puta poziva u jednoj proceduri ili na jednoj ASP stranici, kreirajmo lokalnu, tj. globalnu varijablu za tu stranicu jer će se brže izvršavati.

Primjer web aplikacije OnlineUčilišteKartelo. Koliko trenutno posjeta ima naša web aplikacija.

Global.asa:

```
<SCRIPT RUNAT=Server LANGUAGE=VBScript>
```

```
Sub Session_OnStart
```

```
Application("NumberOfVisitors") =  
Application("NumberOfVisitors") + 1
```

```
End Sub
```

```
Sub Session_OnEnd
```

```
Application("NumberOfVisitors") =  
Application("NumberOfVisitors") - 1
```

```
End Sub
```

```
Sub Application_OnStart
```

```
Application("NumberOfVisitors") = 0
```

End Sub

</SCRIPT>

U kôd Home Page index.asp sam ubacio ovu naredbu:

```
Online users: <%=Application("NumberofVisitors")%>
```

Koristili smo ugrađeni ASP objekt Application i u njegovu instanciju smo pohranili najprije "0" (u trenutku prvog posjetitelja), a onda tu nulu uvećavali za 1 za svakog posjetitelja koji dođe, i umanjivali za 1 za svakoga koji iziđe iz naše web aplikacije.

Konstante

Na početku knjige smo opisali primjer stranice MultiScrolling_VBScript.asp. Budući da mnogi kupci ove knjige već imaju predznanje u programiranju, najvjerojatnije će najprije pregledati primjere i zadržati se na onima koji im donose nešto novo.

ADO komponenta koja dolazi s ASP i omogućava našem ASP skriptu pristup bazama podataka, ima tablice konstanti napisane na posebnim dokumentima. Konstante se odnose na rad s bazama, a u primjeru ove stranice se radi o konstanti adOpenStatic čija je vrijednost 3. Konstante služe za lakšu uporabu brojnih vrijednosti jer je programeru lakše zapamtiti imena konstanti nego brojeve.

Dokument ili biblioteka je pozvan ovako:

```
<!--METADATA TYPE="typelib"
```

```
uuid="00000206-0000-0010-8000-00AA006D2EA4" -->
```

Kada želimo da gornji poziv biblioteke koristi svim ASP stranicama u web aplikaciji, taj ćemo poziv napisati u Global.asa.

Mogli smo se pozvati i na put datoteke:

```
<!-- METADATA TYPE="typelib" FILE="c:\program  
files\common files\system\ado\msado15.dll"-->
```

Prijenos Arrays-a u proceduru

Kada želimo prenijeti cijeli jedan array u neku proceduru, koristimo ime array-a prazne okrugle zagrade.

Rad s kolekcijama

Naš veliki primjer iz prakse, e-knjižara Kartelo, koristi puno manipulacija s kolekcijama. U jednoj kolekciji mogu biti pohranjeni stringovi, brojevi, objekti i druge vrijednosti. Mnogi ugrađeni ASP objekti donose kolekcije. Kolekcija je takva struktura podataka koja se automatski povećava ili smanji s učitavanjem novog člana (Item-a) ili brisanjem postojećega. Zato se pozicija članova stalno mijenja u strukturi, pa se tako mijenja i index članova. Zato je najsigurnije dohvatiti pojedine članove kolekcije po njihovom jedinstvenom imenu ili ključu, stringu Key. Članovi se mogu dohvatiti pomoću Key, Index-a (pozicije u kolekciji) i iteracije kroz sve Item-e u kolekciji.

Primjer objekta Session i kolekcije Contents

Često koristimo instancije ugrađenog ASP objekta Session. U instanciju objekta Session pohranjujemo varijable, ali i instancije drugih ugrađenih objekata koje smo kreirali pomoću metode **Server.CreateObject**. Objekt Session ima kolekciju **Contents** u kojoj se nalaze sve pohranjene varijable i objekti.

Naši primjeri su puni kolekcija Contents, u koje smo pohranili osobne podatke naših kupaca:

```
<%  
    Session.Contents("Ime") = "Ivica"  
    Session.Contents("Prezime") = "Kartelo"  
    Session.Contents("Grad") = Split  
%>
```

Dohvat pojedinog člana kolekcije možemo učiniti pomoću Key, koji uvijek dajemo članu prilikom pohranjivanja istoga u kolekciju:

```
<%= Session.Contents("Ime") %>
```

Gornja naredba će izbaciti Key "Ivica".

Ili pomoću Index-a ili pozicije člana u kolekciji:

```
<%= Session.Contents(1) %>
```

Opet, i ova naredba izbacuje "Ivica" jer je "Ivica" prvi član kolekcije.

Ovakav skraćeni dohvat člana kolekcije...

```
<%= Session("Ime") %>
```

... je moguć kod objekata Session i Application, ali nije preporučljiv kod objekta Request, u kojem se imena znaju ponoviti.

Objekt Request ima ove kolekcije: ClientCertificate, Cookies, Form, QueryString, ServerVariables.

Član ...

```
<%= Request("Ime") %>
```

... može postojati u više kolekcija...

```
<%= Request.ClientCertificate("Ime") %>
```

...ili...

```
<%= Request.Cookies("Ime") %>
```

...ili...

```
<%= Request.Form("Ime") %>
```

...itd.

Dohvat članova kolekcije pomoću iteracije **For...Each...**

```
<%  
    'Deklaracija varijable broja.  
    Dim strClanKolekcije  
  
    'Za svakog člana u kolekciji prikaži njegov broj.  
    For Each strClanKolekcije In Session.Contents  
        Response.Write Session.Contents(strClanKolekcije) &  
    "<BR>"  
    Next  
%>
```

Možemo dohvatiti članove kolekcije i pomoću iteracije VBScript **For...Next** naredbe. Npr., iteracija samo kroz prva tri člana kolekcije:

```
<%  
    'Deklaracija varijable broja.  
    Dim intClan
```

```
For intClan = 1 To 3
    Response.Write Session.Contents(intClan) & "<BR>"
Next
%>
```

ASP podržava svojstvo (properties) **Count** pomoću kojega ću saznati koliko članova ima u kolekciji.

```
<%
    Dim lngItem, lngCount

    lngCount = Session.Contents.Count

    For lngItem = 1 To lngCount
        Response.Write Session.Contents(lngItem) & "<BR>"
    Next
%>
```

Razlika između VB i ASP kolekcija

ASP kolekcije podržavaju property Count i metode Item, Remove i RemoveAll, ali ne podržavaju metodu Add.

Cookie

je naziv za malu tekstualnu datoteku koju naša web aplikacija pošalje na klijentovo računalo. Svaka posjeta automatski generira na poslužitelju jedinstveni broj ID, koji pomoću Response možemo poslati u Cookie datoteku klijentu. Također, preko objekta Session prenesemo taj ID u tablicu naše baze. Kada klijent ponovo dođe na našu aplikaciju, u njegov HTTP zahtjev će se "ukrcati" i vrijednost iz Cookie-a. Naša web aplikacija će usporediti pristigli ID s onima u tablici svoje baze, i ako ga pronađe, može odaslati poruku "Drago nam je što ste opet s nama!", a ako ga ne pronađe, znači da je novi, pa mu možemo mu poslati poruku "Dobro došli!", ali i poslati Cookie s generiranim ID.

Na taj bismo način mogli puno toga saznati o našim posjetiteljima: što kupuju, koje ih stranice zanimaju te puno toga o njihovom browser-u itd. A potom kreirati i Gausovu krivulju i otkriti tko je naša najbolja publika.

Ukoliko je posjetitelj na svom browser-u isključio opciju Cookie, gornji koncept nije izvediv.

Lekcija 6 ADO – Knjiga gostiju

ADO (ActiveX Data Objects) je ugrađena ASP komponenta koja omogućava skriptu na našim ASP stranicama pristup bazama podataka, Excelu, e-mail mapama, tekstualnim i drugim datotekama. Podaci kojima se pristupa pomoću ADO su pohranjeni u sustavima koji podržavaju standarde OLE DB i ODBC, bez obzira na proizvođače softvera za pohranu podataka.

Napomena: U ovoj knjizi radimo s Access bazama. Za projekte u fazi razvoja ili "žive" projekte do 10 konkurentnih posjetitelja, uporaba Access-a je u redu. Preko toga ne daje dobre rezultate. Access je Desktop aplikacija, a ne klijent-server. Zato na Internet uvijek izlazimo s pravim klijent-server bazama, kao što su SQL Server ili Oracle i dr.

Spajanje na izvor podataka

Komponenta ADO, kao most između našeg skripta i baze podataka, sadrži objekt Connection. Taj objekt ima svojstva i metode pomoću kojih će naš skript otvoriti bazu, aktivirati upite za ažuriranje baze te zatvoriti bazu.

Objasnimo na primjeru knjige gostiju objekt Connection (cijeli kôd je u mapi Lekcija6).

Analiza ASP stranice `guestbook1.asp`

Ako varijabla Message ne sadrži vrijednost True, tada...

```
<% @Language=VBScript %>

<html>

<head>
```

```

<title>Knjiga gostiju pomo&#263u objekta
Connection</title>

</head>

<body>

<font face="MS Gothic">

<h2>Knjiga gostiju pomo&#263u objekta Connection</h2>

<%
If Not Request.QueryString("Message") = "True" Then
%>

...izbacimo na zaslon formu u koju će posjetitelj upisati podatke. Svaka forma
mora imati gumb tipa Submit. Pritiskom na taj gumb aktivira se HTTP zahtjev za
onim što je navedeno u atributu ACTION: guestbook1.asp. ...

%>

<p>

<FORM NAME="GuestBook1" METHOD="GET"
ACTION="guestbook1.asp">

<table>

<tr>

<td><font face="MS Gothic">From:</td><td><INPUT
TYPE="TEXT" NAME="From"></td>

</tr><tr>

<td><font face="MS Gothic">E-mail
Address:</td><td><INPUT TYPE="TEXT"
NAME="EmailAdd"></td>

</tr><tr>

<td><font face="MS Gothic">CC:</td><td><INPUT
TYPE="TEXT" NAME="CC"></td>

```

```

        </tr><tr>
        <td><font face="MS
Gothic">Subject:</td><td><INPUT TYPE="TEXT"
NAME="Subject"></td>
        </tr>
</table>

Message:<br><TEXTAREA NAME="Memo" ROWS=6
COLS=70></TEXTAREA>

</p>
<p>
<INPUT TYPE="HIDDEN" NAME="Message" VALUE="True">
<INPUT TYPE="SUBMIT" VALUE="Submit Information">
</FORM>
</p>
<%

```

U gornjoj se formi opet od poslužitelja traži ista ASP stranica na kojoj je forma, samo će ovaj put varijabla Message imati vrijednost True i u povratnu web stranicu ulazi skript Else... Obratite pozornost na način kako je kreirana varijabla Message. Kreiran je gumb koji posjetitelj neće vidjeti jer je tip HIDDEN. Atribut NAME tog gumba kreirao je varijablu Message. Nakon procesuiranja ovog retka, varijabla Message će dobiti vrijednost True.

```
<INPUT TYPE="HIDDEN" NAME="Message" VALUE="True">
```

Dakle, kada je Message = True, browser-u se šalje rezultat skripta Else...

```

Else

    dim strTB1, strTB2, strTB3, strTB4, strMB1,
strCommand

    strTB1 =
Server.HTMLEncode(Request.QueryString("From"))

```

```

    strTB2 =
Server.HTMLEncode(Request.QueryString("EMailAdd"))

    strTB3 =
Server.HTMLEncode(Request.QueryString("CC"))

    strTB4 =
Server.HTMLEncode(Request.QueryString("Subject"))

    strMB1 =
Server.HTMLEncode(Request.QueryString("Memo"))

```

U gornjem skriptu su najprije globalno deklarirane varijable (6 ukupno). Zatim je za 5 od 6 varijabli pridodana vrijednost, ovako...

```
strTB1 = Server.HTMLEncode(Request.QueryString("From"))
```

U varijablu strTB1 pohranjuje se string vrijednost iz memorije poslužitelja, Request.QueryString("From"). Kolekcija QueryString ugrađenog ASP objekta Request je uhvatila string iz kontrole <INPUT TYPE="TEXT" NAME="From"> i pohranila ga u memoriju poslužitelja. Taj string se sada pridodaje globalnoj varijabli strTB1 koja će živjeti koliko i ASP stranica i sve naredbe do kraja stranice će moći dohvatiti tu varijablu ako bude trebalo. Prije nego je string pohranjen u varijablu strTB1, enkodiran je pomoću metode HTMLEncode koja pripada ugrađenom ASP objektu Server. Zašto je potrebno HTMLEncode-irati string i tako enkodiran pohraniti kasnije i u bazu? Zato što netko može napisati u polje From na formi i ovako <TABLE>. Kutne zagrade pripadaju HTML skriptu, i u tom skriptu imaju ulogu okruživanja taga. Browser neće prikazati <TABLE> nego će "misliti" da je to tag, pa ga neće prikazati. Nakon metode HTMLEncode string <TABLE> će izgledati ovako <TABLE> i taj string će dospjeti u bazu. Kada taj string pozovemo kasnije na browser, vidjet ćemo ovo <TABLE>, a ne <TABLE>

Gore je pet varijabli uhvatilo podatke iz forme, a šesta će varijabla dobiti vrijednost kasnije.

Imamo globalne varijable, imamo u njima stringove iz forme, preostaje nam još samo te stringove pohraniti u tablicu GuestBook u Access bazi guestbook.mdb.

GuestBook : Table						
	FID	FTB1	FTB2	FTB3	FTB4	FMB1
▶	16	Ivica Kartelo	skola-e-92@st.		Web	Ovoj je samo te
	17	Valentina	w@valentini.coi		web	Tnms asasn ak
	18	Jedan	Dva	Tri	Cetri	Tjkasnb sasmk
	19	<TABLE>	nesto			GF asmnams a
	20	MacMac	nesto			GF asmnams a
	21	<%tablica%&				JKhak snkas m
	22	("<%t&				
*	(AutoNumber)					

Polje FID u gornjoj tablici je AutoNumber, pa ga mi ne unosimo preko forme, nego takav tip podataka unosi automatski Access.

ASP sadrži Access driver. Kreirat ćemo varijablu u kojoj će biti pohranjen *connection string*. ADO će se pomoću tog stringa, a kroz Access driver, spojiti na bazu. String sadrži varijable odijeljene točkom-zarezom ','.

```
strProvider = "Provider=Microsoft.JET.OLEDB.4.0;Data
Source=C:\KnjigaASP\Lekcija6\guestbook.mdb;Mode=ReadWri
te"
```

Prva varijabla definira dobavljača Access driver-a `Provider=Microsoft.JET.OLEDB.4.0;`, druga fizički put baze podataka `Data Source=C:\KnjigaASP\Lekcija6\guestbook.mdb;`, a treća način rada `Mode=ReadWrite`", koji će nam omogućiti i čitanje i upisivanje novih podataka u bazu.

Sada kad imamo varijablu *connection string*, možemo kreirati instanciju ADO objekta *Connection*. ADO ima još četiri druga objekta koji imaju svoje metode i svojstva pomoću kojih se može napraviti sve s bazama podataka.

Instancija ADO objekta *Connection* kreira se ovom naredbom:

```
Set objConn = server.createobject("ADODB.Connection")
```

Objekt oživljuje činom spajanja na bazu:

```
objConn.Open strProvider
```

U ovoj točki našeg VBScripta imamo spoj s Access bazom *guestbook.mdb*. SQL skript koji gradi Access bazu, podržava sve potrebne SQL upite za rad s bazom. Pomoću SQL upita dodajemo nove retke tablici (*Insert*), brišemo retke (*Delete*), ažuriramo podatke (*Update*) ili samo gledamo podatke (*Select*).

SQL upit koji slijedi treba unijeti podatke u tablicu GuestBook, pa će zato koristiti SQL ključne riječi INSERT INTO. Bit će potrebni i apostrofi oko stringova. GuestBook je naziv tablice.

```
strCommand = "INSERT INTO GuestBook  
(FTB1,FTB2,FTB3,FTB4,FMB1) VALUES ('"  
  
strCommand = strCommand & strTB1 & "','" & strTB2 &  
"','" & strTB3 & "','" & strTB4 & "','" & strMB1  
  
strCommand = strCommand & "')
```

U gornjem kôdu, varijabla strCommand u tri je naredbe, radi preglednosti, pohranila cijeli string koji se sastoji od svih 5 varijabli za svih 5 polja u tablici, za jedan novi redak.

Slijedi naredba izvršavanja SQL upita. Ova je naredba najjednostavnija moguća naredba. Poziva metodu Execute, koja pripada objektu Connection...

```
objConn.Execute strCommand
```

ASP objekt Response šalje browser-u naredbu "ispiši tekst..."

```
Response.Write("Hvala, va&#353a poruka je pohranjena u  
na&#353oj bazi. Sve poruke mo&#382ete vidjeti ako  
odaberete donji link")
```

Kraj strukture If...End If...

```
End If  
  
%>
```

Nalazimo se izvan ASP procesuiranja, u prostoru HTML tagova. Ako želimo vidjeti sve poruke, u tablici GuestBook odaberimo Link...

```
<A HREF="GuestBookList1.asp">Vidi sve poruke u knjizi  
gostiju</A>  
  
</font>  
  
</body>  
  
</html>
```

Gledanje poruka u knjizi gostiju.
Analiza ASP stranice GuestBookList1.asp.

```
<% @Language=VBScript %>

<html>

<head>

<title>Knjiga gostiju pomo&#263u objekta
Connection</title>

</head>

<body>

<font face="MS Gothic">

<h2>Knjiga gostiju pomo&#263u objekta Connection</h2>
```

Moramo učiniti sve što smo učinili i u prethodnoj stranici. Najprije deklaracija varijabli...

```
<%

dim strTB1, strTB2, strTB3, strTB4, strMB1, strCommand

...pa kreiranje connection stringa...

strProvider = "Provider=Microsoft.JET.OLEDB.4.0;Data
Source=C:\KnjigaASP\Lekcija6\guestbook.mdb;Mode=ReadWri
te"
```

Slijedi kreiranje instancije objekta Connection...

```
Set objConn = server.createobject("ADODB.Connection")
```

...pa otvaranje baze pomoću metode Open objekta ...

```
objConn.Open strProvider
```

Slijedi nešto što nismo imali u gornjoj ASP stranici; kreiranje instancije ADO objekta Recordset...

```
set objrs=server.CreateObject("ADODB.Recordset")
```

U ovoj točki našeg VBScripta imamo dvije instancije objekata. Jedna je instancija objekta Connection, a druga objekta Recordset. Prva uspostavlja vezu s bazom, a druga instancija objekta je spremna primiti vrijednost SQL upita. Zato kreiramo globalnu varijablu sql i pridodajemo joj vrijednost SQL upita SELECT...

```
sql="select * from guestbook"
```

Konačno možemo oživjeti konkretni objekt Recordset, naredbom open...

```
objrs.open sql, objconn, 3, 2, 1
```

Gornja naredba open nad instancijom objekta Recordset-a objrs, ima pet parametara: "sql" je varijabla čiji će se sadržaj otvoriti u objektu objrs. "objconn" je objekt Connection. Slijede brojevi umjesto naziva konstanti i, između ostaloga, definiraju i tip kursora. O tipu kursora ovisi što sve možemo raditi s objektom Recordset (brisati retke ili ne brisati, dodavati retke ili ne, mijenjati podatke itd.).

Objekt Recordset objrs je objekt kolekcija. Drži u memoriji cijelu tablicu GuestBook. Ispis u browser-u svih redaka iz objekta objrs vršimo pomoću ASP objekta Response i iteracije kroz kolekciju objrs pomoću petlje Do...While...Loop...

```
do while not objrs.EOF
```

```
Response.Write "<b>" & objrs(1) & " write:</b><br><i>"  
& objrs("fmb1") & "</i><br><br>"
```

```
objrs.MoveNext
```

```
loop
```

```
objrs.close
```

```
%>
```

Kada je petlja odvrtila i posljednji redak, rasterećujemo memoriju zatvaranjem objekta kolekcije objrs naredbom objrs.close. Oslobađamo i zauzeće baze.

Slijede HTML tagovi...

```
Pritisnite <A HREF="GuestBook1.asp">ovdje</A> ako  
&#382elite poslati poruku u na&#353u bazu.
```



```
</font>
```

```
</body>
```

```
</html>
```

Lekcija 7 ADO – Unos, brisanje, editiranje, ažuriranje

Tri su temeljne radnje nad svakom bazom podataka: dodavanje (Add) novog sloga (retka), brisanje retka (Delete) i izmjene u retcima (Edit). Ima još jedna radnja, četvrta, koja prati radnje Edit i Add. To je ažuriranje ili Update. Ona se zbiva iza scene, ali ne automatski, nego je programer mora isprogramirati. Sve smo ovo već vidjeli u mojoj knjizi "Visual Basic baze podataka".

Tamo smo iz Visual Basic sučelja upravljali Access bazom podataka. Radilo se o desktop aplikaciji. Desktop aplikacija se sastojala od dvije datoteke: Visual Basic EXE datoteke i Access MDB datoteke. Obje su datoteke bile na lokalnom disku našeg računala.

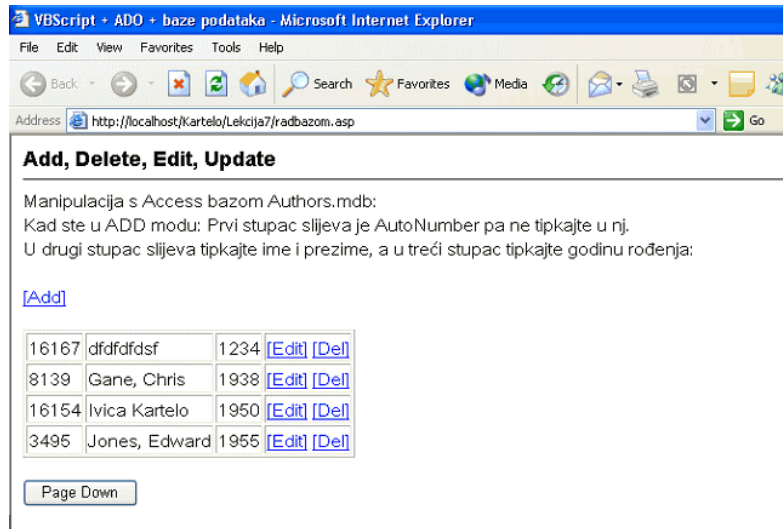
Ovdje se radi o ASP stranicama i datoteci MDB. Iz browser-a ćemo upravljati bazom podataka na drugoj strani Zemlje. U ASP stranicama ćemo pomoću VBScripta i HTML kreirati Web sučelje preko kojega ćemo upravljati bazom podataka. Neće biti velikih razlika u pisanju kôda. VBScript je samo jedna od "library" iz porodice Visual Basic (Visual Basic, Visual Basic for Application, VBScript).

U mapi Lekcija7 se nalazi ASP stranica radbazom.asp i Microsoft Access baza Authors.

Analiza stranice radbazom.asp

1. Otvorite IIS

2. Browse ASP stranicu Lekcija7/radbazom.asp



3. Možemo odabrati sljedeće zahtjeve: Add, Edit, Del i Page Down (kasnije i Page Up).

4. Pogledajmo HTML Source:

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE>VBScript + ADO + baze podataka</TITLE>
```

```
  </HEAD>
```

```
  <BODY BGCOLOR="White" topmargin="10"  
leftmargin="10">
```

```
    <font size="4" face="Arial, Helvetica">
```

```
    <b>Add, Delete, Edit, Update</b></font><br>
```

```
<hr size="1" color="#000000">
```

Manipulacija s Access bazom Authors.mdb:

Kad ste u ADD modu: Prvi stupac slijeva je AutoNumber pa ne tipkajte u nj.

U drugi stupac slijeva tipkajte ime i prezime, a u treći stupac tipkajte godinu rođenja:

Slijedi link ADD, koji poziva istu ASP stranicu radbazom.asp, potom slijedi upitnik i varijabla "act" jednaka vrijednosti Add. Ako odaberemo link Add, iza upitnika act=Add uhvatit će ASP objekt Request i pohraniti u memoriji poslužitelja kao Request("act")=Add...

```
<A HREF="radbazom.asp?act=Add">[Add]</A>
```

Slijedi FORM-a i tablica s prva četiri retka iz tablice Authors. Prikazana su sva tri polja tablice. U svakom retku se nalaze linkovi Edit i Del. Ako ih odaberemo oni će ...

```
<FORM action="radbazom.asp?PageNo=1&act=Update&id="
method=POST id=form1 name=form1>
```

```
<TABLE BORDER=1>
```

```
<TR>
```

```
<TD VALIGN=TOP>16167</TD>
```

```
<TD VALIGN=TOP>dfdfdfdsf</TD>
```

```
<TD VALIGN=TOP>1234</TD>
```

...poslati ove zahtjeve. U zahtjevima su definirane vrijednosti varijabli act=Edit ili act=Delete i id=Au_ID retka u bazi u kojem se nalaze linkovi Edit i Del. Provjerite te linkove u browser-u i pogledajte kako izgledaju zahtjevi kada ste odabrali Edit i kada ste odabrali Del.

```
<TD VALIGN=TOP><A
HREF="radbazom.asp?PageNo=1&act=Edit&id=16167">[Edit]</
A> <A
HREF="radbazom.asp?PageNo=1&act=Delete&id=16167">[Del]<
/A></TD>
```

Kraj prvog retka tablice. Slijede još tri retka, za koja vrijedi sve rečeno za prvi redak.

```
</TR>

<TR>

    <TD VALIGN=TOP>8139</TD>

    <TD VALIGN=TOP>Gane, Chris</TD>

    <TD VALIGN=TOP>1938</TD>

    <TD VALIGN=TOP><A
    HREF="radbazom.asp?PageNo=1&act=Edit&id=8139">[Edit]</A
    > <A
    HREF="radbazom.asp?PageNo=1&act=Delete&id=8139">[Del]</
    A></TD>

</TR>

<TR>

    <TD VALIGN=TOP>16154</TD>

    <TD VALIGN=TOP>Ivica Kartelo</TD>

    <TD VALIGN=TOP>1950</TD>

    <TD VALIGN=TOP><A
    HREF="radbazom.asp?PageNo=1&act=Edit&id=16154">[Edit]</
    A> <A
    HREF="radbazom.asp?PageNo=1&act=Delete&id=16154">[Del]<
    /A></TD>

</TR>

<TR>

    <TD VALIGN=TOP>3495</TD>

    <TD VALIGN=TOP>Jones, Edward</TD>

    <TD VALIGN=TOP>1955</TD>
```

```
<TD VALIGN=TOP><A  
HREF="radbazom.asp?PageNo=1&act=Edit&id=3495">[Edit]</A  
> <A  
HREF="radbazom.asp?PageNo=1&act=Delete&id=3495">[Del]</  
A></TD>
```

```
</TR>
```

```
</TABLE>
```

```
</FORM>
```

Slijedi forma koja metodom POST šalje zahtjev za istom ASP stranicom radbazom.asp. Kada odaberemo gumb Page Down, u zahtjevu će se nalaziti PageNo=1 koji će poslati skrivena kontrola INPUT TYPE="HIDDEN"...

```
<Form Action=radbazom.asp Method="POST">
```

```
<Input Type="Hidden" Name="PageNo" Value="1">
```

```
<INPUT TYPE="Submit" Name="Mv" Value="Page  
Down">
```

```
</Form>
```

```
</BODY>
```

```
</HTML>
```

Kada pritisnemo gumb Page Down i pogledamo HTML Source, zadnja će forma izgledati ovako:

```
<Form Action=radbazom.asp Method="POST">
```

```
<Input Type="Hidden" Name="PageNo" Value="2">
```

```
<!-- Only show appropriate buttons -->
```

```
<INPUT TYPE="Submit" Name="Mv" Value="Page  
Down">
```

```
<INPUT TYPE="Submit" Name="Mv" Value="Page  
Up">
```

```
</Form>
```

Forma će imati dva gumba Submit i poslat će poslužitelju podatak PageNo=2. Skriveni gumb u formi javlja poslužitelju broj stranice s koje dolazi. Ako dolazi s broja 1, to znači da treba poslati 4 nova retka iz tablice Authors, i to od broja 5 do uključno broja 8. Ako dolazi sa stranice broj 2, mora se još znati je li korisnik odabrao Page Down (tada će izbaciti retke 9, 10, 11 i 12) ili Page Up (tada će izbaciti retke 1,2,3,4).

5. Pogledajmo sada VBScript kôd u stranici radbazom.asp

```
<% @LANGUAGE="VBSCRIPT" %>

<% Option Explicit %>

<HTML>

    <HEAD>

        <TITLE>VBScript + ADO + baze podataka</TITLE>

    </HEAD>

    <BODY BGCOLOR="White" topmargin="10"
leftmargin="10">

        <font size="4" face="Arial, Helvetica">

            <b>Add, Delete, Edit, Update</b></font><br>

            <hr size="1" color="#000000">

            Manipulacija s Access bazom Authors.mdb:<br>

            Kad ste u ADD modu: Prvi stupac slijeva je
AutoNumber pa ne tipkajte u nj.<br>

            U drugi stupac slijeva tipkajte ime i prezime, a u
treći stupac tipkajte godinu rođenja:<br><br>

Nakon <% Option Explicit %> slijedi deklaracija varijabli...

    <%

        Dim oConn

        Dim oRs
```

```
Dim filePath  
  
Dim Mv  
  
Dim PageNo  
  
Dim j  
  
Dim i  
  
dim sql
```

Pomoću metode MapPath kreirajmo fizički put do baze authors.mdb i pohranimo ga u varijablu filePath...

```
filePath = Server.MapPath("authors.mdb")
```

Kreiranje instancije ADO Connection Component za spoj s bazom i pomoću metode Open objekta Connection spajanje na bazu..

```
Set oConn = Server.CreateObject("ADODB.Connection")  
  
oConn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=" & filePath
```

Kao što smo gore vidjeli, prvo pojavljivanje stranice donosi prva četiri retka iz tablice Authors i linkove Add, Edit, Del te gumb Page Down.

Kad sada budemo odabrali neki od tih linkova ili gumb, generirat ćemo HTTP zahtjev za istom ASP stranicom radbazom.asp koji će tada u sebi imati definirane varijable "act", "id" itd. Te varijable iz HTTP zahtjeva "hvata" ASP objekt Request i pohranjuje u memoriju poslužitelja.

Naša VBScript mora pomoću strukture IF...END IF utvrditi koji smo link odabrali. To će saznati po vrijednosti varijabli act, id, itd.

Veza s bazom je uspostavljena i sada slijedi "hvatanje" našeg zahtjeva. Ako smo poslali zahtjev za ažuriranjem Update, bit će uhvaćen u objektu Request i slijedi SQL UPDATE. U SQL UPDATE su ovi ključni izrazi: UPDATE AUTHORS SET...WHERE... Svojim riječima bi mogli ovako opisati ovaj SQL. Ažuriraj onaj redak u tablici Authors čiji je Au_ID jednak vrijednosti u memoriji poslužitelja request("id"). request("id") je vrijednost od onoga retka koji smo

mi odabrali u browser-u, a gore smo vidjeli kako se generira u HTTP zahtjevu iz browser-a.

Authors : Table			
	Au_ID	Author	YearBorn
▶	16165	Valentina Valentini	123
	16167	dfdfdfsf	1234
	8139	Gane, Chris	1938
	4694	Steele, Philip	1941
	3193	Wraye, Toby	1950
	16154	Ivica Kartelo	1950
	16160	Rose Carma	1952
	3461	Simpson, Mery	1953
	3495	Jones, Edward	1955
	3915	jsashnakj	1957
	4816	Rodgers, Ulka	1958
	16166	Perica Valentiæ	1970
	16162	Mario	1980
*	(AutoNumber)		0

Evo strukture IF za act=Update...

```
if request("act")="Update" then

    sql="Update authors set Author='" &
request("author") & "',yearborn=" & request("yearborn")
& " where au_id=" & request("id")
```

Slijedi metoda Execute od objekta Connection; to je metoda koja izvršava SQL UPDATE pohranjen u varijabli sql.

```
oConn.Execute sql
```

```
end if
```

Sve gore rečeno za UPDATE vrijedi za strukturu IF, koja slijedi i odnosi se na naš zahtjev DELETE...

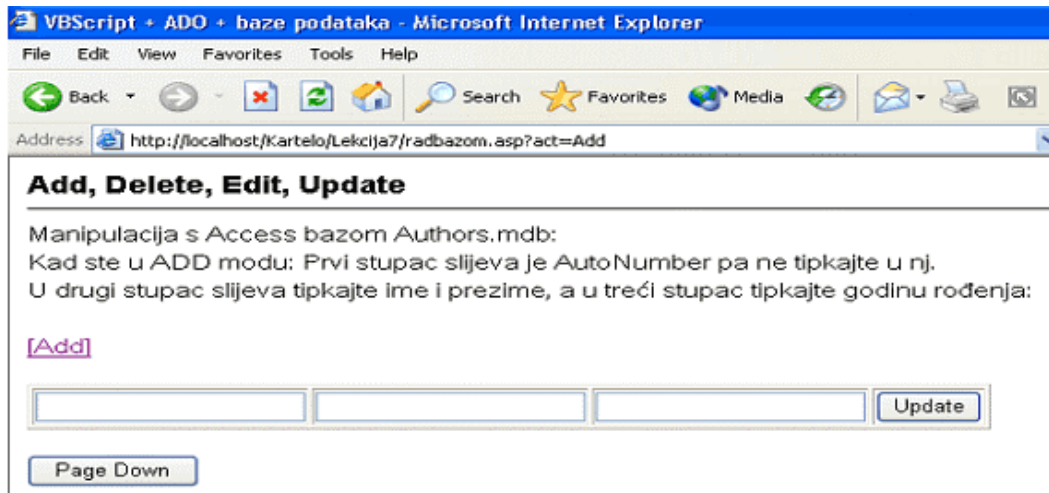
```
if request("act")="Delete" then

    sql="delete from authors where au_id=" &
request("id")

    oConn.Execute sql

end if
```

...pa naš zahtjev ADD2 iz forme koja izgleda ovako u browser-u:



U gornju formu nas je doveo zahtjev act=Add (pogledajte Address u browser-u):

`http://localhost/Kartelo/Lekcija7/radbazom.asp?act=Add`

Pritiskom na gumb Update, koji je tip Submit, šaljemo HTTP zahtjev definiran u Action argumentu forme. Zahtjev izgleda ovako:

```
<FORM action="radbazom.asp?PageNo=1&act=Add2&id="
method=POST id=form1 name=form1>
```

U zahtjevu se traži prva stranica (prva četiri retka iz tablice Authors), a opcija act=Add2 će u strukturi IF kreirati SQL INSERT INTO AUTHORS ime i prezime autora (polje Author u tablici i polje YearBorn u istoj tablici). U novi redak u tablici Authors ubacit će se vrijednosti koje smo tipkali u browser-u. SQL ubacuje samo dva polja jer je Au_ID tip AutoNumber, pa će ga automatski generirati engine Access-a u ASP-u.

Ovdje je zanimljivo što će ASP poslati browser-u nakon izvršenja SQL-a (nakon što je unio novi redak u tablicu Authors). Hoće li generirati posljednja četiri retka na zaslonu? Ili samo zadnji redak? Ne...

```
if request("act")="Add2" then

    sql="insert into authors (Author,yearborn)
values ('" & request("author") & "'," &
request("yearborn") & ")"

    oConn.Execute sql
```

...nego ASP objekt Response, tj. njegova metoda Redirect šalje ponovo zahtjev za istom ASP stranicom radbazom.asp. Posljedica toga je pojava početne forme na zaslonu klijenta, u kojoj su prva četiri retka iz tablice, poredana po abecedi autora.

```
Response.Redirect "radbazom.asp"  
end if
```

Kreiranje instancije objekta Recordset iz komponente ADO...

```
Set oRs = Server.CreateObject("ADODB.Recordset")
```

Koji je tekući PageNumber...

```
Mv = Request("Mv")  
If Request("PageNo") = "" Then  
PageNo = 1  
Else
```

Funkcija Cint vrši konverziju bilo kojeg izraza u podtip variant-integer...

```
PageNo = cint(Request("PageNo"))  
End If
```

Uspostava Query Recordset-a sa četiri sloga (retka) po stranici i složenih po abecedi stupca Author (order by author)...

```
oRs.Open "SELECT * FROM Authors order by author",  
oConn, 3
```

```
oRs.PageSize = 4
```

Generiranje odgovarajuće stranice (PageNumber)...

```
If Mv = "Page Up" or Mv = "Page Down" Then  
Select Case Mv  
Case "Page Up"  
If PageNo > 1 Then
```

```

        PageNo = PageNo - 1
    Else
        PageNo = 1
    End If

    Case "Page Down"
        If oRs.AbsolutePage < oRs.PageCount Then
            PageNo = PageNo + 1
        Else
            PageNo = oRs.PageCount
        End If
    Case Else
        PageNo = 1
    End Select

End If

oRs.AbsolutePage = PageNo

%>

```

Crtanje tablice na browser-u...

```

<A HREF="radbazom.asp?act=Add">[Add]</A>

<% if request("act")<>"Add" then %>

<FORM
action="radbazom.asp?PageNo=<%=PageNo%>&act=Update&id=<
%=request("id")%>" method=POST id=form1 name=form1>

<% else %>
<FORM
action="radbazom.asp?PageNo=<%=PageNo%>&act=Add2&id=<%=

```

```

request("id")%>" method=POST id=form1 name=form1>
<% end if %>

<TABLE BORDER=1>

<% if request("act")<>"Add" then %>

<% For j = 1 to oRs.PageSize

if request("act")="Edit" and
cint(request("id"))=cint(ors("Au_ID")) then

    %>

        <TR>

            <% For i = 0 to oRs.Fields.Count - 1 %>

<TD VALIGN=TOP><INPUT type="text" id=<%=oRs(i).Name%>
name=<%=oRs(i).Name%> value="<%= oRs(i) %>"></TD>

            <% Next %>

<TD VALIGN=TOP><INPUT type="submit" value="Update"
id=button1 name=button1></TD>

        </TR>

<% else %>

        <TR>

            <% For i = 0 to oRs.Fields.Count - 1 %>

<TD VALIGN=TOP><%= oRs(i) %></TD>

            <% Next %>

<TD VALIGN=TOP><A
href="radbazom.asp?PageNo=<%=PageNo%>&act=Edit&id=<%=ors("Au_ID")%>">[Edit]</A> <A
href="radbazom.asp?PageNo=<%=PageNo%>&act=Delete&id=<%=ors("Au_ID")%>">[Del]</A></TD>

        </TR>

```

```

    <% end if

        oRs.MoveNext

        If oRs.EOF Then

            Exit For

        End If

    Next %>

<% else %>

<TR>

    <% For i = 0 to oRs.Fields.Count - 1 %>

<TD VALIGN=TOP><INPUT type="text" id=<%=oRs(i).Name%>
name=<%=oRs(i).Name%>></TD>

    <% Next %>

<TD VALIGN=TOP><INPUT type="submit" value="Update"
id=button1 name=button1></TD>

    </TR>

<% end if %>

</TABLE>

</FORM>

```

Navigacija Page Up i Page Down...

```

    <Form Action=radbazom.asp Method="POST">

        <Input Type="Hidden" Name="PageNo" Value="<%=
PageNo %>">

```

Prikaži odgovarajuće gumbе...

```

    <% If PageNo < oRs.PageCount Then %>

```

```

        <INPUT TYPE="Submit" Name="Mv" Value="Page
Down" >

        <% End If %>

        <% If PageNo > 1 Then %>

        <INPUT TYPE="Submit" Name="Mv" Value="Page Up" >

        <% End If %>

    </Form>

</BODY>

</HTML>

```

U gornjem smo primjeru koristili svojstva Recordset-a AbsolutePage i PageSize. Drugom smo svojstvu zadali četiri (4 retka po jednom prikazu, jednoj stranici), a AbsolutePage svojstvo na osnovu ukupnog broja redaka zna koliko ukupno ima takvih stranica u Recordsetu.

Execute metoda objekta Connection

```
cnn.Execute strSQL, ,adCmdText + adExecuteNoRecords
```

Mi smo u gornjim primjerima koristili samo:

```
cnn.Execute strSQL
```

adCmdText je opcijski parametar koji u gornjim primjerima nismo koristili. Taj parametar određuje tip naredbe, dajući do znanja provideru da mora evoluirati query statement ili pohranjenu proceduru kao tekstualnu definiciju naredbe.

adExecuteNoRecords parametar instruirá ADO "ne kreirati recordset podataka ako je povratni rezultat prema aplikaciji prazan". Ovaj parametar može íći samo s parametrom adCmdText.

Iako su ova dva parametra opcijaska, kada ih koristimo popravljamo performanse rada s bazom.

ADO parametri moraju biti definirani prije nego ih koristimo. Svi ADO parametri su definirani u jednoj datoteci. Implementaciju te datoteke u našu ASP stranicu ili web aplikaciju vršimo njenom deklaracijom...

```
<!--METADATA NAME="Microsoft ActiveX Data Objects 2.5  
Library" TYPE="TypeLib" UUID="{00000205-0000-0010-8000-  
00AA006D2EA4}" -->
```

...na početku ASP lista ili u datoteci Global.asa.

SQL naredbe INSERT, DELETE, UPDATE služe za UNOS novog retka u bazu, brisanje retka baze i promjene podataka u retku baze.

Napomena. Treba biti oprezan sa SQL naredbom DELETE. Uvijek treba koristiti klauzulu WHERE za određivanje točno određenog retka za brisanje. Bez te klauzule će biti obrisani svi retci u tablici.

ADO objekt Recordset

ADO sadrži objekt Recordset. Taj objekt služi VBScriptu na našim ASP stranicama za kopiranje redaka iz tablica baze u memoriju poslužitelja. Svi kopirani retci su memorirani u instanciji objekta Recordset. Recordset može biti sastavljen od nijednog, jednog ili više redaka. Svaki redak ima polja (Fields). Objekt Recordset ima kolekciju Fields, koju smo koristili u gornjim primjerima, a još ćemo je puno koristiti i u velikim projektima u ovoj knjizi (e-knjžara Kartelo).

Objekt Connection i objekt Recordset u paru mogu učiniti sve potrebne radnje nad bazom podataka. Prvi objekt uspostavlja vezu VEBScripta i baze, a drugi objekt kreira u memoriji SQL naredbu, npr. SQL SELECT kad želi kopirati retke iz baze. S klauzulom WHERE, SELECT...Where sužavamo krug pozvanih redaka.

Ukoliko nemamo potrebe objektu Connection specificirati neka svojstva, već ga koristiti default, možemo uštedjeti u tipkanju naredbi ovako.

Definiramo connection string...

```
<%  
strConnectionString =  
"Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=C:\Podaci\Razred.mdb"
```

Kreiramo SQL SELECT...WHERE naredbu...


```
strSQL = "SELECT Ime, Prezime FROM Ucenici WHERE Prezime  
= 'Kartelo' "
```

Kreiramo instanciju objekta RECORDSET. Toj naredbi uvijek prethodi, automatski, iza scene a da to nismo naredili u našem kôdu, kreiranje instancije objekta Connection s njegovim default postavkama...

```
Set rstCustomers = Server.CreateObject("ADODB.Recordset")
```

Iako u svom kôdu nigdje nismo spomenuli objekt Connection, on je u ovoj točki prisutan i otvaramo ga, metodom OPEN, od objekta RECORDSET, ovako...

```
rstCustomers.Open strSQL, strConnectionString
```

Prođimo kroz retke RECORDSET-a , prikazujući ih u browser-u pomoću metode MoveNext...

```
Set objIme = rstUcenici("Ime")  
Set objPrezime = rstCustomers("Prezime")  
Do Until rstUcenici.EOF  
    Response.Write objIme & " " & objPrezime & "<BR>"  
    rstUcenici.MoveNext  
Loop  
%>
```

Metoda Open objekta Recordset pruža mogućnost uporabe opsijskog parametra cursor. Taj parametar definira kako će provider drivera izvući retke, kako ćemo se moći kretati od retka do retka itd. Tako, na primjer, cursor parametar adOpenKeyset, koji dodamo naredbi Open Recordseta, omogućava klijentu punu *navigaciju* kroz retke recordseta.

Tako otvoren Recordset može koristiti svoja svojstva, na primjer RecordCount, da bi script točno znao na kojem se retku nalazi kursor.

```
<%  
Set rs = Server.CreateObject("ADODB.Recordset")  
rs.Open "SELECT * FROM Narudzbe",  
"Provider=Microsoft.Jet.OLEDB.3.51;Data  
Source='C:\Prodaja\Narudzbe.mdb'", adOpenKeyset,  
adLockOptimistic, adCmdText  
    If rs.RecordCount >= 5 then  
        Response.Write "Primili smo sljedeće " &  
rs.RecordCount & " nove narudzbe<BR>"  
        Do Until rs.EOF  
            Response.Write rs("Ime") & " " & rs("Prezime") &  
" <BR>"
```

```

        Response.Write rs("BrojRacuna") & "<BR>"
        Response.Write rs("Kom") & "<BR>"
        Response.Write rs("IsporukaDatum") & "<BR><BR>"
        rs.MoveNext
    Loop
    Else
        Response.Write "Ima manje od " & rs.RecordCount &
" novih narudžbi."
    End If
    rs.Close
%>

```

ADO objekt Command

ADO objekt Command izvršava upit ili pohranjene procedure. Čini, dakle, potpuno isto što i objekti Connection i Recordset. Objekt Command nema metodu Open i zato koristi objekt Connection izravno (eksplicitno), a ne posredno kao objekt Recordset. Prednosti objekta Command nad objektom Recordset su u tome što pomoću parametara imamo bolju kontrolu nad setom podataka.

Primjer objekta Command.

Najprije se kreira spoj s bazom pomoću connection stringa i objekta Connection...

```

<%
strConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Data\Inventory.mdb"
Set cnn = Server.CreateObject("ADODB.Connection")
cnn.Open strConnectionString

```

...zatim kreiramo instanciju objekta Command i koristimo njegovo svojstvo ActiveConnection za priključak objekta Command na objekt Connection...

```

Set cmn= Server.CreateObject("ADODB.Command")
Set cmn.ActiveConnection = cnn

```

Slijedi definicija SQL upita. SQL upit se pohranjuje u svojstvo CommandText objekta Command...

```

cmn.CommandText = "INSERT INTO Dnevnik (Ucenik,
Ocjena) VALUES (?, ?)"

```

Pohranimo predkompajliranu verziju upita specificiranog u svojstvu
CommandText prije nego objekt Command izvršimo prvi put...

```
cmn.Prepared = True
```

Definirajmo parametre upita...

```
cmn.Parameters.Append  
cmn.CreateParameter("ucenik_ime_i_prezime",adVarChar,  
,255 )  
cmn.Parameters.Append  
cmn.CreateParameter("ocjena",adVarChar, ,255 )
```

Definicija i prvo izvršenje naredbe INSERT INTO...

```
cmn("ucenik_ime_i_prezime ") = "Valentina Valentini"  
cmn("ocjena") = "5"  
cmn.Execute ,,adCmdText + adExecuteNoRecords
```

Definicija i drugo izvršenje naredbe INSERT INTO...

```
cmn("ucenik_ime_i_prezime ") = "Ivana Martini"  
cmn("ocjena") = "5"  
cmn.Execute ,,adCmdText + adExecuteNoRecords  
.  
.  
.  
%>
```

Prednosti Command nad Recordset objektom u gornjem primjeru vide se u tome što smo upit kreirali i kompajlirali samo jednom, a onda, po volji, ponavljali samo definiciju podataka koje unosimo u tablicu i izvršenje upita. Definicija podataka koje unosimo, izvršenje upita... koliko nam god puta treba.

Lekcija 8

Transaction

Što je transakcija u svim aplikacijama, pa tako i u distribuiranim? To je najmanja moguća programska cjelina koja uspije ili ne uspije. Neću uzeti za primjer banke i prijenos novca s jednog računa na drugi. Uzet ću drugi primjer. Na stranicama site www.e92.hr postoji forma s jednim tekstualnim poljem i gumbom tipa Submit. U polje možete upisati svoj e-mail. Kada pritisnete gumb pokrenuli ste HTTP zahtjev definiran u argumentu Action te forme. Neka je to zahtjev za ASP stranicom novipretplatnikemailliste.asp. HTTP zahtjev je na poslužitelju uhvatio ASP objekt Request i pohranio u memoriju poslužitelja. U VBScript ASP stranice slijede kreiranje objekta Connection, pa objekta Recordset koji izvršava SQL upit INSERT INTO... tablicu emailista od Access baze elist.mdb. Vaš e-mail je pohranjen u bazu. Slijedi objekt Response i njegova metoda Redirect koja šalje klijent stranicu zahvale i URL besplatne knjige.

U gornjem procesu se dogodila transakcija. Vaš e-mail je prešao put od browser-a do baze na poslužitelju. Na tom putu je moglo iskrsnuti mnogo zapreka u svim segmentima procesuiranja. Koja je u gornjoj priči najmanja moguća programska cjelina? I koliko ih ima? Ima samo jedna. Počela je na browser-u kada je korisnik pritisnuo gumb tip Submit, a završila je s naredbom zatvaranja (Close) objekta Connection nakon što je e-mail uspješno pohranjen u bazu na poslužitelju.

Kratka povijest aplikacija klijent – poslužitelj

U izgradnji aplikacija za tehnologiju klijent-poslužitelj. prvi zadatak projektanta aplikacije jest što će staviti na klijent stranu, a što na poslužitelj stranu. Mojim učenicima koji sad čitaju ove retke, i meni, najbliži je Access, pa ću na primjeru Access baze prikazati moguće dileme.

Varijanta A: debeli klijent - mršavi poslužitelj

Samo tablice možemo ostaviti na poslužitelju, a sve ostalo: upite, forme, module, izvješća i makroe staviti na klijent stranu.

Varijanta B: mršavi klijent – debeli poslužitelj

Ostaviti sve na poslužitelju, a samo linkom vezati klijente na forme i upite koji su potrebni od klijenta do klijenta.

Plus i minus varijante A.

Plus: manje je opterećena mreže jer aplikacija (Access baza) obavi gotovo sve poslove na klijent strani. Nema čekanja klijenta za pristup bazi.

Minus: skupo održavanje i promjene. Programer mora poći od klijenta do klijenta i obaviti planirane izmjene.

Plus i minus varijante B.

Plus: jeftinije održavanje jer treba intervenirati samo na jednom računalu, poslužitelju, i svi će klijenti istog trenutka vidjeti te promjene.

Minus: veće opterećenje mreže i moguća čekanja klijenata. Drugo, toliko se toga s godinama stavi na poslužitelja da je najjeftinije u pojedinim etapama razvoja novih tehnologija sve baciti. Malo ih za to ima novca, a i tehnologija se sve brže mijenja, pa novca za takav rad nema nitko.

S pojavom Interneta i distribuiranih aplikacija gornji nedostaci jedne i druge varijante su skočili na stotu potenciju. Više nas u mreži nema 10 nego 10 milijuna, i ne dolazi u obzir da naši klijenti čekaju na vezu, ili još gore, da je ne ostvare nikako. Isto tako nema govora o tome da naši programeri obilaze 10 milijuna klijenata i svako malo im mijenjaju software, sukladno promjenama tehnologije.

Uz to, porasla je potreba pojedinca, osobno i poslovno, za resursima mreže. Dok je nekad lokalnu mrežu koristilo samo računovodstvo, a ostali zaposlenici s menagementom nimalo, danas Internet koriste svi i u sve većoj mjeri, a računovodstvo još najmanje.

TCP/IP

To ne bi bilo moguće bez TCP/IP tehnologije, prije svega, a onda i COM tehnologije.

Što se tako revolucionarno dogodilo s TCP/IP tehnologijom? Dogodilo se sljedeće. Donijela je porast "konkurentnog" (istovremenog) korištenja istih resursa milijardu posto. Do jučer kao mogućnost nevjerojatno niti u teoriji.

Godine 1981. radio sam u timu ispitivača telefonske centrale za desetak tisuća pretplatnika. Koristit ću današnju terminologiju za priču u onom vremenu. Svaki pretplatnik je jedan URL (telefonski broj). Jednom URL istovremeno može prići samo jedan URL. Kada ste dobili sugovornika na drugoj strani, i vi i on ste bili zauzeti za cijeli svijet. Tek kada ste obavili razgovor i spustili slušalicu mogao vas je dobiti netko drugi. To je bilo tako jer još nije bilo TCP/IP tehnologije.

Bilo je to vrijeme prekidačke veze, bez obzira je li prekidač tranzistor ili bakrena žica. Svejedno. Kada se ostvari veza između dva broja, ostvarila se kontinuirana bakrena žica između vas i sugovornika ma koliko udaljeni bili. Milijuni prekidača na milijunima centrala se zatvorilo i gradi tu vezu. Jedna veza, a toliko je resursa zarobila. Tone i tone bakra i silicija u jednoj jedinjoj vezi. Sada je jasno zašto se telefon tada smatrao velikom povlasticom, jer se takvo što moralo odraziti na cijenu.

I dok sam ja radio na toj centrali, moja generacija kolega na MIT-u je izmišljala TCP/IP i uspjela. Oni su vidjeli da je "*nonsens*" trošiti toliko bakra na jednu vezu (i njima je dosta generacija odraslo na bakru jer takvo stanje je postojalo od izuma telegrafa) i zaključili kako je konačno moguće sjeckati informaciju u male segmente, umjesto što je bakrena veza isjeckana u prekidače.

I tako se današnjim mrežama naš govor prenosi na daljinu isjeckan u milijarde sitnih paketića koji čak ne putuju držeći se za ruke, nego se kao rakova djeca razmire po Zemlji, da bi se ipak našli na drugoj strani Zemlje i složili u cjelinu kao da nikad nisu ni bili rasuti.

Rezultat toga je fiksna bakrena žica, bez prekidača, po kojoj "trče" milijuni paketića konkurentno, rame uz rame, svaki paketić neka tuđa informacija, nečiji tuđi HTTP zahtjev, neki tuđi TV prijenos ili GIF animacija. Milijuni konkurentnih pristupa istom URL.

COM

I sada, gornjoj tehnologiji dodajmo COM tehnologiju pa smo dobili distribuiranu aplikaciju. Idemo redom.

Klijent

Klijent je jedna COM aplikacija, jednaka na svim računalima. Ono što ne može napraviti sama, iza scene, brzo i neprimjetno downloada COM softvere koji to može napraviti. Klijent služi samo za prezentaciju sadržaja. Zahvaljujući univerzalnosti HTML skripta, svi klijenti znaju primiti i prezentirati sadržaje koji im dolaze. Klijentima samo dolazi HTML. Ako ikada i dođe nešto drugo, ništa zato; proizvođač se pobrinuo da sa sadržajem na klijent dođe i software koji će "to nešto" interpretirati.

Poslužitelj

Klijentu sve dođe gotovo s poslužitelja. To znači da se ukupno procesuiranje današnjih aplikacija događa na poslužiteljskoj strani. Kako poslužitelj uspijeva obaviti posao na zadovoljstvo milijuna konkurentnih klijenata, koji uz to ne čekaju više ni tren (ADSL, kabelske veze, satelitske itd.)?

Poslužitelj to uspijeva opet pomoću "isjeckanosti". Ovaj put su isjeckani poslovi i procesi. Pod poslovima smatramo niz naredbi u jednom programu. Pod procesom pak smatramo izolirani prostor u kojem se procesuiraju taj program. Jedan glomazan program koji bi imao u IIS-u isto tako glomazan proces, u kojem bi naredbe sporo dolazile na red, isjeckan je na puno malih programa koje nazivamo komponentama (kada je riječ o Microsoft tehnologiji uvijek mislimo na COM sukladne komponente).

Procesi su isjeckani na *threadove*, tj. operacijski sustav Windows radi u threadovima. Windows upravlja procesorom. Procesor procesuiraju (izvršava) naredbe programa. Zahvaljujući threadovima Windows može napustiti jedan proces i skočiti u drugi, ili unutar jednog procesa skakati od *thread* do *thread* prema planu programera.

VBScript u ASP stranicama poziva jednu komponentu koja će pozvati drugu komponentu, koja će pozvati bazu i pronaći pretplatnika, vratiti tog pretplatnika prvoj komponenti, a ova VBScriptu. I tako milijuni konkurentnih zahtjeva. Nije svejedno je li jedan posao jedna komponenta u svom posebnom threadu ili je deset poslova u jednoj komponenti. Jer naše ASP stranice uvijek traže jedan po jedan

posao, jer se naredbe VBScripta izvršavaju jedna po jedna. Poslovi su metode u Class formi. Puno je bolje napraviti deset metoda u deset različitih komponenti nego svih deset u jednoj komponenti. Jer kad ASP stranica traži određenu metodu u komponenti, brže će je naći ako je to i jedina metoda u toj komponenti.

Gledano obratno, komponenta s jednom metodom obavi puno više konkurentnih poslova nego komponenta s deset metoda.

Nameće mi se sljedeća usporedba. Tko će više kurirskih poslova obaviti u gradu. Kurir na motoru ili kurir u kamionu s prikolicom.

Web aplikacija u tri sloja

Cijelo vrijeme govorimo o klijentu i poslužitelju, i na koju stranu što i koliko staviti. A strana je tradicionalno uvijek bilo dvije: klijent i poslužitelj. Koristio se termin i aplikacije u dva sloja. Pojavom Web-a, a kasnije i COM tehnologije, kao najbolje se rješenje pokazala distribuirana aplikacija u tri sloja: prezentacijski sloj, sloj izvora podataka i sloj biznis logike.

Prezentacijski sloj aplikacije se događa na klijentu, gore smo vidjeli kako. Tu više nema nedoumica. Na kraju se čak dogodilo da gotovo cijeli svijet koristi browser istog proizvođača. Prezentacijski sloj aplikacije dolazi s poslužitelja i u prezentacijskom sloju nema puno procesuiranja. Možda malo manipulacije sa stringom kojeg unosi korisnik u formu na HTML stranici, radi provjere stringa na licu mjesta (na primjer, provjeru je li korisnik u svom e-mailu tipkao znak et @ je puno učinkovitije skript-programirati u prezentacijskom sloju, nego na poslužitelju i trošiti resurse Interneta za nekoliko HTTP zahtjeva samo zato što će poslužitelj vraćati zahtjev radi takvih tipografskih grešaka na klijentu).

Dok nam je prezentacijski sloj postao standardiziran i svi ga koriste na isti način, druga dva sloja još uvijek otimaju poslove jedan drugome.

Sloj izvora podataka je najčešće baza podataka. Svakako klijent-poslužitelj baza, a ne desktop baza. Baze, vidjeli smo u slučaju Access-a, nisu samo tablice (kontejneri podataka). I Access i SQL Server baze leže na porodici SQL skripta u kojem se mogu programirati upiti, procedure. Access ima i komponentu Modules u kojoj možemo pisati procedure u VBA programu.

Ono što želim reći jest sljedeće: cjelovito procesuiranje jedne distribuirane aplikacije moguće je staviti u bazu i za biznis sloj ne ostaviti ništa. Što se time dobiva, a što gubi?

Plus: brže izvršavanje procesa U BAZI jer je sve tu: i podaci i programski kôd, pa nema gubljenja vremena na komunikaciju podaci – biznis sloj (ASP, VBScript, komponente).

Minus: radi toga što baza radi sve, ona je i dulje zauzeta po jednom zahtjevu, pa takva aplikacija u cijelosti ima nižu granicu konkurentnih zahtjeva. To je ono što najmanje želimo.

Minus: s vremenom takvoj aplikaciji veoma oteža sloj izvora podataka da postaje skupa i za unapređenja i promjene.

Biznis sloj su komponente i ASP stranice. U taj sloj ide poslovna logika i logika spajanja na baze podataka. Biznis sloj se u cijelosti može izvesti u VBScriptu i na ASP stranicama, samo uz pomoć ASP ugrađenih komponenti. Znači, takva distribuirana aplikacija bi opteretila programera samo za rad ASP stranica i izradu baze podataka.

Više je prednosti, međutim, u kombinaciji ASP skripta i kompajliranih komponenti. Evo zašto:

Puno je sigurnija web aplikacija sa sto komponenti i jednom ASP stranicom, nego ona sa sto ASP stranica i ni jednom komponentom. ASP stranice "leže" na poslužitelju kao otvorena knjiga za zlonamjerne, a i teže je ukupno administriranje te gomile otvorenog skripta.

Komponente su ista ta biznis logika, gotovo isti kôd, samo kompajliran i kao takav brži. Komponente su "crna kutija" za sve koji bi ih se zlonamjerno dočepali, i naša biznis logika je nedostupna našoj konkurenciji. Programerska kuća će graditi knjižnicu gotovih komponenti i uvijek ih iznova koristiti u novim narudžbama. Komponente pojeftinjuju proizvodnju web aplikacija.

Bilo da ste biznis sloj napravili samo u ASP skriptiranju ili kombinirano ASP i komponente, imamo gubitke vremena radi dodatne komunikacije ona dva sloja sada s tim trećim slojem.

Tehnologija COM se nastavlja ubrzano razvijati tako da je već danas najbolje rješenje distribuiranih aplikacija ono u tri sloja.

Vratimo se transakciji

S verzijom IIS 4 Microsoft je izbacio OptionPack NT4 i u njemu Microsoft Transaction Server. Već u sljedećoj verziji IIS 5, isti Server naziva COM+ i u IIS 5.1 Component Services. U prozoru u kojem administriramo taj je Server ostao cijelo vrijeme gotovo nepromijenjen.

Prozor Component Services u Windows XP ćete otvoriti ovako:

1. Start/Control Panel/Administrative Tools/Component Services. Ako ga nema, instalirajte ga s CD-a Windowsa XP.

Od prve verzije do današnje, ovaj Server ima zadatak osloboditi programera u cijelosti bilo kakve brige o transakciji kako smo je definirali na početku ove lekcije.

Ukoliko bilo gdje u transakciji dođe do greške, okruženje će to *uhvatiti* i umjesto programera obaviti operaciju *rollback* ili povratak na stanje kakvo je bilo prije nego je transakcija počela. Nešto slično naredbi Undo, koju smo naučili u Wordu i Excelu.

Budući da je distribuirana aplikacija u tri sloja općeprihvaćena kao najbolje rješenje, transaction okruženje (Component Services) ima sve više posla. Evo zašto. Prezentacijski sloj su ASP stranice koje leže na poslužitelju i na zahtjev HTML stranice šalju klijentu. ASP stranice komuniciraju međusobno, kao i s ugrađenim ASP komponentama, s IIS komponentama, komponentama operacijskog sustava i s našim osobnim komponentama koje smo napravili samo za potrebe naše web aplikacije. Komponente u biznis sloju komuniciraju s ASP stranicama i s bazom podataka. Gledano s odmakom, naša web aplikacija zapravo je puno malih segmenata koji međusobno komuniciraju.

Zatim nastupa Transaction Server (MTS) ili Component Services (CS). On vodi brigu o svim djelićima web aplikacije kao cjeline. Dvije komponente koje puno komuniciraju međusobno on će staviti u isti proces, a neku drugu komponentu će staviti u onaj povoljniji *thread* itd. Takvo run-time okruženje, koje brine u koji će proces i u koji *thread* staviti koju komponentu, umnogome oslobađa projektanta aplikacije.

MTS vodi brigu i o ASP stranicama, kao i o komponentama. Jedna transakcija može početi na jednoj ASP stranici, nastaviti se u komponentama biznis sloja i

završiti na drugoj ASP stranici. MTS vodi brigu o procesima i *threadovima* svih sudionika transakcije.

ObjectContext

Run-time okruženje se ostvaruje pomoću ASP objekta *ObjectContext*.

Prijavljivanje tekućoj transakciji ostvaruje se preko ovog objekta. Objekt ima dvije metode kojima ASP skript može pristupiti:

Metoda

Opis

SetComplete metoda deklarira "transakcija je obavljena u redu". Ova se metoda implicitno automatski poziva kad završi procesuiranje ASP stranice. Odnosi se samo na skript u ASP stranici. Istu metodu će pozvati i sve komponente koje sudjeluju u transakciji ukoliko u redu prođe njihov dio posla u transakciji. Ako svi sudionici nakon obavljenog posla generiraju ovu metodu, znači da je transakcija uspjela.

SetAbort metoda javlja da transakcija inicirana u skriptu nije uspjela i da baza podataka neće biti ažurirana.

Component Services brine samo o bazama podataka, i to bazama klijent-server, kao što su SQL Server baze, a ne podržava Access baze. Također, promjene podataka na tekstualnim datotekama, Excelu i sl. ne spadaju u nadležnost objekta *ObjectContext* i nema *rollback* ako transakcija ne uspije.

Evo primjera ASP skripta na ASP stranici od naredbe `@TRANSACTION` pa do metoda `SetComplete` i `SetAbort` objekta *ObjectContext*:

```
<%@ TRANSACTION = Required %>
```

Ova ASP naredba mora biti prvi redak skripta, a to je objektu *ObjectContext* dovoljno da uzme skript u svoje okruženje.

```
...
```

```
<%
```

```
If strStanje = "NedovoljnoNaZalihama"
```

```
    ObjectContext.SetAbort
```

```
Response.Write "U ovom trenutku ovu transakciju ne  
možemo obaviti"
```

```
Else
```

```
ObjectContext.SetComplete
```

```
isporuka = Request("isporuka")
```

```
Azuriranje = objProdaja.Isporuka(isporuka)
```

```
End If
```

```
%>
```

U gornjem primjeru ASP skript je kreirao varijablu u koju je pohranio podatak koji je mogao dobiti i od komponente. Na osnovu vrijednosti te varijable programer ispisuje ASP skript u kojem eksplicitno poziva metodu SetComplete ili metodu SetAbort. Programer utvrđuje: ako je vrijednost te varijable tolika, transakcija je uspjela, a ako nije tolika, transakcija nije uspjela.

ObjectContext ima i dva događaja koji će automatski pozvati sljedeće metode objekta ObjectContext: OnTransactionCommit i OnTransactionAbort.

Otvorite u browser-u iz IIS 5.1 prozora sljedeći Microsoftov primjer:

http://localhost/inetpub/iissamples/sdk/asp/transactional/SimpleTransaction_VBScrip.asp

Najvjerojatnije ćete ugledati ovaj tekst, jer transakcija nije imala nikakvog razloga neuspjeti budući da se odnosi samo na nekoliko redaka HTML tagova.

The Transaction just comitted.This message came from the OnTransactionCommit() event handler.

ASP stranica je počela s naredbom...

```
<%@ TRANSACTION = Required LANGUAGE = "VBScript" %>
```

```
<HTML>
```

```
...nekoliko redaka samo HTML tagova...
```

```
</HTML>
```

Na kraju su navedene subrutine. Jedna od tih subrutina će sigurno biti pozvana. AkoObjectContext registrira bilo kakvu grešku u ASP stranici, pozvat će subrutinu Sub OnTransactionAbort(); u suprotnome će pozvati subrutinu Sub OnTransactionCommit().

```
<%
```

```
Sub OnTransactionCommit()
```

```
Response.Write "<p><b>The Transaction just  
comitted</b>."
```

```
Response.Write "This message came from the "
```

```
Response.Write "OnTransactionCommit() event handler."
```

```
End Sub
```

```
Sub OnTransactionAbort()
```

```
Response.Write "<p><b>The Transaction just  
aborted</b>."
```

```
Response.Write "This message came from the "
```

```
Response.Write "OnTransactionAbort() event handler."
```

```
End Sub
```

```
%>
```

Lekcija 9

Profesionalni primjer E- knjižara Kartelo ASP stranice

Najprije ću prikazati cijeli kôd web aplikacije *E-knjižara Kartelo*. Počnimo od ASP stranica.

ekartelo.asp

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<% Response.Expires = 0 %>

<html>
<head>
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
```

```
<title></title>
<link rel="stylesheet" type="text/css" href="glavni.css">
</head>

<body>

<% dim stanje %>
<% dim objPomocni %>
<% dim objPodaciKupca %>
<% dim objZanimaSeZa %>
<% dim objZanimaSeZa2 %>
<% dim objZanimaSeZaOvo2 %>
<% dim objOdgovor %>
<% dim objKosare %>
<% dim objKupci %>
<% dim objKatalog %>
<% dim objElementi %>
<% dim objKnjiga %>
<% dim vPodaci %>

<table width=100% cellspacing=0 cellpadding=0 border=0>
<tr class="zaglavljestranice">
<th align=left>&nbsp;&nbsp;&nbsp;E-knji&#382ara Kartelo
</th>
</tr>
</table>
```

```

<% if not Request.QueryString("stanje") = "" then %>
<%     stanje = Request.QueryString("stanje") %>
<% else %>
<%     if Session("posjetitelj") = true then %>
<%         stanje = 3 %>
<%     else %>
<%         stanje = 1 %>
<%     end if %>
<% end if %>

<% if stanje > 2 and Session("posjetitelj") = false then %>
<%     stanje = 1 %>
<% end if %>

<% if stanje = 6 then %>

<%     set objKosare =
Server.CreateObject("eKarteloASPKosarice.Kosarice") %>

<%     set objOdgovor = objKosare.AddItem(Session("narudzbaID"),
Session("email"), _
        Request.QueryString("katalogid"), 1) %>

<%     if objOdgovor("greska") then %>

<!--#include file=error.inc-->

<%     else %>

<%     Session("narudzbaID") = objOdgovor("narudzbaID") %>

<p>Vas odabir je dodan u kosaru.</p>

```



```

<hr>
<input type="button" value="Kupovina"
      onclick="window.history.back(1)">
<input type="button" value="Kosara"
      onclick="window.document.location.href='pogleduKosaru.asp'">
<input type="button" value="Blagajna"
      onclick="window.document.location.href='blagajna.asp'">
<%   end if %>
<%   set objKosare = nothing %>
<% end if %>

<% if stanje = 5 then %>
  <i>
  <b>Katalo&#353ki broj knjige je:
  <% = Request.QueryString("katalogid") %>
  </b>
  <a href="knj/<% = Request.QueryString("katalogid")
  %>.htm">Op&#353irni je o knjizi</a>
  </i>
<% end if %>

<% if stanje = 4 then %>
  <%   set objPomocni =
  Server.CreateObject("eKarteloASPPomocna.NovaKolekcija") %>
  <%   set objZanimaSeZaOvo2 = objPomocni.NovaKolekcija() %>

```

```

<%   for each objZanimaSeZa in Session("zanimaseza") %>
<%       set objZanimaSeZa2 = objPomocni.NovaKolekcija() %>
<%   objZanimaSeZa2.Add objZanimaSeZa("naziv"),"naziv" %>
<%   objZanimaSeZa2.Add objZanimaSeZa("kod"),"kod" %>
<%       if Request.Form(objZanimaSeZa("naziv")) = "" then %>
<%           objZanimaSeZa2.Add false,"ukljuceno" %>
<%       else %>
<%           objZanimaSeZa2.Add true,"ukljuceno" %>
<%       end if %>
<%       objZanimaSeZaOvo2.Add objZanimaSeZa2 %>
<%       set objZanimaSeZa2 = nothing %>
<%   next %>

<%   set objKatalog =
Server.CreateObject("eKarteloASPKatalog.Katalog") %>

<%   set objOdgovor = objKatalog.VratiKatalogListupoZanimaseza(
objZanimaSeZaOvo2 ) %>

<%   if objOdgovor("greska") then %>

<!--#include file=error.inc-->

<%   else %>

<%       set objElementi = objOdgovor("elementi") %>

<table>

<%       for each objKnjiga in objElementi %>

<tr>

<td rowspan=2>

<%           Response.Write "<img border=1 src=sl/" &
objKnjiga.item("slikaur1") & ">" %>

```

```

        </td>

        <td>

        <font size=4>

<%           Response.Write "<b>" & objKnjiga.item("naslov") &
"</b>" %>

<%           if objKnjiga.item("novada") = true then %>

<%           Response.Write "<img src=sl/nova.gif>" %>

<%           end if %>

        </font>

        </td>

        <td>

        </td>

        <td>

<%           dim hlink %>

<%           hlink = "detaljiKnjige.asp?katalogid=" & _
objKnjiga.item("katalogid") %>

<%           Response.Write "<input type=""button"" value=""Kupujem
za " & _
objKnjiga.item("danasnjacijena") & "" " & _
"onclick=window.document.location.href='" & _
"dodajKnjigu.asp?katalogid=" & objKnjiga.item("katalogid")
& _
"' > " %>

<%           if objKnjiga.item("promidzbada") = true then %>

<%           Response.Write "<img src=sl/promidzba.gif>" %>

```



```

<%     end if %>

<center>Kraj liste</center>

<%     set objKatalog = nothing %>

<%     set objZanimaSeZaOvo2 = nothing %>

<% end if %>

<% if stanje = 3 then %>

    <form action="ekartelo.asp?stanje=4" method="POST">

    <p>Koju kategoriju knjiga zelite vidjeti?</br>

    <% vPodaci = "" %>

    <% for each objZanimaSeZa in Session("zanimaseza") %>

        <%     vPodaci = vPodaci & "<input type=" & chr(34) &
        "checkbox" & chr(34) %>

        <%     if objZanimaSeZa("ukljuceno") then %>

        <%         vPodaci = vPodaci & " checked " %>

        <%     end if %>

        <%     vPodaci = vPodaci & " name = " & chr(34) &
        objZanimaSeZa("naziv") & chr(34) %>

        <%     vPodaci = vPodaci & " value = " & chr(34) &
        objZanimaSeZa("naziv") &
        "&nbsp;&nbsp;&nbsp;" & chr(34) %>

        <% next %>

    <% Response.Write vPodaci %>

    <p>

    <hr>

```

```

<input type="submit" name="List" value="Lista knjiga">
    <input type="button" value="Kosara"
    onclick="window.document.location.href='pogleduKosaru.asp'">
    <input type="button" value="Blagajna"
    onclick="window.document.location.href='blagajna.asp'">
</form>
</form>
<% end if %>

<% if stanje = 2 then %>

    <% if Request.Form("knjizaral") = "1" then %>
    <%     Session("email") = Request.Form("email") %>
    <%     Session("zaporka") = Request.Form("zaporka") %>
    <%     if Session("email") = "" or Session("zaporka") = "" then
%>
        <p class=porukaogresci>Potrebno je ispuniti oba
polja</p>
    <%         stanje = 1 %>
    <%     end if %>
    <% end if %>

    <% if stanje = 2 then %>

    <% set objPomocni =
Server.CreateObject("eKarteloASPPomocna.NovaKolekcija") %>

```

```

<% set objPodaciKupca = objPomocni.NovaKolekcija() %>
<% objPodaciKupca.Add Session("email"),"email" %>
<% objPodaciKupca.Add Session("zaporka"),"zaporka" %>

<% set objKupci = Server.CreateObject("eKarteloASPKupci.Kupci")
%>

<% set objOdgovor = objKupci.ProvjeriKupca(objPodaciKupca) %>

<% set objPodaciKupca = nothing %>
    <% set objKupci = nothing %>
<% set objPomocni = nothing %>

<% if objOdgovor("greska") then %>
<!--#include file=error.inc-->

    <% else %>

        <% if objOdgovor("postoji") and objOdgovor("zaporkatocna")
then %>

            <% set objPodaciKupca = objOdgovor("podacikupca") %>
            <% Session("posjetitelj") = True %>
            <% Session("email") = objPodaciKupca("email") %>
            <% Session("ime") = objPodaciKupca("ime") %>
            <% Session("prezime") = objPodaciKupca("prezime") %>
            <% Session("adresa") = objPodaciKupca("adresa") %>
            <% Session("grad") = objPodaciKupca("grad") %>
            <% Session("postanskibroj") =
objPodaciKupca("postanskibroj") %>

```

```

<%      Session("drzava") = objPodaciKupca("drzava") %>

<%      Session("telefon") = objPodaciKupca("telefon") %>

<%      Session("koristicookie") =
objPodaciKupca("koristicookie") %>

<%      set Session("zanimaseza") = objPodaciKupca("zanimaseza")
%>

<%      set objPodaciKupca = nothing %>

<p>Dobro do&#353li

<%      Response.Write Session("ime") & " " & Session("prezime")
%>

</p><hr>

      <input type="button" value="A&#382uriraj podatke"

      onclick="window.document.location.href='registracija.asp?stanje
=2'">

      <input type="button" value="Idem u kupovinu"

      onclick="window.document.location.href='ekartelo.asp?stanje=3'"
>

<%      else %>

<%      Session("posjetiteljBroj") = Session("posjetiteljBroj")
+ 1 %>

<%      if Session("posjetiteljBroj") > 3 then %>

      <p>Ako ste zaboravili svoju zaporku javite nam
na

      <a href="mailto:skola-e-92@st.tel.hr">email</a></p>

<%      else %>

      <p>Vi niste u na&#353oj bazi ili je va&#353a
zaporka pogre&#353na</p>

```



```

                <hr>
                <input type="button" value="Poku&#353ajte
ponovo"
                onclick="window.document.location.href='ekartelo.asp?stanje=1'"
                >
                <input type="button" value="Registracija"
                onclick="window.document.location.href='registracija.asp?stanje
=1'">
                <%     end if %>
                <%     end if %>
                <% end if %>
                <% end if %>
                <% end if %>
                <% end if %>
                <% if stanje = 1 then %>
                <p>Ako su va&#353i podaci ve&#263 u na&#353oj bazi, molimo vas
prona&#273ite se
                <form action="ekartelo.asp?stanje=2" method="POST">
                <table border="0">
                <tr>
                <td>Email</td>
                <td><input type="text" size="20" name="email"
                value="<% =Session("email") %>"></td>

```

```

        </tr>

        <tr>

            <td>zaporka</td>

            <td><input type="password" size="20" name="zaporka"
value="<% =Session("zaporka") %>"></td>

        </tr>

    </table>

    <input type="hidden" name="knjizara1" value="1">

    <hr>

    <input type="submit" name="submit" value="Ja sam ve&#263 u
va&#353oj bazi">

    <input type="button" value="Ja sam novi(a) - registrirajte me"

    onclick="window.document.location.href='registracija.asp?stanje
=1'">

    </form>

<% end if %>

</body>

</html>

```

registracija.asp

```

<%@ LANGUAGE="VBSCRIPT" %>

<% Option Explicit %>

<% Response.Expires = 0 %>

<% if stanje = 3 then %>

```

```
<% if Request.Form("koristicookie") <> "" then %>
<%     Response.Cookies("email") = Session("email") %>
<%     Response.Cookies("email").expires = #1/1/2004# %>
<%     Response.Cookies("zaporka") = Session("zaporka") %>
<%     Response.Cookies("zaporka").expires = #1/1/2004# %>
<% end if %>
<% end if %>
```

```
<html>
```

```
<head>
```

```
<meta HTTP-EQUIV="Pragma" CONTENT="no-cache">
```

```
<title>Home</title>
```

```
<link rel="stylesheet" type="text/css" href="glavni.css">
```

```
</head>
```

```
<body bgcolor=Cyan>
```

```
<% Dim stanje %>
```

```
<% Dim objKupci %>
```

```
<% Dim objPomocni %>
```

```
<% Dim objPodaciKupca %>
```

```
<% Dim objOdgovor %>
```

```
<% Dim objZanimaSeZa %>
```

```
<% Dim vPodaci %>
```

```
<table width=100% cellspacing=0 cellpadding=0 border=0>
```

```
<tr class="zaglavljestranice">
<th align=left>&nbsp;Registracija za eKartelo
</th>
</tr>
</table>
```

```
<% if not Request.QueryString("stanje") = "" then %>
<%   stanje = Request.QueryString("stanje") %>
<% else %>
<%   if Session("posjetitelj") = true then %>
<%     stanje = 2 %>
<%   else %>
<%     stanje = 1 %>
<%   end if %>
<% end if %>
```

```
<% if stanje = 2 then %>
```

```
  <% if Request.Form("registracija1") = "1" then %>
  <%   Session("email") = Request.Form("email") %>
  <%   Session("zaporka") = Request.Form("zaporka") %>
  <%   if Session("email") = "" or Session("zaporka") = "" then
  %>
  <%     stanje = 1 %>
  <%     <p class=porukaogresci>Potrebno je ispuniti oba
  polja.</p>
```

```

<%     else %>

<%     set objPomocni =
Server.CreateObject("eKarteloASPPomocna.NovaKolekcija") %>

<%         set objPodaciKupca = objPomocni.NovaKolekcija() %>

<%         objPodaciKupca.Add Session("email"),"email" %>

<%         objPodaciKupca.Add Session("zaporka"),"zaporka" %>

<%     set objKupci =
Server.CreateObject("eKarteloASPKupci.Kupci") %>

<%         set objOdgovor =
objKupci.RegistrirajKupca(objPodaciKupca) %>

<%         if objOdgovor("greska") then %>
<!--#include file=error.inc-->

<%             else %>

<%                 if objOdgovor("postoji") then %>

<%                     if objOdgovor("zaporkatocna") = false then %>
<p class=porukaogresci>Email postoji / zaporka
pogresna</p>

<%                         stanje = 1 %>

<%                     end if %>

<%                 end if %>

<%             if stanje = 2 then %>

<%                 set objPodaciKupca = objOdgovor("podacikupca") %>

<%                     Session("posjetitelj") = True %>

<%                     Session("email") = objPodaciKupca("email") %>

<%                     Session("ime") = objPodaciKupca("ime") %>

<%                     Session("prezime") = objPodaciKupca("prezime") %>

<%                     Session("adresa") = objPodaciKupca("adresa") %>

```

```

<%           Session("grad") = objPodaciKupca("grad") %>

<%           Session("postanskibroj") =
objPodaciKupca("postanskibroj") %>

<%           Session("drzava") = objPodaciKupca("drzava") %>

<%           Session("telefon") = objPodaciKupca("telefon") %>

<%           Session("koristicookie") =
objPodaciKupca("koristicookie") %>

<%           set Session("zanimaseza") =
objPodaciKupca("zanimaseza") %>

<%           end if %>

<%           end if %>

<%           set objKupci = Nothing %>

<%           set objOdgovor = Nothing %>

<%           set objPodaciKupca = Nothing %>

<%           end if %>

<% end if %>

<% if stanje = 2 then %>

<p>Molimo vas, unesite svoje podatke</p>

<form action="registracija.asp?stanje=3" method="POST">

    <table border="0">

        <tr>

            <td align="right">Email:</td>

            <td><% =Session("email") %></td>

```

```

</tr>

<tr>
    <td align=right>Ime:</td>
    <td><input type="text" size="20" name="ime"
value="<% = Session("ime") %>" ></td>

    <td align=right>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Prezime:</td>
    <td><input type="text" size="20" name="prezime"
value="<% = Session("prezime") %>" ></td>

</tr>

<tr>
    <td align=right>Adresa:</td>
    <td><input type="text" size="20" name="adresa"
value="<% = Session("adresa") %>" ></td>

    <td align=right>Dr&#382ava:</td>
    <td><input type="text" size="20" name="drzava"
value="<% = Session("drzava") %>" ></td>

</tr>

<tr>
    <td align=right>Grad:</td>
    <td><input type="text" size="20" name="grad"
value="<% = Session("grad") %>" ></td>

    <td align=right>Po&#353tanski broj:</td>
    <td><input type="text" size="20" name="postanskibroj"
value="<% = Session("postanskibroj") %>" ></td>

</tr>

<tr>

```

```

        <td align=right>Telefon:</td>

        <td><input type="text" size="20" name="telefon"
value="<% = Session("telefon") %>" ></td>

        <td align=right>Zaporka:</td>

        <td><input type="password" size="20" name="zaporka"
value="<% = Session("zaporka") %>" ></td>

    </tr>

</table>

<p>&#381elite li pohraniti svoje podatke za registraciju u
cookie?

    <input type="checkbox" size="20" name="koristicookie"

<%    if Session("koristicookie") = "Y" then %>
        checked
<%    end if %>

value="Y">

    <p>&#352to vas zanima od ovih opcija?</br>

<% vPodaci = "" %>

<% for each objZanimaSeZa in Session("zanimaseza") %>
    <%    vPodaci = vPodaci & "<input type=" & chr(34) &
"checkbox" & chr(34) %>

    <%    if objZanimaSeZa("ukljuceno") then %>

    <%        vPodaci = vPodaci & " checked " %>

    <%    end if %>

    <%    vPodaci = vPodaci & " name = " & chr(34) &
objZanimaSeZa("naziv") & chr(34) %>

    <%    vPodaci = vPodaci & " value = ""Y" > " %>

    <%    vPodaci = vPodaci & objZanimaSeZa("naziv") &
"&nbsp;&nbsp;" %>

```



```

        <% next %>

    <% Response.Write vPodaci %>

        <hr>

    <input type="submit" name="Update" value="A&#382uriraj">

        <input type="button" value="Idem u kupovinu"

onclick="window.document.location.href='OdabirKategorija.asp'">

    </form>

<% end if %>

<% end if %>

<% if stanje = 3 then %>

    <% Session("ime") = Request.Form("ime") %>

    <% Session("prezime") = Request.Form("prezime") %>

    <% Session("adresa") = Request.Form("adresa") %>

    <% Session("grad") = Request.Form("grad") %>

    <% Session("postanskibroj") = Request.Form("postanskibroj") %>

    <% Session("drzava") = Request.Form("drzava") %>

    <% Session("telefon") = Request.Form("telefon") %>

    <% Session("zaporka") = Request.Form("zaporka") %>

    <% If Request.Form("koristicookie") = "" then %>

    <%     Session("koristicookie") = "N" %>

    <% else %>

    <%     Session("koristicookie") = "Y" %>

```

```

<% end if %>

<% for each objZanimaSeZa in Session("zanimaseza") %>
<%   objZanimaSeZa.Remove "ukljuceno" %>
<%   if Request.Form(objZanimaSeZa("naziv")) = "" then %>
<%     objZanimaSeZa.Add false, "ukljuceno" %>
<%   else %>
<%     objZanimaSeZa.Add true, "ukljuceno" %>
<%   end if %>
<% next %>

<% set objPomocni =
Server.CreateObject("eKarteloASPPomocna.NovaKolekcija") %>
<% set objPodaciKupca = objPomocni.NovaKolekcija() %>
<% objPodaciKupca.Add Session("email"),"email" %>
<% objPodaciKupca.Add Session("zaporka"),"zaporka" %>
<% objPodaciKupca.Add Session("ime"),"ime" %>
<% objPodaciKupca.Add Session("prezime"),"prezime" %>
<% objPodaciKupca.Add Session("adresa"),"adresa" %>
<% objPodaciKupca.Add Session("grad"),"grad" %>
<% objPodaciKupca.Add Session("postanskibroj"),"postanskibroj"
%>
<% objPodaciKupca.Add Session("drzava"),"drzava" %>
<% objPodaciKupca.Add Session("telefon"),"telefon" %>
<% objPodaciKupca.Add Session("koristicookie"),"koristicookie"
%>
<% objPodaciKupca.Add Session("zanimaseza"),"zanimaseza" %>

```

```

    <% set objKupci = Server.CreateObject("eKarteloASPKupci.Kupci")
    %>

    <% set objOdgovor = objKupci.AzurirajKupca(objPodaciKupca) %>

    <% if objOdgovor("greska") then %>
<!--#include file=error.inc-->

    <% else %>

        <p>Podaci kupca <% = Session("ime")%>&nbsp;&nbsp;&nbsp;<% =
Session("prezime")%> su azurirani</p>

        <hr>

        <input type="button" value="Idem u kupovinu"

onclick="window.document.location.href='OdabirKategorija.asp'">

    <% end if %>

    <% Set objKupci = Nothing %>
    <% Set objPomocni = Nothing %>
    <% Set objPodaciKupca = Nothing %>

<% end if %>

<% if stanje = 1 then %>

<%
dim dio
for each dio in request.cookies
    response.write request.cookies(dio) & "<br>"

```

next

%>

<p>Knjizaru eKartelo mogu koristiti samo oni kupci koji su se registrirali kod nas.

Registrirati se mozete tako sto cete upisati svoj email i svoju zaporku.

Isti ce biti pohranjeni u nasu bazu podataka i moci cete ih koristiti kad god nam se

vratite.

```
<form action="registracija.asp?stanje=2" method="POST">
```

```
  <table border="0">
```

```
    <tr>
```

```
      <td>Email</td>
```

```
      <td><input type="text" size="20" name="email"
```

```
value="<% =Session("email") %>"></td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>Zaporka</td>
```

```
      <td><input type="password" size="20" name="zaporka"
```

```
value="<% =Session("zaporka") %>"></td>
```

```
    </tr>
```

```
  </table>
```

```
  <input type="hidden" name="registracijal" value="1">
```

```
<hr>

    <input type="submit" name="submit" value="Ovo su moji podaci
za ulazak u eKartelo">

</form>

<% end if %>

</body>

</html>
```

provjeraregistracije.asp

```
<%@ LANGUAGE="VBSCRIPT" %>

<% Option Explicit %>

<% Response.Expires = 0 %>

<%     if Session("posjetitelj") = true then %>
<%         Response.redirect "OdabirKategorija.asp" %>
<%     else %>
<%         Response.redirect "registracija.asp?state=1" %>
<%     end if %>
```

PrikaziKnjige.asp

```
<%@ LANGUAGE="VBSCRIPT" %>

<% Option Explicit %>

<% Response.Expires = 0 %>

<% dim stanje %>
```

```

<% dim objPomocni %>
<% dim objZanimaseza %>
<% dim objZanimasezaItem %>
<% dim objZanimasezaList %>
<% dim objKatalog %>
<% dim objOdgovor %>
<% dim objElementi %>
<% dim objKnjiga %>

<html>
<head>
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<link rel="stylesheet" type="text/css" href="glavni.css">
</head>
<body>
<table width=100% cellspacing=0 cellpadding=0 border=0>
<tr class="zaglavljestranice">
<th align=left>&nbsp;E-knji&#382ara Kartelo
</th>
</tr>
</table>

<% set objPomocni =
Server.CreateObject("eKarteloASPPomocna.NovaKolekcija") %>

<% set objZanimasezaList = objPomocni.NovaKolekcija() %>

```

```

<%   for each objZanimaseza in Session("zanimaseza") %>
<%       set objZanimasezaItem = objPomocni.NovaKolekcija() %>
<%       objZanimasezaItem.Add objZanimaseza("naziv"),"naziv" %>
<%       objZanimasezaItem.Add objZanimaseza("kod"),"kod" %>
<%       if Request.Form(objZanimaseza("naziv")) = "" then %>
<%           objZanimasezaItem.Add false,"ukljuceno" %>
<%       else %>
<%           objZanimasezaItem.Add true,"ukljuceno" %>
<%       end if %>
<%       objZanimasezaList.Add objZanimasezaItem %>
<%       set objZanimasezaItem = nothing %>
<%   next %>

<%   set objKatalog =
Server.CreateObject("eKarteloASPKatalog.Katalog") %>

<%   set objOdgovor =
objKatalog.VratiKatalogListuPoZanimaseza(objZanimasezaList ) %>

<%   if objOdgovor("greska") then %>

<!--#include file=error.inc-->

<%   else %>

<%       set objElementi = objOdgovor("elementi") %>

<table>

<%       for each objKnjiga in objElementi %>

<tr>

<td rowspan=2>

<%           Response.Write "<img border=1 src=sl/" &
objKnjiga.item("slikaur1") & ">" %>

```

```

        </td>

        <td>

            <font size=4>

<%           Response.Write "<b>" & objKnjiga.item("naslov") &
"</b>" %>

<%           if objKnjiga.item("novada") = true then %>

<%           Response.Write "<img src=sl/nova.gif>" %>

<%           end if %>

            </font>

        </td>

        <td>

        </td>

        <td>

<%           Response.Write "<input type=""button"" value=""Kupi za
" & _
            objKnjiga.item("danasnjacijena") & "" " & _
            "onclick=window.document.location.href='" & _
            "dodajKnjigu.asp?katalogid=" & objKnjiga.item("katalogid")
& _
            "' > " %>

<%           if objKnjiga.item("promidzbada") = true then %>

<%           Response.Write "<img src=sl/promidzba.gif>" %>

<%           end if %>

        </td>

    </tr>

    <tr>

```



```
<% set objZanimasezaList = nothing %>
```

```
</body>
```

```
</html>
```

pogleduKosaricu.asp

```
<%@ LANGUAGE="VBSCRIPT" %>
```

```
<% Option Explicit %>
```

```
<% Response.Expires = 0 %>
```

```
<html>
```

```
<head>
```

```
<title></title>
```

```
<link rel="stylesheet" type="text/css" href="glavni.css">
```

```
</head>
```

```
<meta HTTP-EQUIV="Pragma" CONTENT="no-cache">
```

```
<body>
```

```
<% dim stanje %>
```

```
<% dim objKosarica %>
```

```
<% dim objElementi %>
```

```
<% dim objKnjiga %>
```

```
<% dim objOdgovor %>
```

```
<table width=100% cellspacing=0 cellpadding=0 border=0>
```

```

<tr class="zaglavljestranice">
<th align=left>&nbsp;&nbsp;&nbsp;Sadrzaj kosarice
</th>
</tr>
</table>

<%    set objKosarica =
Server.CreateObject("eKarteloASPKosarice.Kosarice") %>

<%    set objOdgovor =
objKosarica.VratiElementeKosarice(Session("narudzbaID"),
Session("email")) %>

<%    if objOdgovor("greska") then %>
<!--#include file=error.inc-->

<%    else %>

<%        set objElementi = objOdgovor("elementi") %>
        <p>Sadrzaj kosarice

<%        Response.Write Session("ime") & " " & Session("prezime")
%>

        </p>

<%        if objElementi.count = 0 then %>
        <p>Vasa kosarica je prazna</p>
        <hr>

<%        else %>

                <form action="azurirajKosaricu.asp" method="POST">
                <table cellspacing=0 cellpadding=0 border=0>
                <tr class="zaglavljetablice">
                <th>Knjiga</th>

```

```

        <th width=15>&nbsp;</th>
        <th>Jedinicna cijena</th>
        <th width=15>&nbsp;</th>
        <th>Kolicina</th>
    </tr>
<%
    for each objKnjiga in objElementi %>
        <tr class="podacitablice">
            <td>
<%
                Response.Write objKnjiga.item("naslov") %>
            </td>
            <td>&nbsp;</td>
            <td align=right>
<%
                Response.Write objKnjiga.item("danasnjacijena") %>
            </td>
            <td>&nbsp;</td>
            <td align=right>
                <input type=text size=3 name=
<%
                    Response.Write objKnjiga.item("kosaricaid") %>
                value=
<%
                    Response.Write objKnjiga.item("kolicina") %>
            >
            </td>
        </tr>
<%
    next %>
</table>
<hr>

```

```

        <input type="submit" name="clear" value="Obrisi sve">
        <input type="submit" name="update" value="Azuriraj">
        <input type="button" value="Idem u kupovinu"

onclick=window.document.location.href='OdabirKategorija.asp'>

        <input type="button" value="Blagajna"

onclick=window.document.location.href='DetaljiNarudzbe.asp'>

                </form>
<%         end if %>

<%     end if %>
<%     set objKosarica = nothing %>

</body>
</html>

```

OdabirKategorija.asp

```

<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<% Response.Expires = 0 %>

<html>
<head>
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">

```

```

</title></title>
<link rel="stylesheet" type="text/css" href="glavni.css">
</head>
<body>
<% dim objZanimaseza %>
<% dim vPodaci %>

<table width=100% cellpadding=0 cellspacing=0 border=0>
<tr class="zaglavljestranice">
<th align=left>&nbsp;Knji&#382ara eKartelo
</th>
</tr>
</table>

  <form action="PrikaziKnjige.asp" method="POST">
    <p>Koje vas kategorije knjiga zanimaju?</br>
    <% vPodaci = "" %>
    <% for each objZanimaseza in Session("zanimaseza") %>
      <% vPodaci = vPodaci & "<input type=" & chr(34) &
"checkbox" & chr(34) %>
      <% if objZanimaseza("ukljuceno") then %>
        <% vPodaci = vPodaci & " checked " %>
      <% end if %>
      <% vPodaci = vPodaci & " name = " & chr(34) &
objZanimaseza("naziv") & chr(34) %>
      <% vPodaci = vPodaci & " value = ""y"" > " %>
      <% vPodaci = vPodaci & objZanimaseza("naziv") &
"&nbsp;&nbsp;" %>

```

```

    <% next %>

    <% Response.Write vPodaci %>

    <p>

    <hr>

    <input type="submit" name="List" value="Prikaz knjiga">

        <input type="button" value="Ko&#353arica"
onclick="window.document.location.href='pogledUKosaricu.asp'">

        <input type="button" value="Ovo naru&#269ujem"
onclick="window.document.location.href='DetaljiNarudzbe.asp'">

    </form>

</body>
</html>

```

Global.asa

```

<SCRIPT LANGUAGE="VBScript" RUNAT="Server">

Sub Session_OnStart

    '==Visual InterDev Generated - DataConnection startspan==

    '--Project Data Connection

        Session("ekartelo_ConnectionString") =
"DSN=EKARTELO;Description=EKartelo;SERVER=(local);UID=sa;APP=Micro
soft (R) Developer Studio;WSID=A021760;DATABASE=ekartelo"

        Session("ekartelo_ConnectionTimeout") = 15

        Session("ekartelo_CommandTimeout") = 30

        Session("ekartelo_RuntimeUserName") = ""

        Session("ekartelo_RuntimePassword") = ""

    '==Visual InterDev Generated - DataConnection endspan==

    Session("posjetiteljBroj") = 0

```

```
        Session("posjetitelj") = false
    End Sub

Sub Application_OnStart
End Sub

</SCRIPT>
```

glavni.css

```
BODY
{
    BACKGROUND: "#FFFFCC";
    COLOR: black;
    FONT-FAMILY: "arial";
    FONT-SIZE: 16px
}

.zaglavljestranice
{
    BACKGROUND: "#990000";
    COLOR: white;
    FONT-FAMILY: "arial";
    FONT-SIZE: 24px
}

.porukaogresci
{
```



```
        COLOR: red;

        FONT-WEIGHT: bold
    }

.zaglavljetablice
{
    BACKGROUND: maroon;

    COLOR: white
}

.podacitablice
{
    BACKGROUND: white;

    COLOR: black
}
```

error.inc

```
<% Response.Write "<p class=porukaogresci>" &
objOdgovor("greskainfo") & "</p>" %>

<% Response.Write "<a href=abort.asp>Abandon</a>" %>

<% stanje = -1 %>
```

dodajKnjigu.asp

```
<%@ TRANSACTION=Required %>

<% Option Explicit %>

<% Response.Expires = 0

    Response.Buffer = True
```

```
%>

<html>
<head>
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">

<title></title>
<link rel="stylesheet" type="text/css" href="glavni.css">
</head>

<body>

<% dim objOdgovor %>
<% dim objKosarice %>

<table width=100% cellspacing=0 cellpadding=0 border=0>
<tr class="zaglavljestranice">
<th align=left>&nbsp;Knji&#382ara Kartelo
</th>
</tr>
</table>

<% set objKosarice =
Server.CreateObject("eKarteloASPKosarice.Kosarice") %>

<% set objOdgovor =
objKosarice.DodajElement(Session("narudzbaID"), Session("email"),
—
```

```

Request.QueryString("katalogid"), 1) %>
<P>Hvala. Va&#353a transakcija je obavljena.</P>
    <% Response.Flush() %>

</BODY>

</HTML>

' Prikazi ovu stranicu ako transakcija uspije.
Sub OnTransactionCommit()

    Response.Write("<p>Va&#353 odabir je dodan u
ko&#353aricu</p>")

    Response.Write("<hr>")

    Response.Write("<input type=BUTTON value='Idem u kupovinu'
onclick='window.history.back(1)''>")

    Response.Write("<input type=BUTTON value='Ko&#353arica'
onclick='\"window.document.location.href='pogleduKosaricu.asp'\">")

    Response.Write("<input type=BUTTON value='Blagajna'
onclick='\"window.document.location.href='DetaljiNarudzbe.asp'\"
>")

    Response.Flush()

end sub

' Prikazi ovu stranicu ako transakcija ne uspije.
Sub OnTransactionAbort()

    Response.Write "<p class=porukaogresci>" &
objOdgovor("errorinfo") & "</p>"

    Response.Write "<a href=abort.asp>Izlazim</a>"

    Response.Flush()

End sub

```

%>

DetaljiNarudzbe.asp

```
<%@ LANGUAGE="VBSCRIPT" %>

<% Option Explicit %>

<% Response.Expires = 0 %>

<% if not Session("posjetitelj") = true then %>
    <% Response.Redirect "provjeraregistracije.asp"%>
<% End If %>

<html>

<head>

<meta HTTP-EQUIV="Pragma" CONTENT="no-cache">

<title></title>

<link rel="stylesheet" type="text/css" href="glavni.css">

</head>

<body>

<% dim stanje %>

<% dim objDB %>

<% dim objNarudzba %>

<% dim objOdgovor %>

<% dim objNarudzbaItems %>

<% dim objElementiKosarice %>
```

```

<% dim objElement %>

<table width=100% cellspacing=0 cellpadding=0 border=0>
<tr class="zaglavljestranice">
<th align=left>&nbsp;Blagajna knjizare
</th>
</tr>
</table>

<% if Session("NarudzbaID") = "" then %>
<% set objDB =
Server.CreateObject("eKarteloSQLKosarice.SQLKosarice") %>
<% set objOdgovor = objDB.OtvoriNarudzbu(Session("email")) %>
<% Session("NarudzbaID") = objOdgovor("NarudzbaID") %>
<% End If %>

<% set objNarudzba =
Server.CreateObject("eKarteloASPObavljanjeNarudzbe.Narudzba") %>
<% set objOdgovor =
objNarudzba.IzradiDetaljeTransakcije(Session("NarudzbaID")) %>
<% if objOdgovor("greska") then %>
<!--#include file=error.inc-->
<% else %>
<% set Session("ElementiNarudzbe") = objOdgovor %>
<% set objNarudzba = nothing %>

<p>Detalji narudzbe za

```

```

<% Response.Write Session("ime") & " " & Session("prezime") %>

</p>

<% set objNarudzbaItems = Session("ElementiNarudzbe") %>

<% set objElementiKosarice =
objNarudzbaItems("kosarica")("kosaricaitems") %>

<% if objElementiKosarice.count = 0 then %>
<p>Vasa kosarica je sad prazna</p>
<% else %>
    <table cellspacing=0 cellpadding=0 border=0>
    <tr class="zaglavljetablice">
    <th>Knjiga</th>
    <th width=15>&nbsp;</th>
    <th>Cijena</th>
    <th width=15>&nbsp;</th>
    <th>Kolicina</th>
    <th width=15>&nbsp;</th>
    <th>Ukupno</th>
    </tr>

<% for each objElement in objElementiKosarice %>
<% if objElement("kolicina") > 0 then %>
    <tr class="tabledata">
        <td>
<% Response.Write objElement("naslov") %>
        </td>

```

```

        <td>&nbsp;</td>
        <td align=right>
<%
    Response.Write objElement("danasnjacijena") %>
    </td>
    <td>&nbsp;</td>
    <td align=right>
<%
    Response.Write objElement("kolicina") %>
    </td>
    <td>&nbsp;</td>
    <td align=right>
<%
    Response.Write objElement("ukupnacijena") %>
    </td>
    </tr>
<%
    end if %>
<%
    next %>
        <tr class="podacitablice">
        <td colspan= 6>
        <i>plus</i> isporuka:
<%
    Response.Write objNarudzbaItems("isporuka")("detalji")
%>
        </td>
        <td align=right>
<%
    Response.Write
objNarudzbaItems("isporuka")("ukupnacijena") %>
        </td>
    </tr>
    <tr class="tabledata">

```

```

                <td colspan=6>
                <i>plus</i> Tax:
<%
    Response.Write objNarudzbaItems("porez")("detalji") %>
                </td>
                <td align=right>
<%
    Response.Write objNarudzbaItems("porez")("ukupnacijena")
%>
                </td>
                </tr>
                <tr class="podacitablice">
                <td colspan=6>
                <b>Narudzba ukupno</b>
                </td>
                <td align=right>
                <hr><b>
<%
    Response.Write
objNarudzbaItems("narudzbaukupno")("ukupnacijena") %>
<%
    Session("ukupnacijena") =
objNarudzbaItems("narudzbaukupno")("ukupnacijena") %>
                <b>
                </td>
                </tr>
                </table>
<%
end if %>
                <hr>
<%
if objElementiKosarice.count > 0 then %>
                <input type=button value="Kupujem"

```



```
        onclick=window.document.location.href='IzbaciRacun.asp'>
        <input type=button value="Amend order"
        onclick=window.document.location.href='pogleduKosaricu.asp'>

    <% end if %>
<% end if %>
</body>
</html>
```

azurirajKosaricu.asp

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<% Response.Expires = 0 %>

<html>
<head>
<title></title>
<link rel="stylesheet" type="text/css" href="glavni.css">
</head>
<meta HTTP-EQUIV="Pragma" CONTENT="no-cache">

<body>
<% dim stanje %>
<% dim objKosarica %>
```

```

<% dim objKnjiga %>

<% dim objOdgovor %>

<% dim bGr %>

<table width=100% cellpadding=0 cellspacing=0 border=0>

<tr class="zaglavljestranice">

<th align=left>&nbsp;Sadr&#382aj ko&#353arice

</th>

</tr>

</table>

<% set objKosarica =
Server.CreateObject("eKarteloASPKosarice.Kosarice") %>

<% if Request.Form("UPDATE") = "Azuriraj" then %>

<% bGr = false %>

<% for each objKnjiga in Request.Form %>

<% if not objKnjiga = "update" then %>

<% if isnumeric(Request.Form(objKnjiga)) then %>

<% if Request.Form(objKnjiga) >= 0 then %>

<% set objOdgovor =
objKosarica.AzurirajBrojElemenata(objKnjiga,
Request.Form(objKnjiga)) %>

<% if objOdgovor("greska") then %>

<!--#include file=error.inc-->

<% end if %>

<% else %>

<% bGr = true %>

```

```

<%           end if %>
<%           else %>
<%           bGr = true %>
<%           end if %>
<%           end if %>
<%           next %>
<%           if bGr then %>
               <p class=porukaogresci>Otkrivena je nepravilnost u
podatku 'Kolicina'</p>
<%           end if %>

<%           else %>
<%           set objOdgovor =
objKosarica.IsprazniKosaricu(Session("narudzbaID")) %>
<%           if objOdgovor("greska") then %>
<!--#include file=error.inc-->
<%           end if %>
<%           end if %>
               <p>A&#382urirani detalji</p>
               <input type="button" value="Pogled u ko&#353aricu"
onclick=window.document.location.href='pogleduKosaricu.asp'>

</body>
</html>

```

abort.asp

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<% Response.Expires = 0 %>

<html>
<head>
<meta HTTP-EQUIV="Pragma" CONTENT="no-cache">

<title>Home</title>
<link rel="stylesheet" type="text/css" href="main.css">
</head>
<body bgcolor=Cyan>

Posjet prekinut.

<% Session.Abandon %>

</body>
</html>
```

Lekcija 10

Profesionalni primjer E- knjižara Kartelo COM Visual Basic 6 komponente

Ovdje je cjelovit Visual Basic 6 programski kôd DLL komponenti. Svaka komponenta ima jedan modul koji se naziva modZajednicki. Taj je modul jednak za sve DLL komponente. DLL komponente se razlikuju samo po class modulima koje imaju. Zato ću najprije dati ispis kôda zajedničkog modula, a onda redom class module svih komponenti.

modZajednicki (modZajednicki.bas)

```
Public Const vSpojNaBazu As Variant = "DSN=EKARTELO;UID=sa;PWD=";  
  
Public Sub Trace(vPoruka As Variant)  
  
    Open "c:\ekartelo.txt" For Append Shared As #1
```

```

Print #1, Time & " " & vPoruka

Close #1

End Sub

Public Function JaviGresku(ByVal vIme As Variant) As Variant

    Dim vGreskaInfo As Variant

    Dim vOdgovor As Collection

    vGreskaInfo = "Ekartelo greska sustava: " & Hex(Err.Number) &
" " & Err.Source & "->" & vIme & " " & Err.Description

    Err.Clear

    Set vOdgovor = New Collection

    vOdgovor.Add True, Key:="greska"

    vOdgovor.Add vGreskaInfo, Key:="greskainfo"

    Trace (vGreskaInfo)

    Set JaviGresku = vOdgovor

End Function

Public Function PodesiDanasnjuCijenu(ByVal vCijena As Variant,
ByVal vPromidzbaKod As Variant, ByVal vPromidzbaKraj As Variant)
As Variant

    Dim vDanasnjaCijena As Variant

    If DateDiff("d", Now, vPromidzbaKraj) < 0 Then

        vDanasnjaCijena = vCijena

    Else

        Select Case vPromidzbaKod

            Case "A"

                vDanasnjaCijena = vCijena * 0.95

            Case "B"

                vDanasnjaCijena = vCijena * 0.9

```

```

Case "C"
    vDanasnjaCijena = vCijena * 0.85
Case "D"
    vDanasnjaCijena = vCijena * 0.8
Case "E"
    vDanasnjaCijena = vCijena * 0.75
Case Else
    vDanasnjaCijena = vCijena
End Select

End If

PodesiDanasnjuCijenu = vDanasnjaCijena

End Function

Public Function IzdvojiNazivZanimaSeZa(ByVal vKod As Variant) As Variant

    Dim vNaziv As Variant

    Select Case vKod

    Case "M"

        vNaziv = "MicrosoftOffice"

    Case "P"

        vNaziv = "Programiranje"

    Case "I"

        vNaziv = "Internet"

    Case "Z"

        vNaziv = "Zanimanje"

    Case "O"

        vNaziv = "Ostalo"

```

End Select

IzdvojiNazivZanimaSeZa = vNaziv

End Function

Public Function IzgradiZanimaSeZa(ByVal vUkljuceno As Variant) As Variant

Dim vZanimaSeZaKodovi As Variant

Dim vZanimaSeZa As Collection

Dim vZanima As Collection

Dim i

vZanimaSeZaKodovi = Array("M", "P", "I", "Z", "O")

Set vZanimaSeZa = New Collection

For i = 0 To 4

Set vZanima = New Collection

Trace ("Zanima se za: " & vZanimaSeZaKodovi(i))

vZanima.Add IzdvojiNazivZanimaSeZa(vZanimaSeZaKodovi(i)),
Key:="naziv"

vZanima.Add vZanimaSeZaKodovi(i), Key:="kod"


```

        vZanima.Add InStr(vUkljuceno, vZanimaSeZaKodovi(i)) <> 0,
Key:="ukljuceno"

        vZanimaSeZa.Add vZanima
        Set vZanima = Nothing
    Next i

Trace ("Zanima se za: " & TypeName(vZanimaSeZa))

Set IzgradiZanimaSeZa = vZanimaSeZa
Set vZanimaSeZa = Nothing

End Function

Public Function IzdvojiZanimaSeZaKodove(ByVal vZanimaSeZa As
Variant) As Variant

    Dim vZanima As Variant
    Dim vNaziv As Variant
    vNaziv = ""

    For Each vZanima In vZanimaSeZa
        If vZanima("ukljuceno") Then
            vNaziv = vNaziv & vZanima("kod") & ", "
        End If
    Next

    vNaziv = Left(vNaziv, Len(vNaziv) - 1)

    IzdvojiZanimaSeZaKodove = vNaziv

```

End Function

eKarteloASPPomocna.dll

Class modul (objekt): NovaKolekcija (NovaKolekcija.cls)

```
,  
  
Option Base 0  
Option Explicit  
Implements ObjectControl  
  
Private oObjectContext As ObjectControl  
  
Private Function ObjectControl_CanBePooled() As Boolean  
    ObjectControl_CanBePooled = True  
End Function  
  
Private Sub ObjectControl_Deactivate()  
    Set oObjectContext = Nothing  
End Sub  
  
Private Sub ObjectControl_Activate()  
    On Error GoTo OCActivateErr  
  
    Set oObjectContext = GetObjectContext()
```

```
Exit Sub
```

```
OCActivateErr:
```

```
Call JaviGresku("OCActivate")
```

```
End Sub
```

```
Public Function NovaKolekcija() As Variant
```

```
On Error GoTo NovaKolekcijaErr
```

```
Dim vOdgovor As New Collection
```

```
oObjectContext.SetComplete
```

```
Set NovaKolekcija = vOdgovor
```

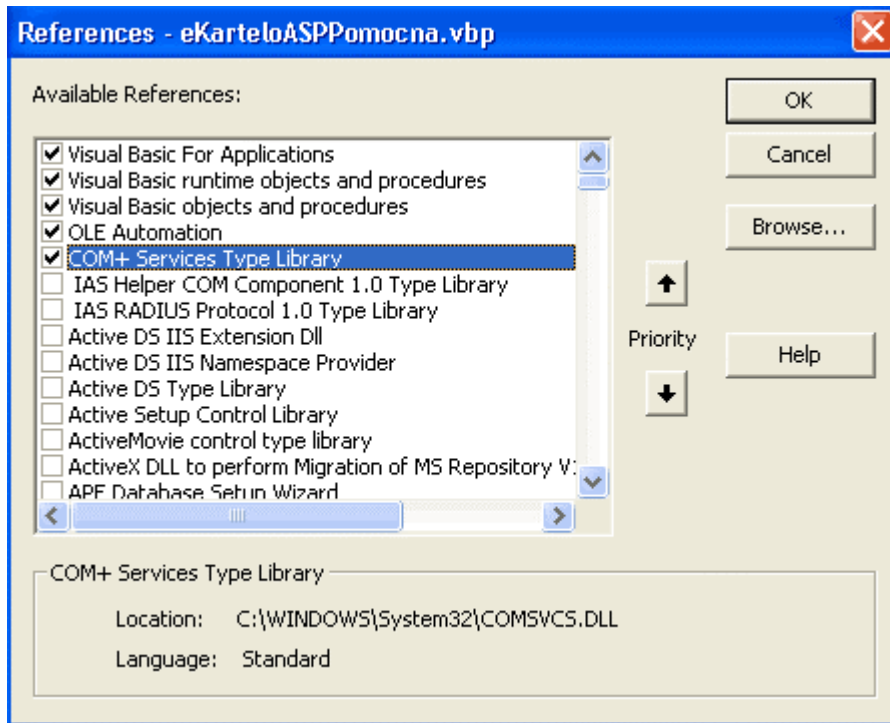
```
Exit Function
```

```
NovaKolekcijaErr:
```

```
oObjectContext.SetAbort
```

```
Set NovaKolekcija = JaviGresku("NovaKolekcija")
```

```
End Function
```



eKarteloSQLKupci.dll

Class modul (objekt): SQLKupci (SQLKupci.cls)

,

Option Base 0

Option Explicit

Implements ObjectControl

Private oDb As ADODB.Connection

Private oRs As ADODB.Recordset

Private oObjectContext As ObjectContext

```

Private Function ObjectControl_CanBePooled() As Boolean
    ObjectControl_CanBePooled = True
End Function

Private Sub ObjectControl_Deactivate()
    Set oObjectContext = Nothing
    Set oDb = Nothing
    Set oRs = Nothing
End Sub

Private Sub ObjectControl_Activate()
    On Error GoTo OCActivateErr

    Set oObjectContext = GetObjectContext()

    Exit Sub

OCActivateErr:
    Call JaviGresku("OCActivate")

End Sub

Public Function AzuriranjeKupca(ByVal vPodaciKupca As Variant) As Variant
    On Error GoTo AzuriranjeKupcaErr

```

```

Dim vOdgovor As New Collection

Dim vSql As Variant

Let vSql = "SELECT * FROM KUPCI " & _
           " WHERE email = '" & LCase(vPodaciKupca("email")) &
"';"

Set oRs = oObjectContext.CreateInstance("ADODB.Recordset")

oRs.CursorType = adOpenDynamic
oRs.LockType = adLockPessimistic
oRs.Open vSql, vSpojNaBazu

oRs.Fields("Email") = Trim(LCase(vPodaciKupca("email")))
oRs.Fields("Zaporka") = Trim(vPodaciKupca("zaporka"))
oRs.Fields("Ime") = Left(Trim(vPodaciKupca("ime")), 25)
oRs.Fields("Prezime") = Left(Trim(vPodaciKupca("prezime")),
25)
oRs.Fields("Adresa") = Left(Trim(vPodaciKupca("adresa")), 40)
oRs.Fields("Grad") = Left(Trim(vPodaciKupca("grad")), 20)
oRs.Fields("PostanskiBroj") =
Left(Trim(vPodaciKupca("postanskibroj")), 10)
oRs.Fields("Drzava") = Left(Trim(vPodaciKupca("drzava")), 20)
oRs.Fields("Telefon") = Left(Trim(vPodaciKupca("telefon")),
20)
oRs.Fields("KoristiCookie") =
Left(Trim(vPodaciKupca("KoristiCookie")), 1)
oRs.Fields("ZanimaSeZa") =
IzvojiZanimaSeZaKodove(vPodaciKupca("zanimaseza"))

```

```
oRs.Update
```

```
oRs.Close
```

```
vOdgovor.Add False, Key:="greska"
```

```
oObjectContext.SetComplete
```

```
Set AzuriranjeKupca = vOdgovor
```

```
Exit Function
```

```
AzuriranjeKupcaErr:
```

```
oObjectContext.SetAbort
```

```
Set AzuriranjeKupca = JaviGresku("AzuriranjeKupca")
```

```
End Function
```

```
Public Function UnesiKupca(ByVal vPodaciKupca As Variant) As  
Variant
```

```
On Error GoTo UnesiKupcaErr
```

```
Dim vOdgovor As New Collection
```

```
Dim vPodaciKupca2 As New Collection
```

```
Set oRs = oObjectContext.CreateInstance("ADODB.Recordset")
```

```

oRs.Open "kupci", vSpojNaBazu, adOpenKeyset,
adLockPessimistic, adCmdTable

oRs.AddNew

oRs.Fields("Email") = LCase(vPodaciKupca("email"))
oRs.Fields("Zaporka") = LTrim(vPodaciKupca("zaporka"))

oRs.Fields("Ime") = ""
oRs.Fields("Prezime") = ""
oRs.Fields("Adresa") = ""
oRs.Fields("Grad") = ""
oRs.Fields("PostanskiBroj") = ""
oRs.Fields("Drzava") = ""
oRs.Fields("Telefon") = ""
oRs.Fields("KoristiCookie") = "N"
oRs.Fields("ZanimaSeZa") = ""

oRs.Update

vPodaciKupca2.Add Trim(oRs.Fields("Email")), "email"
vPodaciKupca2.Add Trim(oRs.Fields("Zaporka")), "zaporka"
vPodaciKupca2.Add "", "ime"
vPodaciKupca2.Add "", "prezime"
vPodaciKupca2.Add "", "adresa"
vPodaciKupca2.Add "", "grad"
vPodaciKupca2.Add "", "postanskibroj"
vPodaciKupca2.Add "", "drzava"
vPodaciKupca2.Add "", "telefon"
vPodaciKupca2.Add "N", "koristicookie"

```



```

vPodaciKupca2.Add IzgradiZanimaSeZa(""), "zanimaseza"

vOdgovor.Add False, Key:="greska"
vOdgovor.Add True, Key:="postoji"
vOdgovor.Add vPodaciKupca2, "podacikupca"

oRs.Close

oObjectContext.SetComplete
Set UnesiKupca = vOdgovor

Exit Function

UnesiKupcaErr:
oObjectContext.SetAbort
Set UnesiKupca = JaviGresku("UnesiKupca")

End Function

Public Function OdaberiKupca(ByVal vPodaciKupca As Variant) As Variant
On Error GoTo OdaberiKupcaErr

Dim vOdgovor As New Collection
Dim vPodaciKupca2 As New Collection

```

```

Dim vZanimaSeZa As Variant

Dim vSql As Variant

Let vSql = "SELECT * FROM KUPCI " & _
           " WHERE email = '" & LCase(vPodaciKupca("email")) &
";"

Set oDb = oObjectContext.CreateInstance("ADODB.Connection")
oDb.Open vSpojNaBazu
Set oRs = oDb.Execute(vSql)

If Not oRs.EOF Then

    vPodaciKupca2.Add Trim(oRs.Fields("Zaporka")), "zaporka"
    vPodaciKupca2.Add CStr(oRs.Fields("email")), "email"
    vPodaciKupca2.Add Trim(oRs.Fields("Ime")), "ime"
    vPodaciKupca2.Add Trim(oRs.Fields("Prezime")), "prezime"
    vPodaciKupca2.Add Trim(oRs.Fields("Adresa")), "adresa"
    vPodaciKupca2.Add Trim(oRs.Fields("Grad")), "grad"
    vPodaciKupca2.Add Trim(oRs.Fields("PostanskiBroj")),
"postanskibroj"
    vPodaciKupca2.Add Trim(oRs.Fields("Drzava")), "drzava"
    vPodaciKupca2.Add Trim(oRs.Fields("Telefon")), "telefon"
    vPodaciKupca2.Add Trim(oRs.Fields("KoristiCookie")),
"koristicookie"

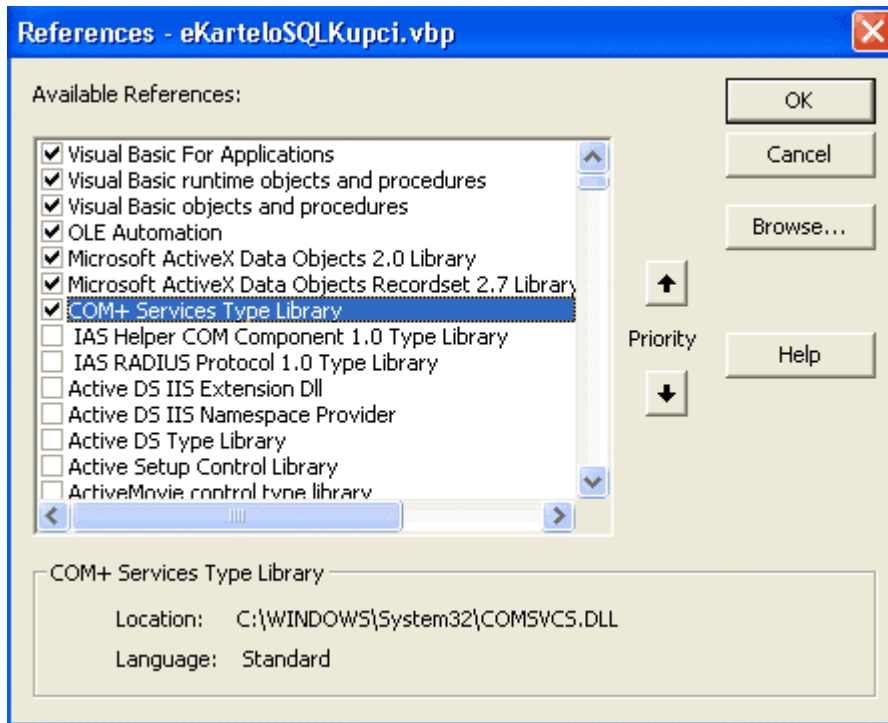
    Trace ("Izgradnja zanima se za:")

    Set vZanimaSeZa =
IzgradiZanimaSeZa(Trim(oRs.Fields("ZanimaSeZa")))

    vPodaciKupca2.Add vZanimaSeZa, "zanimaseza"

```

```
        vOdgovor.Add True, Key:="postoji"  
        vOdgovor.Add vPodaciKupca2, "podacikupca"  
Else  
        vOdgovor.Add False, Key:="postoji"  
End If  
  
oDb.Close  
  
vOdgovor.Add False, Key:="greska"  
  
oObjectContext.SetComplete  
Set OdaberiKupca = vOdgovor  
  
Exit Function  
  
OdaberiKupcaErr:  
    oObjectContext.SetAbort  
    Set OdaberiKupca = JaviGresku("OdaberiKupca")  
  
End Function
```



eKarteloASPkupci.dll

Class modul (objekt): Kupci (Kupci.css)

```

'
Option Base 0
Option Explicit
Implements ObjectControl

Private oSQLKupci As eKarteloSQLKupci.SQLKupci
Private oObjectContext As ObjectContext

Private Function ObjectControl_CanBePooled() As Boolean
    ObjectControl_CanBePooled = True
End Function

```

```
Private Sub ObjectControl_Deactivate()
```

```
    Set oObjectContext = Nothing
```

```
    Set oSQLKupci = Nothing
```

```
End Sub
```

```
Private Sub ObjectControl_Activate()
```

```
    On Error GoTo OCActivateErr
```

```
    Set oObjectContext = GetObjectContext()
```

```
    Set oSQLKupci =  
oObjectContext.CreateInstance("eKarteloSQLKupci.SQLKupci")
```

```
    Exit Sub
```

```
OCActivateErr:
```

```
    Call JaviGresku("OCActivate")
```

```
End Sub
```

```
Public Function AzurirajKupca(ByVal vPodaciKupca As Variant) As  
Variant
```

```
On Error GoTo AzurirajKupcaErr
```

```
Dim vOdgovor As Variant
```

```
Set vOdgovor = oSQLKupci.AzuriranjeKupca(vPodaciKupca)
```

```
oObjectContext.SetComplete  
Set AzurirajKupca = vOdgovor
```

```
Exit Function
```

```
AzurirajKupcaErr:
```

```
oObjectContext.SetAbort  
Set AzurirajKupca = JaviGresku("AzurirajKupca")
```

```
End Function
```

```
Public Function RegistrirajKupca(ByVal vPodaciKupca As Variant) As  
Variant
```

```
On Error GoTo RegistrirajKupcaErr
```

```
Dim vOdgovor As Variant
```

```
Set vOdgovor = ProvjeriKupca(vPodaciKupca)
```

```
If Not vOdgovor("greska") And _
```

```
Not vOdgovor("postoji") Then
```

```
Set vOdgovor = oSQLKupci.UnesiKupca(vPodaciKupca)
```

```
vOdgovor.Add True, Key:="zaporkatocna"
```

```
End If
```

```
oObjectContext.SetComplete
```

```
Set RegistrirajKupca = vOdgovor
```

```
Exit Function
```

```
RegistrirajKupcaErr:
```

```
oObjectContext.SetAbort
```

```
Set RegistrirajKupca = JaviGresku("RegistrirajKupca")
```

```
End Function
```

```
Public Function ProvjeriKupca(ByVal vPodaciKupca As Variant) As Variant
```

```
On Error GoTo ProvjeriKupcaErr
```

```
Dim vZaporka2 As Variant
```

```
Dim vOdgovor As Variant
```

```
Dim vPodaciKupca2 As Variant
```

```
Dim vOdgovor2 As Variant
```

```
vZaporka2 = vPodaciKupca("zaporka")
```

```
Set oSQLKupci =  
oObjectContext.CreateInstance("eKarteloSQLKupci.SQLKupci")
```

```
Set vOdgovor = oSQLKupci.OdaberiKupca(vPodaciKupca)
```

```
If Not vOdgovor("greska") Then
```

```

    If vOdgovor("postoji") Then
        Set vPodaciKupca2 = vOdgovor("PodaciKupca")
        Set vOdgovor2 = New Collection
        vOdgovor2.Add False, Key:="greska"
        vOdgovor2.Add True, Key:="postoji"
        vOdgovor2.Add vPodaciKupca2, Key:="podacikupca"

        If (Trim(vPodaciKupca2("zaporka")) Like
Trim(vZaporka2)) Then
            vOdgovor2.Add True, Key:="zaporkatocna"
        Else
            vOdgovor2.Add False, Key:="zaporkatocna"
        End If
        Set vOdgovor = vOdgovor2
    End If

End If

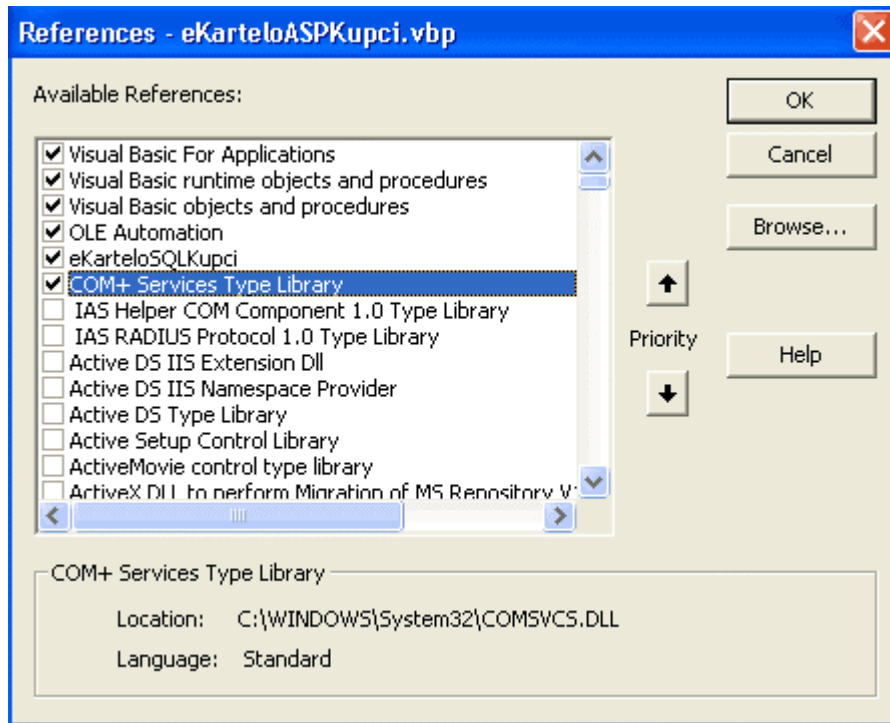
oObjectContext.SetComplete
Set ProvjeriKupca = vOdgovor

Exit Function

ProvjeriKupcaErr:
oObjectContext.SetAbort
Set ProvjeriKupca = JaviGresku("ProvjeriKupca")

End Function

```

eKarteloSQLKatalog.dll

Class modul (objekt): SQLKatalog (SQLKatalog.cls)

```
'  
  
Option Base 0  
  
Option Explicit  
  
Implements ObjectControl  
  
  
Private oDb As ADODB.Connection  
  
Private oRs As ADODB.Recordset  
  
  
Private oObjectContext As ObjectContext
```

```
Private Function ObjectControl_CanBePooled() As Boolean
    ObjectControl_CanBePooled = True
End Function
```

```
Private Sub ObjectControl_Deactivate()
    Set oObjectContext = Nothing
    Set oDb = Nothing
    Set oRs = Nothing
End Sub
```

```
Private Sub ObjectControl_Activate()
    On Error GoTo OCActivateErr

    Set oObjectContext = GetObjectContext()

    Exit Sub
```

```
OCActivateErr:
    Call JaviGresku("OCActivate")

End Sub
```

```
Public Function IzbaciElementeKatalogaPoTipu(ByVal vTipKodovi As
Variant) As Variant
```

```
On Error GoTo IzbaciElementeKatalogaPoTipuErr
```

```
Dim vOdgovor As New Collection
```

```
Dim vElementi As New Collection
```

```
Dim vSql As Variant
```

```
Dim vKod As Variant
```

```
Dim vNaslov As Variant
```

```
Dim vCijena As Variant
```

```
Dim vKatalogId As Variant
```

```
Dim vOpis As Variant
```

```
Dim vSlikaURL As Variant
```

```
Dim vPromidzbaKod As Variant
```

```
Dim vPromidzbaKraj As Variant
```

```
Dim vObjavljeno As Variant
```

```
Dim vTipKodoviLen As Variant
```

```
Dim iOdKojegZnakaPoceti As Integer
```

```
Dim iPozicijaZareza As Integer
```

```
Dim vKategorijaID As Variant
```

```
Dim bILI As Boolean
```

```
Let vSql = "SELECT DISTINCTROW Katalog.*,  
Promidzbe.PromidzbaKod, Promidzbe.PromidzbaKraj FROM (Katalog  
INNER JOIN Kategorije ON Katalog.KatalogId = Kategorije.KatalogID)  
LEFT JOIN Promidzbe ON Katalog.KatalogId = Promidzbe.KatalogID  
WHERE"
```

```
vTipKodoviLen = Len(vTipKodovi)
```

```

If vTipKodoviLen > 0 Then
    iOdKojegZnakaPoceti = 1
    Do
        iPozicijaZareza = InStr(iOdKojegZnakaPoceti,
vTipKodovi, ",")
        If iPozicijaZareza = 0 Then
            vKategorijaID = Mid(vTipKodovi,
iOdKojegZnakaPoceti, vTipKodoviLen)
            bILI = False
        Else
            vKategorijaID = Mid(vTipKodovi,
iOdKojegZnakaPoceti, iPozicijaZareza - iOdKojegZnakaPoceti)
            bILI = True
        End If
        iOdKojegZnakaPoceti = iPozicijaZareza + 1
        vSql = vSql & "(Kategorije.KategorijaID = '" &
vKategorijaID & "')"
        If bILI Then vSql = vSql & " OR "
    Loop Until iPozicijaZareza = 0

vSql = vSql & " ORDER BY Objavljeno DESC"

Trace (vSql)

Set oDb =
oObjectContext.CreateInstance("ADODB.Connection")
oDb.Open vSpojNaBazu
Set oRs = oDb.Execute(vSql)

```

```

While Not oRs.EOF

    Dim KatalogElementi As New Collection

    vKod = oRs.Fields("kod")

    vNaslov = oRs.Fields("naslov")

    vCijena = Format(oRs.Fields("cijena"), "#,##0.00")

    vKatalogId = oRs.Fields("katalogid")

    vOpis = oRs.Fields("opis")

    vSlikaURL = oRs.Fields("slikaurL")

    vPromidzbaKod = oRs.Fields("promidzbakod")

    vPromidzbaKraj = oRs.Fields("promidzbakraj")

    vObjavljeno = oRs.Fields("objavljeno")

    KatalogElementi.Add vKod, Key:="kod"

    KatalogElementi.Add vNaslov, Key:="naslov"

    KatalogElementi.Add vCijena, Key:="cijena"

    KatalogElementi.Add vKatalogId, Key:="katalogid"

    KatalogElementi.Add vOpis, Key:="opis"

    KatalogElementi.Add vSlikaURL, Key:="slikaurL"

    KatalogElementi.Add vPromidzbaKod, Key:="promidzbakod"

    KatalogElementi.Add vPromidzbaKraj,
Key:="promidzbakraj"

    KatalogElementi.Add vObjavljeno, Key:="objavljeno"

    vElementi.Add KatalogElementi

```

```
        Set KatalogElementi = Nothing
        oRs.MoveNext
    Wend

    oRs.Close

End If

vOdgovor.Add False, Key:="greska"
vOdgovor.Add vElementi, Key:="elementi"

oObjectContext.SetComplete

Set IzbaciElementeKatalogaPoTipu = vOdgovor

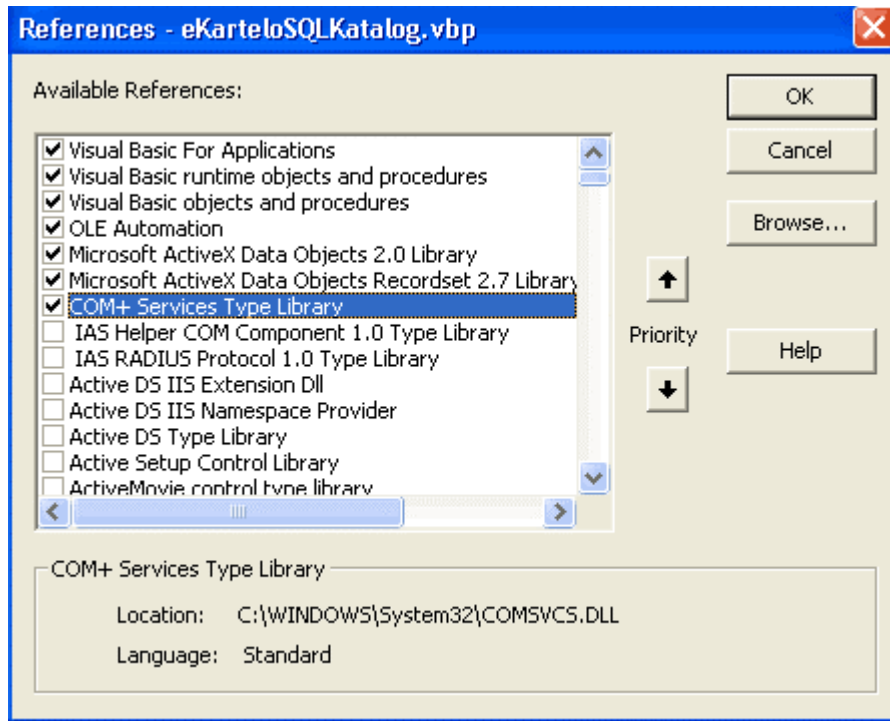
Exit Function

IzbaciElementeKatalogaPoTipuErr:

oObjectContext.SetAbort

Set IzbaciElementeKatalogaPoTipu =
JaviGresku("IzbaciElementeKatalogaPoTipu")

End Function
```



eKarteloASPKatalog.dll

Class modul Katalog (Katalog.cls)

```
'  
' katalog.cls  
'  
  
Option Base 0  
  
Option Explicit  
  
Implements ObjectControl  
  
  
  
Private oSQLKatalog As eKarteloSQLKatalog.SQLKatalog
```

```

Private oObjectContext As ObjectContext

Private Function ObjectControl_CanBePooled() As Boolean
    ObjectControl_CanBePooled = True
End Function

Private Sub ObjectControl_Deactivate()
    Set oObjectContext = Nothing
    Set oSQLKatalog = Nothing
End Sub

Private Sub ObjectControl_Activate()
    On Error GoTo OCActivateErr

    Set oObjectContext = GetObjectContext()

    Set oSQLKatalog =
oObjectContext.CreateInstance("eKarteloSQLKatalog.SQLKatalog")

    Exit Sub

OCActivateErr:
    Call JaviGresku("OCActivate")

End Sub

```



```
Public Function VratiKatalogListuPoZanimaseza(ByVal vZanimaSeZa As Variant) As Variant
```

```
    On Error GoTo VratiKatalogListuPoZanimasezaErr
```

```
    Dim vOdgovor As Variant
```

```
    Dim vKatalog As Variant
```

```
    Dim vKatalogItem As Variant
```

```
    Dim vObjavljeno As Variant
```

```
    Dim vDanasnjaCijena As Variant
```

```
    Dim vCijena As Variant
```

```
    Dim vPromidzbaKod As Variant
```

```
    Dim vPromidzbaKraj As Variant
```

```
    Dim vPromidzbaDa As Variant
```

```
    Dim vNovaDa As Variant
```

```
    Dim vZanimaSeZaKodovi As Variant
```

```
    Trace "Ispis kodova Zanimaseza"
```

```
    vZanimaSeZaKodovi = IzdvojiZanimaSeZaKodove(vZanimaSeZa)
```

```
    Trace "Calling SQLKatalog.IzbaciElementeKatalogaPoTipu"
```

```
    Set vOdgovor =  
oSQLKatalog.IzbaciElementeKatalogaPoTipu(vZanimaSeZaKodovi)
```

```
    If Not vOdgovor("greska") Then
```

```
        Set vKatalog = vOdgovor("elementi")
```

```
        For Each vKatalogItem In vKatalog
```

```

vObjavljeno = CDate(vKatalogItem("objavljeno"))
vCijena = vKatalogItem("cijena")
vPromidzbaKod = vKatalogItem("promidzbakod")
vPromidzbaKraj = vKatalogItem("promidzbakraj")

0 Then
    If DateDiff("d", Now, DateAdd("m", 3, vObjavljeno)) >
        vNovaDa = True
    Else
        vNovaDa = False
    End If

    vKatalogItem.Add vNovaDa, Key:="novada"

    vDanasnjaCijena = PodesiDanasnjuCijenu(vCijena,
vPromidzbaKod, vPromidzbaKraj)

    If vDanasnjaCijena = vCijena Then
        vPromidzbaDa = False
    Else
        vPromidzbaDa = True
    End If

    vDanasnjaCijena = Format(vDanasnjaCijena, "#,##0.00")

    vKatalogItem.Add vDanasnjaCijena, Key:="danasnjacijena"
    vKatalogItem.Add vPromidzbaDa, Key:="promidzbada"

```

Next

End If

oObjectContext.SetComplete

Set VratiKatalogListuPoZanimaseza = vOdgovor

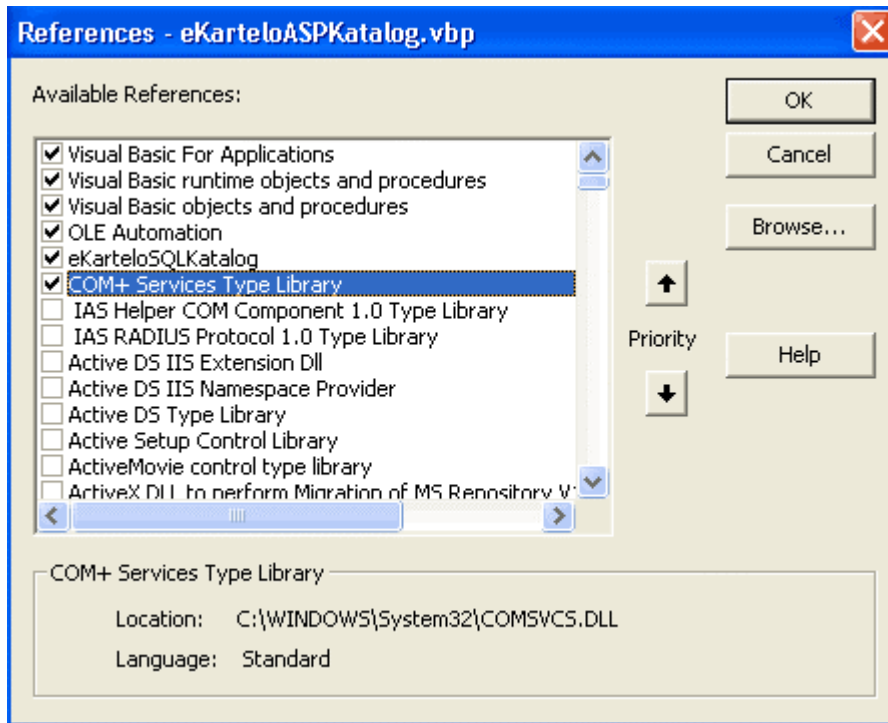
Exit Function

VratiKatalogListuPoZanimasezaErr:

oObjectContext.SetAbort

Set VratiKatalogListuPoZanimaseza =
JaviGresku("VratiKatalogListuPoZanimaseza")

End Function



eKarteloSQLKosarice.dll

Class modul (objekt): SQLKosarice (SQLKosarice.cls)

```
'  
'
```

```
Option Base 0
```

```
Option Explicit
```

```
Implements ObjectControl
```

```
Private oDb As ADODB.Connection
```

```
Private oRs As ADODB.Recordset
```

```
Private oObjectContext As ObjectContext

Private Function ObjectControl_CanBePooled() As Boolean
    ObjectControl_CanBePooled = True
End Function

Private Sub ObjectControl_Deactivate()
    Set oObjectContext = Nothing
    Set oDb = Nothing
    Set oRs = Nothing

End Sub

Private Sub ObjectControl_Activate()
    On Error GoTo OCActivateErr

    Set oObjectContext = GetObjectContext()

    Exit Sub

OCActivateErr:
    Call JaviGresku("OCActivate")

End Sub
```

```

Public Function VратиElementeKosarice(ByVal vNarudzbaID As
Variant) As Variant

    On Error GoTo VратиElementeKosariceErr

    Dim vOdgovor As New Collection

    Dim vElementi As New Collection

    Dim vSql As Variant

    Dim vKod As Variant

    Dim vNaslov As Variant

    Dim vKosaricaId As Variant

    Dim vCijena As Variant

    Dim vPromidzbaKod As Variant

    Dim vPromidzbaKraj As Variant

    Dim vKolicina As Variant

    Let vSql = "SELECT KosariceSadrzaj.*, Katalog.*,
Promidzbe.PromidzbaKod, Promidzbe.PromidzbaKraj" & _

        " FROM (KosariceSadrzaj INNER JOIN Katalog ON
KosariceSadrzaj.KatalogId = Katalog.KatalogId) LEFT JOIN Promidzbe
ON Katalog.KatalogId = Promidzbe.KatalogID" & _

        " WHERE KosariceSadrzaj.NarudzbaID = " & vNarudzbaID & ";"

    Trace vSql

    Set oDb = oObjectContext.CreateInstance("ADODB.Connection")

    oDb.Open vSpojNaBazu

    Set oRs = oDb.Execute(vSql)

    While Not oRs.EOF

```

```
Dim KosaricaItem As New Collection
```

```
vKosaricaId = oRs.Fields("kosaricaid")
```

```
vKod = oRs.Fields("kod")
```

```
vNaslov = oRs.Fields("naslov")
```

```
vCijena = oRs.Fields("cijena")
```

```
vKolicina = oRs.Fields("kolicina")
```

```
vPromidzbaKod = oRs.Fields("promidzbakod")
```

```
vPromidzbaKraj = oRs.Fields("promidzbakraj")
```

```
KosaricaItem.Add vKosaricaId, Key:="kosaricaid"
```

```
KosaricaItem.Add vKod, Key:="kod"
```

```
KosaricaItem.Add vNaslov, Key:="naslov"
```

```
KosaricaItem.Add vCijena, Key:="cijena"
```

```
KosaricaItem.Add vKolicina, Key:="kolicina"
```

```
KosaricaItem.Add vPromidzbaKod, Key:="promidzbakod"
```

```
KosaricaItem.Add vPromidzbaKraj, Key:="promidzbakraj"
```

```
vElementi.Add KosaricaItem
```

```
Set KosaricaItem = Nothing
```

```
oRs.MoveNext
```

```
Wend
```

```
oRs.Close
```

```
vOdgovor.Add False, Key:="greska"  
vOdgovor.Add vElementi, Key:="elementi"
```

```
oObjectContext.SetComplete  
Set VratElementeKosarice = vOdgovor
```

```
Exit Function
```

```
VratElementeKosariceErr:
```

```
oObjectContext.SetAbort  
Set VratElementeKosarice =  
JaviGresku("VratElementeKosarice")  
End Function
```

```
Public Function IsprazniKosaricu(ByVal vNarudzbaID As Variant) As  
Variant
```

```
On Error GoTo IsprazniKosaricuErr
```

```
Dim vOdgovor As New Collection  
Dim vSql As Variant
```

```
Let vSql = "DELETE FROM KosariceSadrzaj" & _  
" WHERE NarudzbaID = " & vNarudzbaID & ";"
```

```
Set oDb = oObjectContext.CreateInstance("ADODB.Connection")  
oDb.Open vSpojNaBazu  
Call oDb.Execute(vSql)
```



```
vOdgovor.Add False, Key:="greska"
```

```
oObjectContext.SetComplete
```

```
Set IsprazniKosaricu = vOdgovor
```

```
Exit Function
```

```
IsprazniKosaricuErr:
```

```
oObjectContext.SetAbort
```

```
Set IsprazniKosaricu = JaviGresku("IsprazniKosaricu")
```

```
End Function
```

```
Public Function DodajElement(ByVal vNarudzbaID As Variant, ByVal  
vKatalogId As Variant, ByVal vKolicina As Variant) As Variant
```

```
On Error GoTo DodajElementErr
```

```
Dim vOdgovor As New Collection
```

```
Dim vSql As Variant
```

```
Let vSql = "SELECT * FROM KosariceSadrzaj " & _
```

```
    "WHERE (NarudzbaID = " & vNarudzbaID & ")" & _
```

```
    " AND (KatalogId = " & vKatalogId & ") ;"
```

```
Set oRs = oObjectContext.CreateInstance("ADODB.Recordset")
```

```
oRs.CursorType = adOpenDynamic
```

```
oRs.LockType = adLockPessimistic
```

```

oRs.Open vSql, vSpojNaBazu

If oRs.EOF Then
    oRs.Close

    oRs.Open "KosariceSadrzaj", vSpojNaBazu, adOpenKeyset,
adLockPessimistic, adCmdTable

    oRs.AddNew

    oRs.Fields("NarudzbaID") = vNarudzbaID

    oRs.Fields("KatalogId") = CLng(vKatalogId)

    oRs.Fields("Kolicina") = CLng(vKolicina)

Else
    oRs.Fields("Kolicina") = oRs.Fields("Kolicina") +
CLng(vKolicina)

End If

oRs.Update

oRs.Close

vOdgovor.Add False, Key:="greska"

ObjectContext.SetComplete

Set DodajElement = vOdgovor

Exit Function

DodajElementErr:

ObjectContext.SetAbort

```

```

Set DodajElement = JaviGresku("DodajElement")

End Function

Public Function OtvoriNarudzbu(ByVal vEmail As Variant) As Variant

    On Error GoTo OtvoriNarudzbuErr

    Dim vOdgovor As New Collection

    Dim vSql As Variant

    Dim vNarudzbaID As Variant

    Let vSql = "SELECT * FROM NarudzbeKosarica WHERE (Email = '" &
Trim(vEmail) & "')" & _
        "AND (Fulfilled = 0);"

    Set oRs = oObjectContext.CreateInstance("ADODB.Recordset")

    Trace vSql

    oRs.CursorType = adOpenDynamic

    oRs.LockType = adLockPessimistic

    oRs.Open vSql, vSpojNaBazu

    If oRs.EOF Then

        Dim vNow As Variant

        vNow = Now

        oRs.AddNew

        oRs.Fields("Email") = vEmail

        oRs.Fields("KosaricaNastala") = vNow

        oRs.Update

```

```

oRs.Close

vSql = "SELECT * FROM NARUDZBEKOSARICA WHERE (Email = '" &
Trim(vEmail) & "')" & _
" AND (Fulfilled = 0) and KosaricaNastala = #" & vNow &
"#;"

Trace vSql

oRs.Open vSql, vSpojNaBazu

End If

vNarudzbaID = oRs("NarudzbaID")

oRs.Close

vOdgovor.Add vNarudzbaID, "NarudzbaID"
vOdgovor.Add False, Key:="greska"

oObjectContext.SetComplete

Set OtvoriNarudzbu = vOdgovor

Exit Function

OtvoriNarudzbuErr:

oObjectContext.SetAbort

Set OtvoriNarudzbu = JaviGresku("OtvoriNarudzbu")

End Function

Public Function AzurirajBrojElemenata(ByVal vKosaricaId As
Variant, ByVal vKolicina As Variant) As Variant

```

```
On Error GoTo AzurirajBrojElemenataErr

Dim vOdgovor As New Collection

Dim bRet As Boolean

Dim vSql As Variant

Let vSql = "SELECT * FROM KosariceSadrzaj " & _
           " WHERE KosaricaId = " & vKosaricaId & ";"

Set oRs = oObjectContext.CreateInstance("ADODB.Recordset")

oRs.CursorType = adOpenDynamic
oRs.LockType = adLockPessimistic
oRs.Open vSql, vSpojNaBazu

If oRs.EOF Then
    bRet = False
Else
    oRs.Fields("Kolicina") = CLng(vKolicina)
    oRs.Update
    bRet = True
End If

oRs.Close

vOdgovor.Add False, Key:="greska"
```

```
oObjectContext.SetComplete
```

```
Set AzurirajBrojElemenata = vOdgovor
```

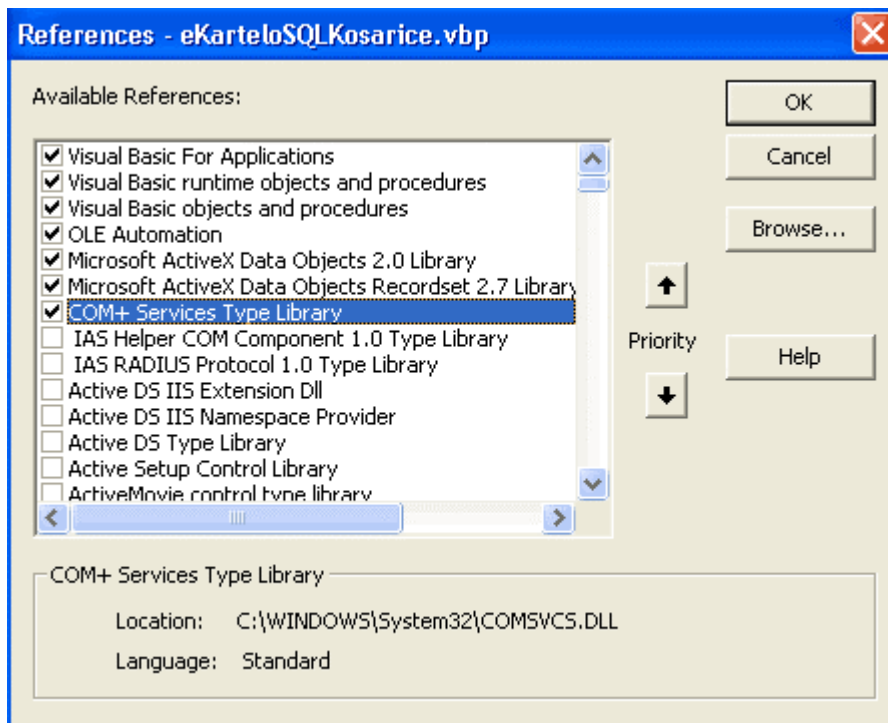
```
Exit Function
```

```
AzurirajBrojElemenataErr:
```

```
oObjectContext.SetAbort
```

```
Set AzurirajBrojElemenata =  
JaviGresku("AzurirajBrojElemenata")
```

```
End Function
```



eKarteloASPKosarice.dll

Class modul (objekt) Kosarice (Kosarice.cls)

'

Option Base 0

Option Explicit

Implements ObjectControl

Private oSQLKosarice As eKarteloSQLKosarice.SQLKosarice

Private oObjectContext As ObjectContext

Private Function ObjectControl_CanBePooled() As Boolean

 ObjectControl_CanBePooled = True

End Function

Private Sub ObjectControl_Deactivate()

 Set oObjectContext = Nothing

 Set oSQLKosarice = Nothing

End Sub

Private Sub ObjectControl_Activate()

 On Error GoTo OCActivateErr

```

Set oObjectContext = GetObjectContext()

Set oSQLKosarice =
oObjectContext.CreateInstance("eKarteloSQLKosarice.SQLKosarice")

Exit Sub

OCActivateErr:

Call JaviGresku("OCActivate")

End Sub

Public Function VратиElementeKosarice(ByVal vNarudzbaID As
Variant, ByVal vKupacID As Variant) As Variant

On Error GoTo VратиElementeKosariceErr

Dim vOdgovor As Variant

Dim vKosarica As Variant

Dim vKosaricaItem As Variant

Dim vCijena As Variant

Dim vPromidzbaKod As Variant

Dim vPromidzbaKraj As Variant

Dim vDanasnjaCijena As Variant

If vNarudzbaID = "" Then

Set vOdgovor = oSQLKosarice.OtvoriNarudzbu(vKupacID)

vNarudzbaID = vOdgovor("NarudzbaID")

```



```

Trace "NarudzbaID - " & vNarudzbaID
End If

Set vOdgovor = oSQLKosarice.VratiElementeKosarice(vNarudzbaID)

If Not vOdgovor("greska") Then

    Set vKosarica = vOdgovor("elementi")

    For Each vKosaricaItem In vKosarica

        vCijena = vKosaricaItem("cijena")
        vPromidzbaKod = vKosaricaItem("promidzbakod")
        vPromidzbaKraj = vKosaricaItem("promidzbakraj")

        vDanasnjaCijena = PodesiDanasnjuCijenu(vCijena,
vPromidzbaKod, vPromidzbaKraj)

        vDanasnjaCijena = Format(vDanasnjaCijena, "$#,##0.00")

        vKosaricaItem.Add vDanasnjaCijena,
Key:="danasnjacijena"

    Next

End If

oObjectContext.SetComplete

Set VratiElementeKosarice = vOdgovor

```

Exit Function

VratiElementeKosariceErr:

oObjectContext.SetAbort

Set VratiElementeKosarice =
JaviGresku("VratiElementeKosarice")

End Function

Public Function DodajElement(ByVal vNarudzbaID As Variant, ByVal
vKupacID As Variant, ByVal vKatalogId As Variant, ByVal vKolicina
As Variant) As Variant

On Error GoTo DodajElementErr

Dim vOdgovor As Variant

If vNarudzbaID = "" Then

Set vOdgovor = oSQLKosarice.OtvoriNarudzbu(vKupacID)

vNarudzbaID = vOdgovor("NarudzbaID")

End If

Set vOdgovor = oSQLKosarice.DodajElement(vNarudzbaID,
vKatalogId, vKolicina)

vOdgovor.Add vNarudzbaID, "NarudzbaID"

oObjectContext.SetComplete

Set DodajElement = vOdgovor

Exit Function

DodajElementErr:

oObjectContext.SetAbort

Set DodajElement = JaviGresku("DodajElement")

End Function

Public Function AzurirajBrojElemenata(ByVal vKosaricaId As Variant, ByVal vKolicina As Variant) As Variant

On Error GoTo AzurirajBrojElemenataErr

Dim vOdgovor As Variant

Set vOdgovor = oSQLKosarice.AzurirajBrojElemenata(vKosaricaId, vKolicina)

oObjectContext.SetComplete

Set AzurirajBrojElemenata = vOdgovor

Exit Function

AzurirajBrojElemenataErr:

oObjectContext.SetAbort

Set AzurirajBrojElemenata = JaviGresku("AzurirajBrojElemenata")

End Function

```
Public Function IsprazniKosaricu(ByVal vNarudzbaID As Variant) As Variant
```

```
    On Error GoTo IsprazniKosaricuErr
```

```
    Dim vOdgovor As Variant
```

```
    Set vOdgovor = oSQLKosarice.IsprazniKosaricu(vNarudzbaID)
```

```
    oObjectContext.SetComplete
```

```
    Set IsprazniKosaricu = vOdgovor
```

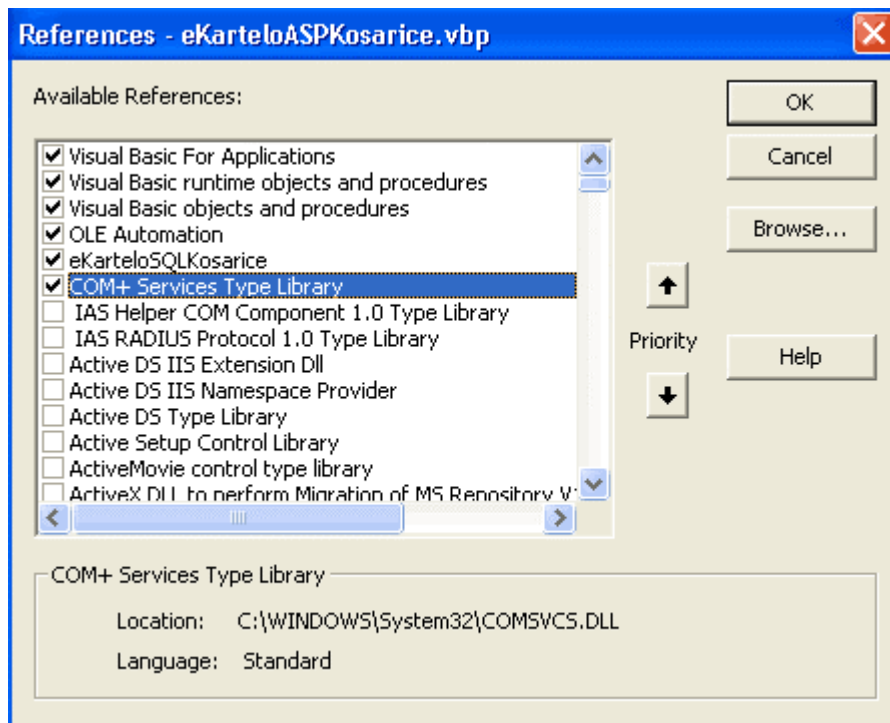
```
Exit Function
```

```
IsprazniKosaricuErr:
```

```
    oObjectContext.SetAbort
```

```
    Set IsprazniKosaricu = JaviGresku("IsprazniKosaricu")
```

```
End Function
```



eKarteloASPObavljanjeNarudzbe.dll

Class modul (objekt): Narudzba (Narudzba.cls)

```
'  
  
Option Base 0  
  
Option Explicit  
  
Implements ObjectControl  
  
Private oSQLKosarice As eKarteloSQLKosarice.SQLKosarice  
  
Private oObjectContext As ObjectContext
```

```

Private Function ObjectControl_CanBePooled() As Boolean
    ObjectControl_CanBePooled = True
End Function

Private Sub ObjectControl_Deactivate()
    Set oObjectContext = Nothing
    Set oSQLKosarice = Nothing
End Sub

Private Sub ObjectControl_Activate()
    On Error GoTo OCActivateErr

    Set oObjectContext = GetObjectContext()

    Set oSQLKosarice =
oObjectContext.CreateInstance("eKarteloSQLKosarice.SQLKosarice")

    Exit Sub

OCActivateErr:

    Call JaviGresku("OCActivate")

End Sub

Public Function IzgradiDetaljeTransakcije(ByVal vNarudzbaID As
Variant) As Variant

    On Error GoTo IzgradiDetaljeTransakcijeErr

    Dim vOdgovor As Variant

    Dim vKosarica As Variant

```

```
Dim vKosaricaItem As Variant
Dim vCijena As Variant
Dim vPromidzbaKod As Variant
Dim vPromidzbaKraj As Variant
Dim vDanasnjaCijena As Variant
Dim vCijenaIsporuke As Variant
Dim vIznosPoreza As Variant
Dim vUkupnaCijena As Variant
Dim vKolicina As Variant
Dim vUkupnaCijenaNarudzbe As Variant
Dim vNarudzba As Collection
Dim vNarudzbaItem As Collection

vUkupnaCijenaNarudzbe = 0
vCijenaIsporuke = 4
Set vOdgovor = oSQLKosarice.VratiElementeKosarice(vNarudzbaID)
If Not vOdgovor("greska") Then

    Set vKosarica = vOdgovor("elementi")

    For Each vKosaricaItem In vKosarica
        vCijena = vKosaricaItem("cijena")
        vPromidzbaKod = vKosaricaItem("promidzbakod")
        vPromidzbaKraj = vKosaricaItem("promidzbakraj")
        vKolicina = vKosaricaItem("kolicina")
```

```

        vDanasnjaCijena = PodesiDanasnjuCijenu(vCijena,
vPromidzbaKod, vPromidzbaKraj)

        vDanasnjaCijena = Format(vDanasnjaCijena,
"$###,##0.00")

        vKosaricaItem.Add vDanasnjaCijena,
Key:="danasnjacijena"

        vUkupnaCijena = vKolicina * vCijena

        vUkupnaCijena = Format(vUkupnaCijena, "$###,##0.00")

        vKosaricaItem.Add vUkupnaCijena, Key:="ukupnacijena"

        vUkupnaCijenaNarudzbe = vUkupnaCijenaNarudzbe +
vUkupnaCijena

        vCijenaIsporuke = vCijenaIsporuke + (1.5 *
CDec(vKolicina))

Next

Set vNarudzba = New Collection

Set vNarudzbaItem = New Collection
vNarudzbaItem.Add vKosarica, Key:="kosaricaelementi"
vNarudzba.Add vNarudzbaItem, "kosarica"

Set vNarudzbaItem = Nothing

Set vNarudzbaItem = New Collection
vUkupnaCijena = Format(vCijenaIsporuke, "$###,##0.00")
vNarudzbaItem.Add vUkupnaCijena, Key:="ukupnacijena"
vNarudzbaItem.Add "$4 and $1.5 per item", "detalji"

```



```

vNarudzba.Add vNarudzbaItem, "isporuka"

vUkupnaCijenaNarudzbe = vUkupnaCijenaNarudzbe +
vCijenaIsporuke

Set vNarudzbaItem = Nothing

Set vNarudzbaItem = New Collection

vIznosPoreza = vUkupnaCijenaNarudzbe * 0.125
vUkupnaCijena = Format(vIznosPoreza, "$###,##0.00")
vNarudzbaItem.Add vUkupnaCijena, Key:="ukupnacijena"
vNarudzbaItem.Add "12.5%", "detalji"
vNarudzba.Add vNarudzbaItem, "porez"

vUkupnaCijenaNarudzbe = vUkupnaCijenaNarudzbe +
vIznosPoreza

Set vNarudzbaItem = Nothing

Set vNarudzbaItem = New Collection

vUkupnaCijena = Format(vUkupnaCijenaNarudzbe,
"$###,##0.00")

vNarudzbaItem.Add vUkupnaCijena, Key:="ukupnacijena"
vNarudzba.Add vNarudzbaItem, "narudzbaukupno"

Set vNarudzbaItem = Nothing

vNarudzba.Add False, Key:="greska"

Set vOdgovor = vNarudzba

End If

```

```
oObjectContext.SetComplete
```

```
Set IzgradiDetaljeTransakcije = vOdgovor
```

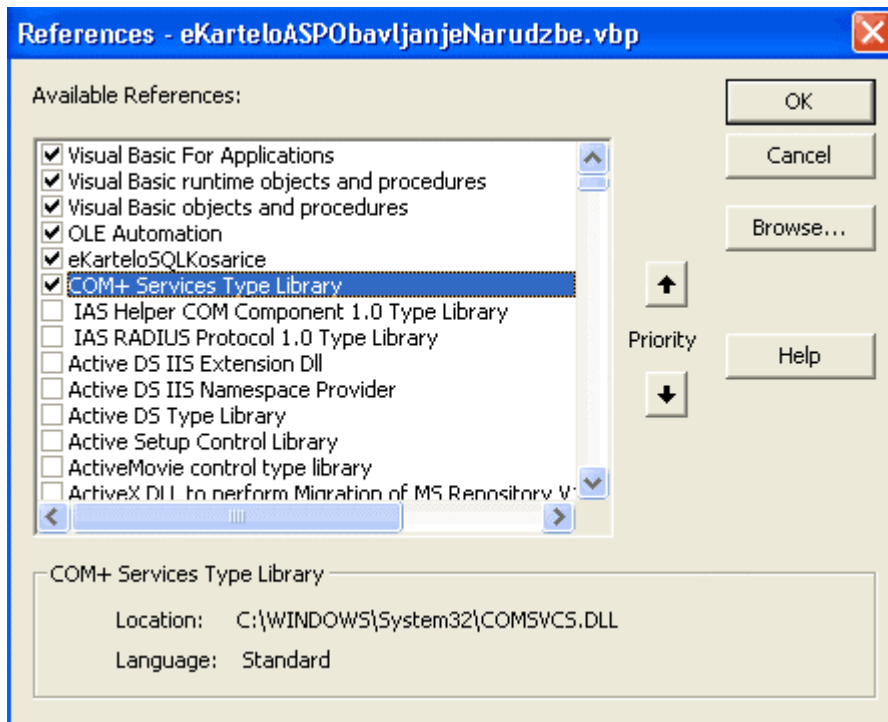
```
Exit Function
```

```
IzgradiDetaljeTransakcijeErr:
```

```
oObjectContext.SetAbort
```

```
Set IzgradiDetaljeTransakcije =  
JaviGresku("IzgradiDetaljeTransakcije")
```

```
End Function
```



Lekcija 11 E- knjižara Kartelo u radu

Primjer e-knjižare je primjer e-commercea (e-trgovine), danas najčešće web aplikacije na Internetu. Namjera mi je napraviti e-trgovinu po mjeri e-kupca. *Kupcoorijentirana* ili *kupcocentrična* e-trgovina. Postoji i component-orijentirana ili component-centrična aplikacija. Moja e-trgovina ima 8 komponenti napravljenih u Visual Basic-u 6. Da je *komponentnocentrična*, imala bi samo jednu komponentu. Kada pravimo *kupcocentričnu* web aplikaciju analiziramo postupke on-line kupca. Svi pođemo na www.amazon.com i kupimo knjigu, drugu, treću, četvrtu. Pokušamo i na www.bn.com i otkrijemo da nam je draže kupovati na jednoj nego na drugoj URL.

Od čega se sastoji e-knjižara

<i>asp</i> stranice/komponente/baza	Količina
<i>asp</i> stranica (zajedno s .inc, .asa, .css)	15
COM komponenti	8
baza podataka	1

Koji sam software koristio za izradu i rad

e-knjižare

Knjige koje sam koristio, a popisane su u poglavlju *Literatura*, izdane su u razdoblju od 1998. do 2000. godine. Tada je još bio glavni Windows NT4 i njegov Option Pack NT4. U Option Pack-u se nalazi IIS4 i PSW (klijent verzija poslužitelja za Windows 95), i knjige su pisane za tu tehnologiju. Neke knjige rade i s VB 5, no ja sam sve radio u Visual Basic-u 6 (Visual Studio 6, s InterDev-om kao ASP editorom).

Tijekom rada premjestio sam se na Windows 2000 Professional i njegovu klijent verziju poslužitelja IIS5, a na kraju, posljednje sam radnje izveo na Windows XP Professional i IIS5.1.

E-knjižara, koju sada imate uz ovu knjigu, radi jednako na svim tim platformama, pa vjerujem da nećete imati problema na kojem god sustavu vi radili kod kuće, u školi ili na poslu.

Prezentacijski, biznis i podatkovni slojevi web aplikacije e-knjižara

Sve ASP stranice spadaju u *prezentacijski sloj* i smještene su u *Start* mapu web aplikacije *ekartelo*. Vidjet ćemo kasnije kako se kreira start mapa. Sve naknadne promjene u prezentacijskom sloju vršimo u kôdu ovih stranica, dodajemo nove itd., bez ikakve potrebe za intervencijom u ostala dva sloja.

Sve DLL komponente su smještene u podmapi Start mape, *KarteloKonacnaVerzija*, ali ih možete staviti na svoje računalo gdje god želite. Ove COM komponente čine srednji sloj web aplikacije koji se naziva sloj *poslovne logike*.

Pet od 8 komponenti radi samo s ASP stranom (u svom nazivu imaju '...ASP...'), a ostale 3 komponente rade s bazom podataka (u svom nazivu imaju '...SQL...'). Ta je podjela napravljena radi lakših naknadnih promjena u kôdu. Naime, sva

poslovna logika (procesuiranje) ugrađena je u 5 komponenti koje rade s ASP slojem.

Sve naknadne promjene u poslovnoj logici ugradit ćemo tu, bez potrebe izmjene ijednog retka u ostalim slojevima. Komponente koje rade s bazom podataka izvode standardizirane radnje nad bazom: ADD, DELETE, UPDATE, EDIT, SELECT baze i tu nema potrebe za promjenama na duže staze.

Access relacijska baza podataka *ekartelo.mdb* smještena je u *Start* mapi, ali je možete na svom računalu pohraniti bilo gdje. Access baza sastavljena samo od tablica čini treći sloj koji zovemo slojem izvora podataka. Ukoliko biste napravili SQL bazu, dovoljno je istu prijaviti u DSN sustav poslužitelja pod istim imenom kao i Access bazu, i web aplikacija radi. Iako Access baze ne moramo prijavljivati u DSN sustav poslužitelja, ovaj put sam to učinio. Aplikaciju sam počeo raditi na IIS4 kada se još Access baza morala prijavljivati u DSN sustav, pa je tako i ostalo. U IIS5 se to već ne mora, i naravno, u IIS 5.1 i IIS6 se također ne mora.

Kada nekoliko puta iskušate ovu web aplikaciju, kada se dobro upoznate s njenom konstrukcijom, njenim kôdom i protokom informacija kroz nju, vidjet ćete koliko će vam lako biti raditi izmjene u svim njenim slojevima i prilagoditi je svojim potrebama.

Kada još k tome znamo da se VB i VBScript gotovo ne razlikuju u kôdu, a s VB-om već imamo iskustva, onda smo "kod kuće" i u sloju ASP stranica, gdje vlada VBScript, i u sloju poslovne logike, gdje radimo s Visual Basic-om.

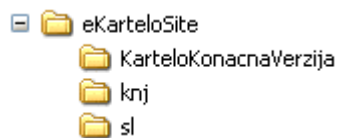
Sva poslovna logika je kompajlirana i tako nedostižna pogledima konkurencije, ili bilo čijim drugim. Komponente se mogu koristiti i u svim našim kasnijim aplikacijama. Na primjer, imamo dvije komponente koje u svom nazivu imaju "...kupci..." i bave se provjerom i ažuriranjem podataka korisnika naše aplikacije. Možemo ih koristiti i u svim našim budućim web aplikacijama.

Instalacija e-knjižare

Download

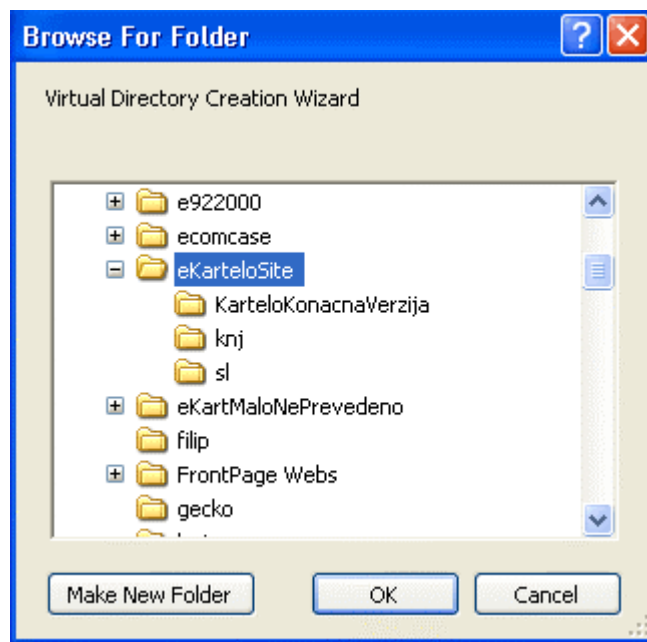
1. Na URL www.e92.hr/knjigaasp.zip se nalazi ZIP datoteka ekartelosite.zip.
Download.

2. Prilikom ražimanja ZIP datoteke, pohranite je na C:\ i ne zaboravite uključiti opciju Use folder names. Tako ćete na C:\ dobiti strukturu mapa:



IIS 5.1

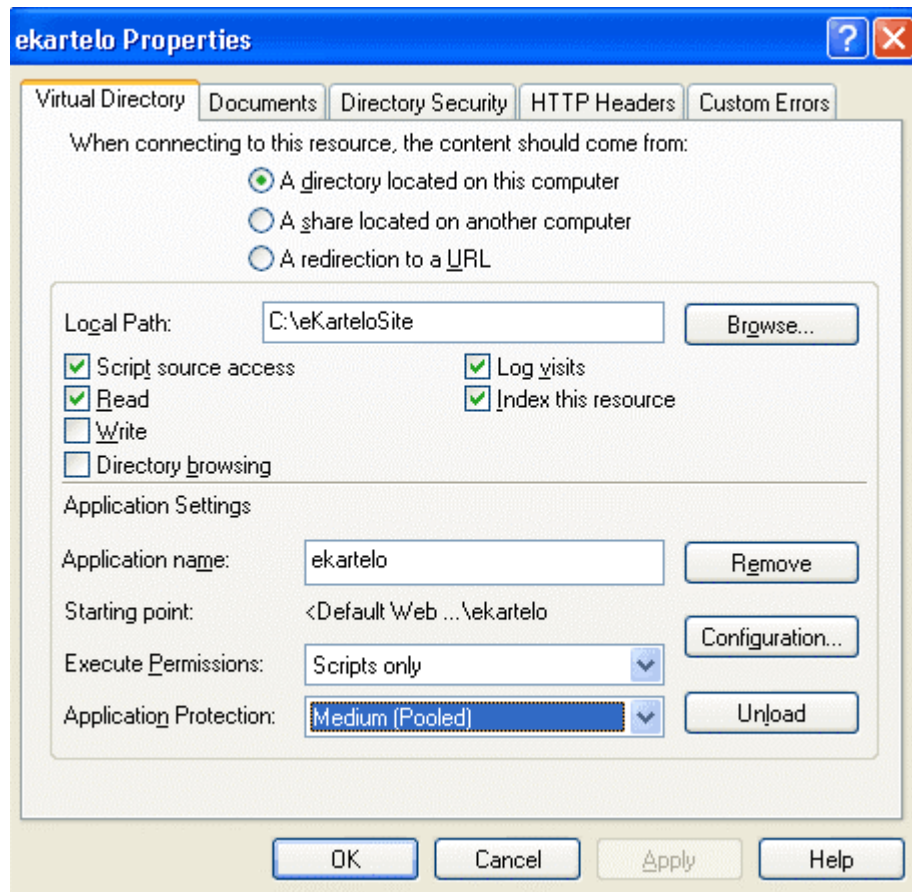
1. Otvorite IIS 5.1 (Start/Control Panel/Administrative Tools/Internet Information Services).
2. Pritisnite desnu tipku miša na Default Web Site i u skraćenom izborniku odaberite New/Virtual Directory.
3. Next.
4. U polje Alias tipkajte **ekartelo**.
5. Next.
6. Odaberite gumb Browse, pa u dijalogu koji slijedi označite mapu eKarteloSite na C:\...



7. Next.
8. Neka su uključene opcije Read i Run scripts, pa gumb Next.
9. Finish. Virtualna mapa je kreirana.

Svojstva web aplikacije

1. Pritisnite desnu tipku miša na ekartelo u IIS5.1, pa odaberite Properties. Odaberite opcije kao na slici dolje.

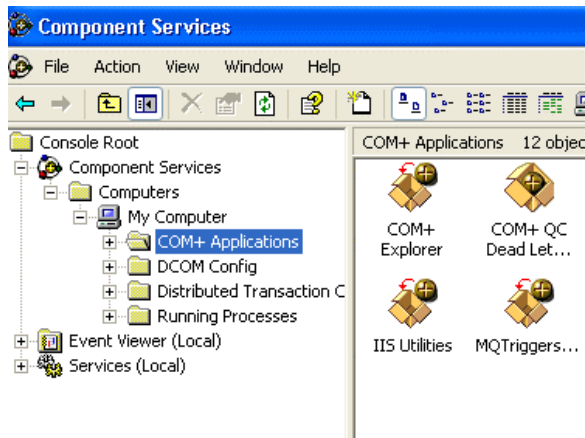


2. U polje Application name tipkajte ime aplikacije **ekartelo**.
3. Start mapa je ona mapa čiji je put napisan u polju Local path. To je naša virtualna mapa ekartelo.

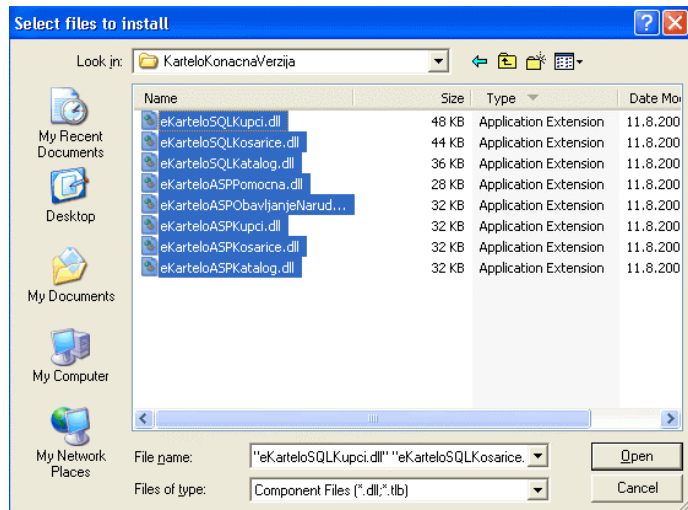
Component Services(IIS 5.1 Windows XP Professional)

Componente Services (IIS 5.1 Windows XP Professional), u verziji Windows 2000 Professional (IIS 5.0), ima naziv COM+ , a u Windows NT4 (IIS4) naziv Microsoft Transaction Server.

1. Otvorite Component Services (Start/Control Panel/Administrative Tools/Component Services).

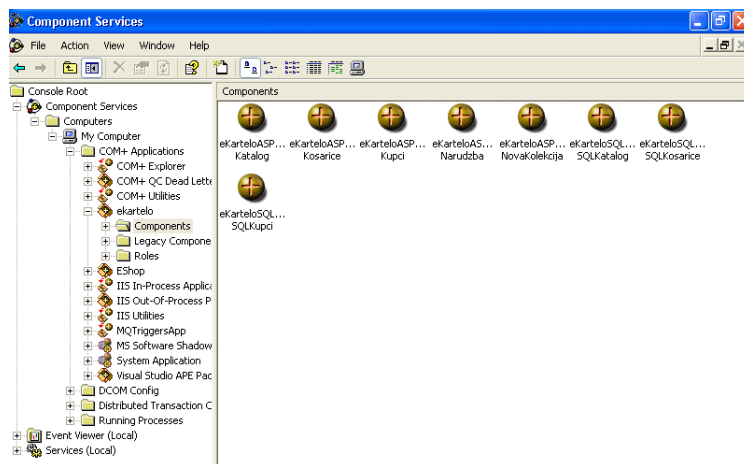


2. Pritisnite desnu tipku miša na COM+ Applications, pa u skraćenom izborniku odaberite New/Application/Next.
3. Odaberite Create an empty application.
4. Tipkajte ime web aplikacije **ekartelo**. Uključite opciju Server application.
5. Next. Uključite This user i u mom slučaju tipkam u polje User: administrator, a ostala su polja prazna.
6. Next. Finish.
7. Razgranajte upravo kreirano ime aplikacije ekartelo. Pritisnite desnu tipku miša na Components, pa odaberite New/Component.
8. Next.
9. Instal new component(s).
10. Označite sve DLL datoteke u mapi C:\eKarteloSite\KarteloKonacnaVerzija\



11. Open.

12. Next. Finish.

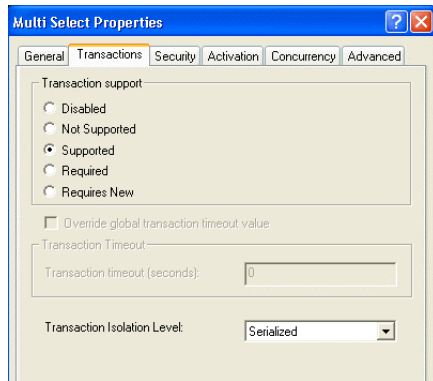


Ovim postupkom 8 komponenti je registrirano. Ukoliko želite ukloniti komponente s vašeg računala, dovoljno ih je na ovom mjestu obrisati.

Budući da transaction server ne podržava Access baze, uključite podršku transakciji samo onim komponentama koje u svom nazivu imaju ASP. Komponente koje u nazivu imaju SQL surađuju s Access bazom i podrška transakciji neka je isključena.

Postupak aktiviranja podrške transakciji:

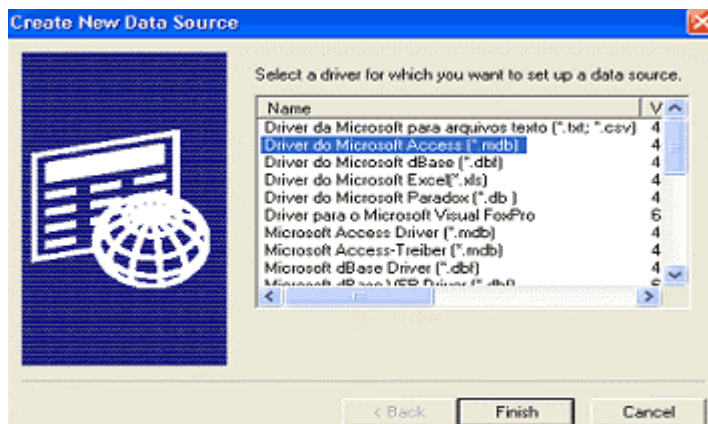
1. Označite pet komponenti koje u svom nazivu imaju ASP, pritisnite desnu tipku miša i odaberite Properties. Odaberite jahača Transactions i uključite opciju Supported.



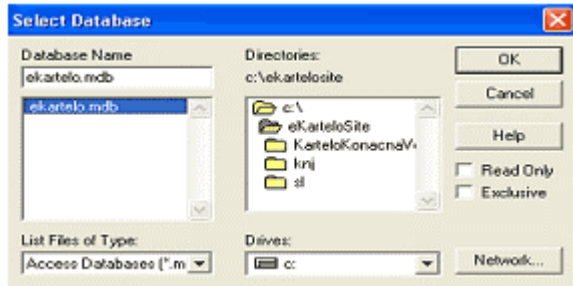
2. OK. Ostale su tri komponente – odaberite opciju Not Supported.

Prijava baze u DSN

1. Start/Control Panel/Administrative Tools/Data Sources (ODBC)/System DSN/



2. Označite Driver do Microsoft Access (*.mdb). Finish.
3. U Data Source Name tipkajte **ekartelo**.
4. Odaberite Select i označite bazu ekartelo.mdb.

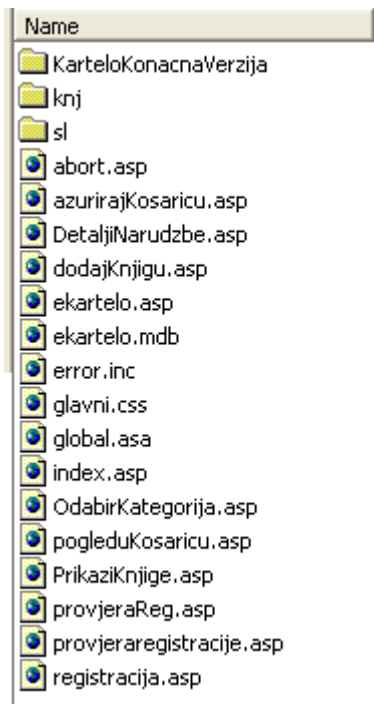


5. OK.

Kreiran je virtualni direktorij u IIS, kreiran je paket COM komponenti u Component Services i prijavljena je baza u DSN. Možemo provjeriti kako radi naša web aplikacija e-knjižara Kartelo.

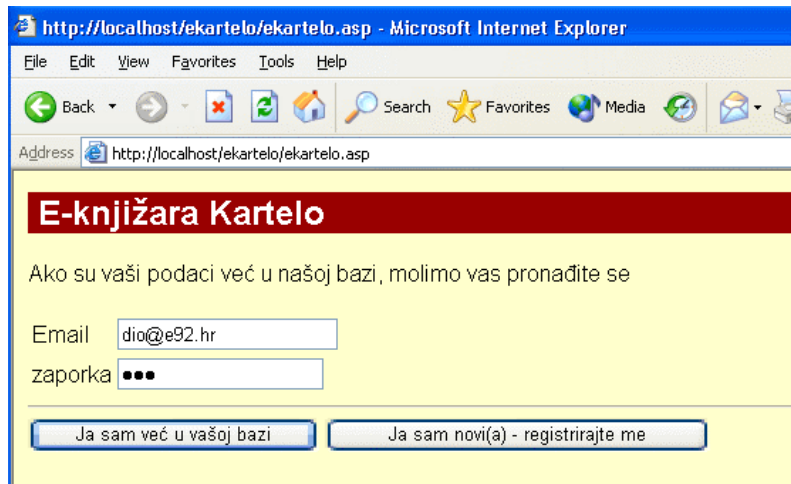
Provjerite e-knjižaru Kartelo

1. U IIS5.1 otvorite virtualnu mapu ekartelo, u kojoj imate sljedeće podmape i ASP stranice:

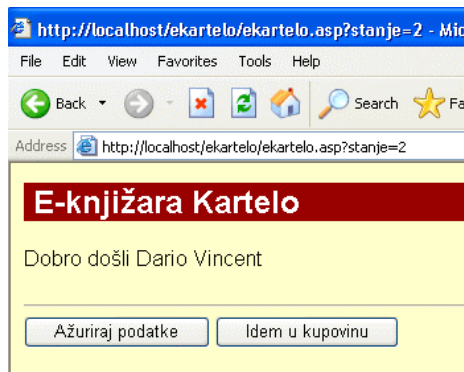


2. Desna tipka miša na index.asp, pa Browse.

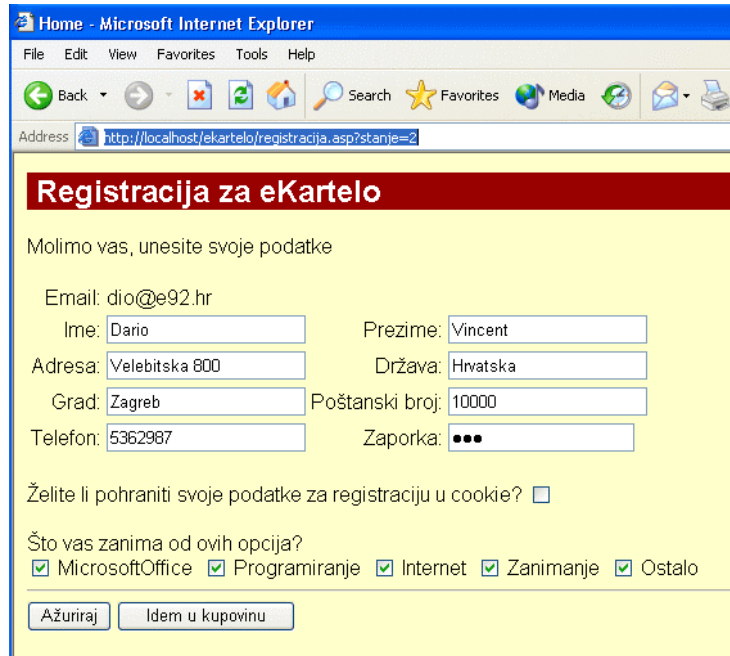
- Uđimo u e-knjižaru pritiskom tipke miša na eKartelo.
- Tipkajte dio@e92.hr i zaporku **dio** kupca koji već postoji u bazi. Obratite pozornost na URL u polju Address browser-a: <http://localhost/ekartelo/ekartelo.asp> Znači, prva ASP stranica naše web aplikacije koja se otvara je ekartelo.asp.



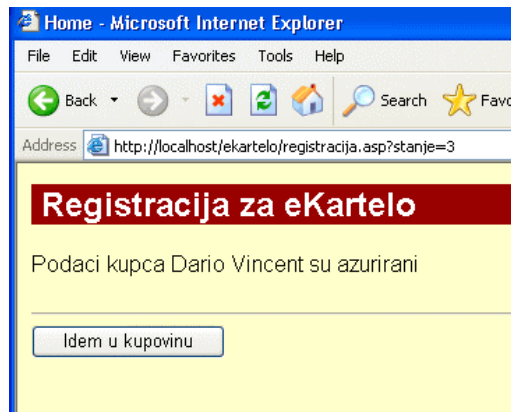
- Odaberite gumb 'Ja sam već u vašoj bazi'. Pogledajte Address: <http://localhost/ekartelo/ekartelo.asp?stanje=2> Ostali smo na istoj ASP stranici, samo je varijabla **stanje** i vrijednost te varijable **stanje=2** dovelo iz iste ASP stranice ovu web stranicu u naš browser:



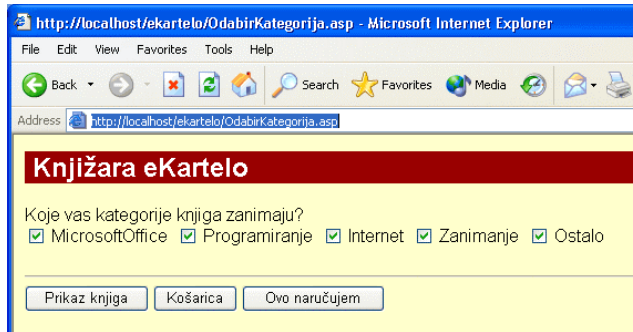
- Odaberite 'Ažuriraj podatke'. Pogledajte u Address: <http://localhost/ekartelo/registracija.asp?stanje=2> Sada smo na ASP stranici registracija.asp i na njoj koristimo varijablu stanje za kreiranje više web stranica pomoću jedne ASP stranice.



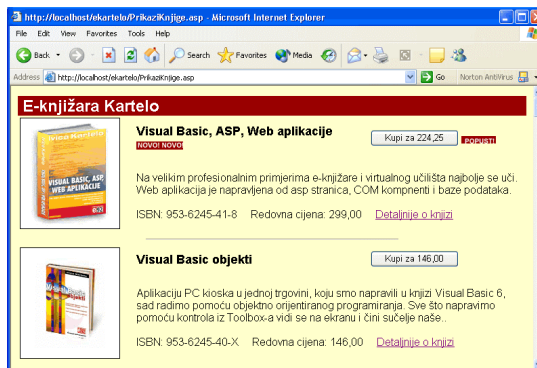
- Promijenite Velebitska 800 u Velebitska 790.
- Odaberite 'Ažuriraj'. Pogledajte u Address: <http://localhost/ekartelo/registracija.asp?stanje=3> Ista ASP stranica ovaj put izbacuje web stranicu sukladno vrijednosti varijable stanje = 3.



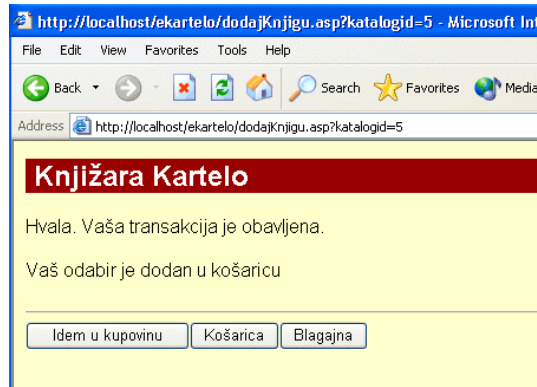
- Odaberite 'Idem u kupovinu'. U Address browser-a je URL: <http://localhost/ekartelo/OdabirKategorija.asp> Sada smo na ASP stranici OdabirKategorija.asp



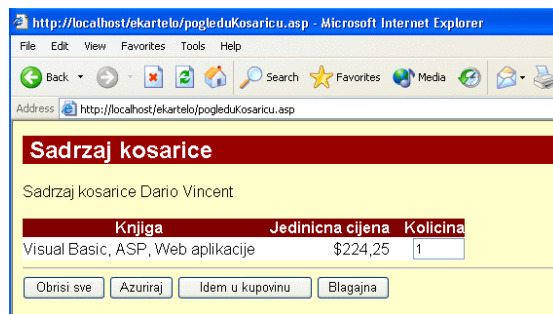
10. Odaberite 'Prikaz knjiga'. Address browser-a: <http://localhost/ekartelo/PrikaziKnjige.asp> Nova ASP stranica PrikaziKnjige.asp.



11. Obratite pozornost na znak 'POPUST' i znak NOVO!NOVO!. To su GIF datoteke koje se pojave kada u bazi napišemo tip promidžbe (koliki je popust) i datum izlaska knjige iz tiska.
12. Odaberite 'Kupi'. Address u browser-u: <http://localhost/ekartelo/dodajKnjigu.asp?katalogid=5> Nova ASP stranica dodajKnjigu.asp. Ova ASP stranica ima naredbu `<% @ TRANSACTION=REQUIRED %>` Pojava gumba na dnu stranice je znak da je Sub procedura na dnu ASP stranice procesuirana i da je transakcija uspješna.

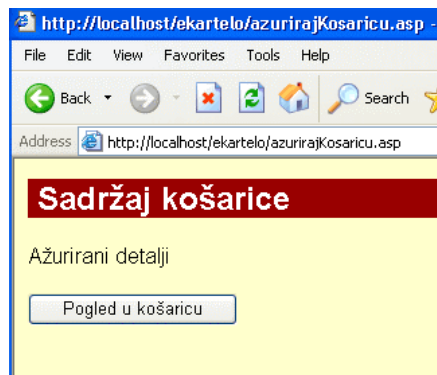


13. Odaberite 'Košarica'. Pogledajte Address: <http://localhost/ekartelo/pogleduKosaricu.asp> Nova ASP stranica [pogleduKosaricu.asp](http://localhost/ekartelo/pogleduKosaricu.asp).



Naša se kupovina se ostvaruje u ovoj edukacijskoj verziji, pa vi možda vidite i više redaka posljednje nedovršene narudžbe.

14. Promijenite količina u 5. Odaberite gumb 'Azuriraj'. Pogledajte u Address: <http://localhost/ekartelo/azurirajKosaricu.asp> Nova ASP stranica [azurirajKosaricu.asp](http://localhost/ekartelo/azurirajKosaricu.asp).



15. itd. Vježbajte s novim kupcem. Gledajte u Address browser-a i vidjet ćete koja je stranica poslala web stranicu browser-u i kako izgleda HTTP zahtjev.

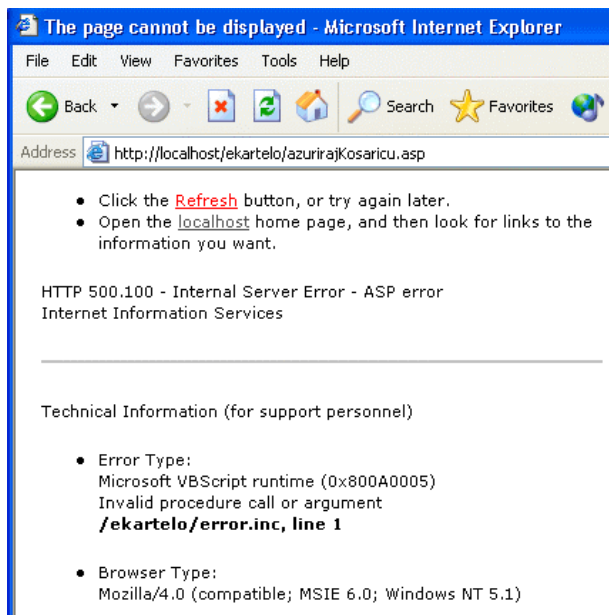
Napomena: e-knjižara je edukacijska inačica tipične profesionalne izvedbe. Ima još nekoliko detalja za doraditi, a kraj aplikacije svatko od nas će ipak dogovoriti s administratorom hostinga. Moja je želja spojiti aplikaciju na vanjskog davatelja usluge provjere kreditnih kartica kako bismo maksimalno bili rasterećeni od ručnog rada ovdje u Školi.

Greške i kako ih ukloniti

Na nekoliko mjesta web aplikacija će vam javiti grešku. Nalazim se na stranici:

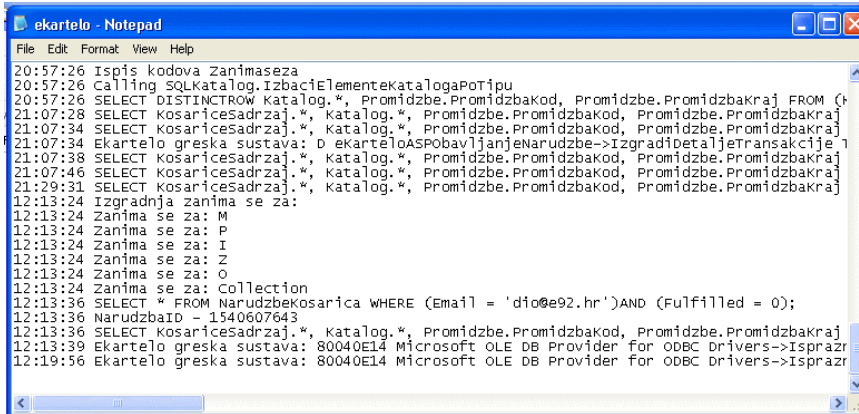
<http://localhost/ekartelo/pogledUKosaricu.asp>

Odabirem gumb 'Obriši sve'. Otvara se ovaj prozor:



Greška je u prvom retku datoteke error.inc, redak 1. Pokušavam odabrati gumb Blagajna. Greška. Ponovo biram Obriši sve. Sada ispravno radi.

A sada pogledajmo tekstualnu datoteku C:\ekartelo.txt. Imate je i vi na svom računalu jer naša web aplikacija automatski kreira tu datoteku i onda u nju upisuje sve što smo joj zadali naredbom Trim (vidjeti kôd).



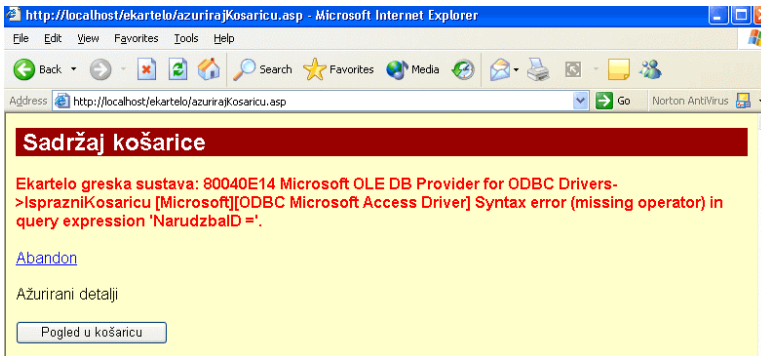
```
ekartelo - Notepad
File Edit Format View Help
20:57:26 Ispis kodova Zanimaseza
20:57:26 Calling SQLKatalog.IzbaciElementeKatalogaPoTipu
20:57:26 SELECT DISTINCTROW Katalog.*, Promidzbe.Promidzbakod, Promidzbe.Promidzbakraj FROM (
21:07:28 SELECT KosariceSadrzaj.*, Katalog.*, Promidzbe.Promidzbakod, Promidzbe.Promidzbakraj
21:07:34 SELECT KosariceSadrzaj.*, Katalog.*, Promidzbe.Promidzbakod, Promidzbe.Promidzbakraj
21:07:34 Ekartelo greska sustava: D ekarteloASpobavljanjeNarudzbe->IzgradiDetaljeTransakcije
21:07:38 SELECT KosariceSadrzaj.*, Katalog.*, Promidzbe.Promidzbakod, Promidzbe.Promidzbakraj
21:07:46 SELECT KosariceSadrzaj.*, Katalog.*, Promidzbe.Promidzbakod, Promidzbe.Promidzbakraj
21:29:31 SELECT KosariceSadrzaj.*, Katalog.*, Promidzbe.Promidzbakod, Promidzbe.Promidzbakraj
12:13:24 Izgradnja zanima se za:
12:13:24 Zanima se za: M
12:13:24 Zanima se za: P
12:13:24 Zanima se za: I
12:13:24 Zanima se za: Z
12:13:24 Zanima se za: O
12:13:24 Zanima se za: Collection
12:13:36 SELECT * FROM NarudzbeKosarica WHERE (Email = 'dio@e92.hr')AND (Fulfilled = 0);
12:13:36 NarudzbaID - 1540607643
12:13:36 SELECT KosariceSadrzaj.*, Katalog.*, Promidzbe.Promidzbakod, Promidzbe.Promidzbakraj
12:13:39 Ekartelo greska sustava: 80040E14 Microsoft OLE DB Provider for ODBC Drivers->IsprazniKosaricu
12:19:56 Ekartelo greska sustava: 80040E14 Microsoft OLE DB Provider for ODBC Drivers->IsprazniKosaricu [Microsoft][ODBC Microsoft Access Driver] Syntax error (missing operator) in query expression 'NarudzbaID ='.
```

U zadnjem retku piše:

```
12:19:56 Ekartelo greska sustava: 80040E14 Microsoft OLE DB Provider for ODBC Drivers->IsprazniKosaricu [Microsoft][ODBC Microsoft Access Driver] Syntax error (missing operator) in query expression 'NarudzbaID ='.
```

Sada ću odabrati Back, pa gumb Blagajna (ili ovo naručujem). Također javlja grešku u prvom retku iste datoteke. U tekstualnom dokumentu ekartelo.txt ništa ne piše o grešci. Treba vidjeti do koje zadnje naredbe Trim je stigla naša aplikacija.

Kada sam otklonio tipografsku grešku u prvom retku error.inc, nisam otklonio i glavnu grešku. Sada sam dobio samo tekst greške i u browser-u, isti onaj tekst koji sam već imao u ekartelo.txt:



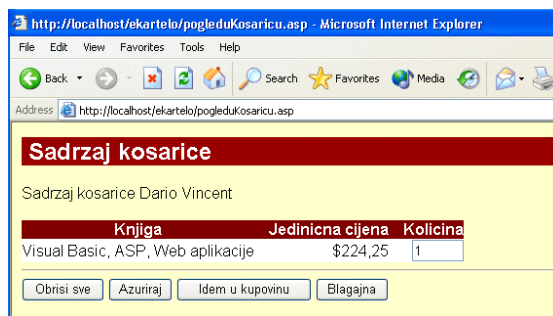
Zadatak za vježbu

Cilj e-knjižare Kartelo jest obrazovanje, a ne imati gotovu "ključ u ruke" aplikaciju koju samo treba staviti na poslužitelj. Do trenutka izlaska knjige u tisak,

u aplikaciji su radile sve komponente osim komponente eKarteloASPObavljanjeNarudzbe.dll. Ta komponenta ima jedan objekt, Narudzba.cls i u tom objektu jednu metodu: IzgradiDetaljeTransakcije. U ovoj komponenti se mogu staviti i svi ostali potrebni objekti: forma za upis podataka s kreditne kartice i slanje tih podataka vanjskom davatelju usluge provjere valjanosti kartice, slanje podataka prema službi koja vrši pakiranje i otpremu itd.

Odobrenje: odobravam kupcima ove knjige slobodnu uporabu kôda web aplikacije e-knjižara Kartelo, pojedinih njenih dijelova, izmjene po volji i sl., kako u obrazovne tako i u komercijalne svrhe. Isto vrijedi i za komponente kao cjeline i za sve ASP stranice.

1. Nakon što ste dobro proučili cijeli kôd web aplikacije, pravite izmjene u kôdu, prilagođavajte aplikaciju svojim potrebama, pronađite i uklonite greške.
2. Promijenite formatiranje u lokalno, tj. da se umjesto znaka dolara \$ pojavi oznaka domaće valute 'kn'.



A sada možemo prijeći na analizu programskog kôda. Paralelno s praćenjem ispisa programskog kôda u knjizi ili na zaslonu, pokrenite aplikaciju kao gore i promatrajte proizvode tog kôda u browser-u na zaslonu.

Lekcija 12 E- knjižara Kartelo analiza ASP kôda

U ASP kôdu i u Visual Basic kôdu mnogo se koriste kolekcije.

index.asp

Stranica index.asp, iako sam joj dao nastavak asp, nema na sebi ASP kôda i mogla je biti i .htm. Davanje nastavka .asp svim stranicama u web aplikaciji danas je dobra navika jer takve se stranice istom brzinom učitavaju kao da su .htm. U tijeku rada na aplikaciji, ako u nekim stranicama odmah i ne bude potrebe za ASP kodiranjem, ne znači da već sljedeći trenutak nećete dodati neki kôd. Tako ćemo na stranici index.asp rado staviti ASP kôd: trenutno je u našoj knjižari 1000 posjetitelja.

ekartelo.asp

Ova jedna ASP stranica šalje puno web stranica browser-u uz pomoć varijable *stanje*. Nakon ASP naredbe `option explicit` slijede deklaracije varijabli, pa sekcija `<BODY>` i u njoj HTML početka web stranice.

```
<table width=100% cellspacing=0 cellpadding=0 border=0>
```

```

<tr class="zaglavljestranice">
<th align=left>&nbsp;E-knji&#382ara Kartelo
</th>
</tr>
</table>

```

Slijedi VBScript. Ako stanje član kolekcije QueryString ASP objekta Request nije prazan, tada...

```
<% if not Request.QueryString("stanje") = "" then %>
```

...varijabli stanje pridodajemo vrijednost zatečenu u memoriji poslužitelja na lokaciji Request.QueryString("stanje")...

```
<%   stanje = Request.QueryString("stanje")   %>
```

```
<% else %>
```

...u suprotnome, dakle ako postoji neka vrijednost različita od NULL "" i ako je potvrdno sljedeće IF...

```
<%   if Session("posjetitelj") = true then %>
```

... neka vrijedi...

```
<%       stanje = 3 %>
```

U Global.asa je kreiran član kolekcije Session 'posjetitelj'. Svaka sesija (posjeta) počinje s vrijednošću false tog člana, a to je vidljivo u Global.asa...

```
Session("posjetitelj") = false
```

...i tek nakon obavljene registracije poprima vrijednost true. Tako ćemo u svakom trenutku trajanja posjete znati je li korisnik registriran ili ne.

Nastavljamo s kôdom u stranici ekartelo.asp.

```
<%   else %>
```

...a ako nije true (tj. ako korisnik nije registriran), neka vrijedi...

```
<%       stanje = 1 %>
```

```
<% end if %>
```

```
<% end if %>
```

Nakon gornje dvostruke strukture If..End If ASP kôd zna ima li posla s neregistriranim ili s registriranim posjetiteljem. Neregistrirane posjetitelje ASP vraća na stanje = 1. Ako na kraju ASP stranice pogledate kôd koji počinje s

```
<% if stanje = 1 then %>
```

```
<p>Ako su va&#353i podaci ve&#263 u na&#353oj bazi, molimo vas  
prona&#273ite se
```

```
<form action="ekartelo.asp?stanje=2" method="POST">
```

```
<table border="0">
```

```
<tr>
```

```
<td>Email</td>
```

```
<td><input type="text" size="20" name="email"
```

```
value="<% =Session("email") %>"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>zaporka</td>
```

```
<td><input type="password" size="20" name="zaporka"
```

```
value="<% =Session("zaporka") %>"></td>
```

```
</tr>
```

```
</table>
```

```
<input type="hidden" name="knjizaral" value="1">
```

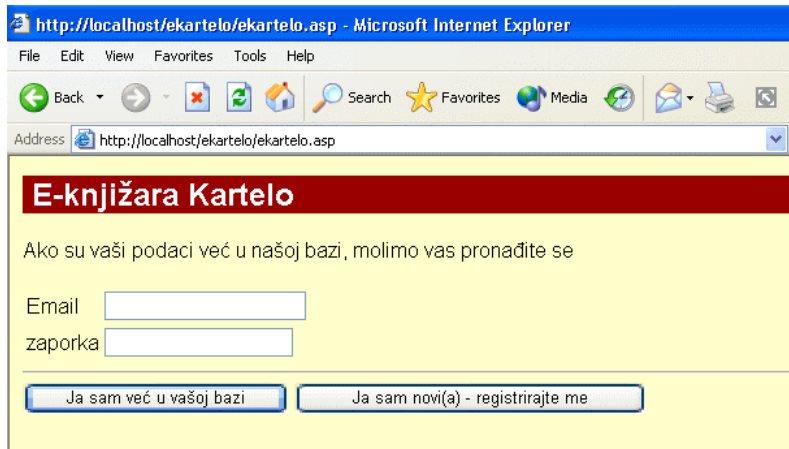
```
<hr>
```

```
<input type="submit" name="submit" value="Ja sam ve&#263 u  
va&#353oj bazi">
```

```
<input type="button" value="Ja sam novi(a) - registrirajte me"  
onclick="window.document.location.href='registracija.asp?stanje=1'  
>
```

```
</form>
<% end if %>
</body>
</html>
```

...i ako tom kôdu na početku dodate onaj HTML s kojim je počela sekcija <BODY>, imat ćete ovu web stranicu u browser-u:



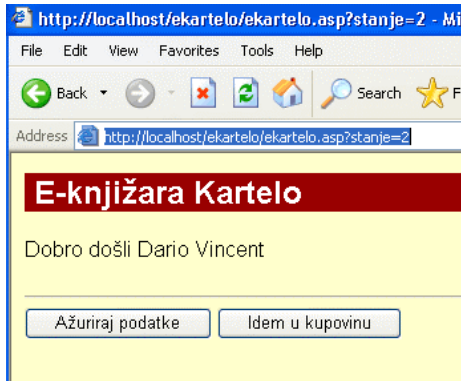
Kada ispunite ovu formu i odaberete gumb tip Submit, bit će poslan HTTP zahtjev definiran u forma-argumentu Action:

```
<form action="ekartelo.asp?stanje=2" method="POST">
```

Kada pak ispunimo oba polja u formi s već postojećim kupcem u bazi i pritisnemo gumb 'Ja sam već u vašoj bazi', prema poslužitelju kreće, kao što znamo, metodom POST, sljedeći HTTP zahtjev:

<http://localhost/ekartelo/ekartelo.asp?stanje=2>

i u browser-u dobijamo odgovor, HTML stranicu...



...koju je proizveo sljedeći dio ASP stranice ekartelo.asp...

```
<p>Dobro do&#353li
```

```
<% Response.Write Session("ime") & " " & Session("prezime") %>
```

```
</p><hr>
```

```
<input type="button" value="A&#382uriraj podatke"
onclick="window.document.location.href='registracija.asp?stanje=2'
">
```

```
<input type="button" value="Idem u kupovinu"
onclick="window.document.location.href='ekartelo.asp?stanje=3' ">
```

U gornjem je kôdu zanimljivo i to kako se racionalno koristi skriptiranje klijent strane. U VBScriptu je napisan HTTP zahtjev koji će se pokrenuti kada se dogodi događaj "onklick" na input type="button", kada pritisnemo gumb "Ažuriraj podatke" ili gumb "Idem u kupovinu".

Poziv komponente

Između dvije gore opisane web stranice, u ASP procesuiranju se dogodilo puno toga, nama vrlo zanimljivoga.

Što se sve desilo od trenutka kada smo upisali E-mail *dio@e92.hr* i zaporku *dio* te pritisnuli gumb tipa Submit.

1. POST metodom je prema poslužitelju krenuo HTTP zahtjev <http://localhost/ekartelo/ekartelo.asp?stanje=2> Zanimljivo je gdje su sve smješteni podaci koji se prenose u HTTP zahtjevu, a koje će na poslužitelju uhvatiti ASP objekt Request i pohraniti u memoriju poslužitelja. Podatak stanje=2 se nalazi iza upitnika ? i vidljiv je u Address polju. Ostali podaci koji

se šalju poslužitelju su naš e-mail i zaporka te podatak iz skrivenog tipa polja, name=knjizaral value="1". Skriveno polje se ubacuje kad god želimo označiti odakle stiže HTTP zahtjev. Sada će naš ASP kôd na poslužitelju znati da zahtjev stiže s početne forme.

```
<td><input type="text" size="20" name="email"
<td><input type="password" size="20" name="zaporka"
<input type="hidden" name="knjizaral" value="1">
```

Iz gornja tri retka Forme prema poslužitelju su, POST metodom, krenuli ovi podaci:

```
name="email" value="dio@e92.hr"
```

```
name="zaporka" value="dio"
```

```
namw="knjizaral" value="1"
```

2. Na početku ASP stranice, ASP kôd će utvrditi da je varijabla *stanje* = 2, jer je ta vrijednost upravo stigla u HTTP zahtjevu i "uhvaćena" je u kolekciju *Request.QueryString("stanje")* te predana varijabli *stanje*. Kada je stanje = 2, vrijedi ovaj ASP kôd:

```
<% if stanje = 2 then %>
<% if Request.Form("knjizaral") = "1" then %>
<%     Session("email") = Request.Form("email") %>
<%     Session("zaporka") = Request.Form("zaporka") %>
<%     if Session("email") = "" or Session("zaporka") = "" then %>
<p class=porukaogresci>Potrebno je ispuniti oba polja</p>
<%     stanje = 1 %>
<%     end if %>
<% end if %>
```

Po Boolovoj algebri, gornji bi kôd mogli tumačiti ovako. "AND" ili "I" logička vrata su dva IF. Ako su oba IF zadovoljena, tj. stanje = 2 i posjetitelj dolazi iz obrasca knjizaral, samo tada će se procesuirati kôd...


```
<% Session("email") = Request.Form("email") %>
```

```
<% Session("zaporka") = Request.Form("zaporka") %>
```

...u kojem se vrijednosti poslane iz Forme, uhvaćene u kolekciji Request.Form prenose u kolekciju Session kao članovi *email* i *zaporka*. To znači da ćemo cijelo vrijeme posjete imati na raspolaganju u memoriji poslužitelja vrijednosti *email* i *zaporka* i njihova referencija će biti session("email") i session("zaporka"). Sve je isto pišemo li mala ili velika slova kada koristimo VBScript.

ASP kôd mora imati iskodiranu i situaciju kada zaboravimo unijeti jedno ili oba od potrebnih polja. Tada nas vraća na stanje = 1, ali ovaj put dodaje poruku Potrebno je ispuniti oba polja...

```
<% if Session("email") = "" or Session("zaporka") = "" then %>
```

```
<p class=porukaogresci>Potrebno je ispuniti oba polja</p>
```

```
<% stanje = 1 %>
```

Kada konačno ispunimo oba polja i stanje bude jednako 2, eto nas na drugi dio ASP kôda koji počinje s ...

```
<% if stanje = 2 then %>
```

...a nastavlja s pozivom komponente eKarteloASPPomocna, točnije kreiranjem objekta spoja na objekt NovaKolekcija koji se nalazi u VB komponenti eKarteloASPPomocna...

```
<% set objPomocni =  
Server.CreateObject("eKarteloASPPomocna.NovaKolekcija") %>
```

Objekt spoja je kreiran. Slijedi kreiranje instancije objekta vezanog na metodu NovaKolekcija. Naime, sada vidim da sam metodi trebao dati drugo ime kako ne bi bilo zabune. Isto ime NovaKolekcija dao sam i class formi u VB projektu eKarteloASPPomocna.vbp i metodi u toj class formi.

Naš ASP kôd mora saznati postoji li E-mail dio@e92.hr i zaporka *dio* u bazi podataka ekartelo.mdb, u tablici Kupci...

Kupci : Table								
	Email	Ime	Prezime	Adresa	Grad	PostanskiBroj	Drzava	Telefon
▶	dio@e92.hr	Dario	Vincent	Velebitska 780	Zagreb	10000	Hrvatska	5362987
	iva@e92.hr	Iva	Avi	Visoka 356	Zagreb	10000	Hrvatska	123456
	mar@e92.hr	Martina						
	valentina60333@e92.hr	Valentina	Valentini	Velebitska 890	Trogir	21320	Hrvatska	877-1234

Ova web aplikacija radi u 3 sloja. Srednji je sloj napravljen u COM komponentama. Mogli smo web aplikaciju napraviti isto od 3 sloja, ali i prezentacijski i srednji sloj biznis logike su mogli biti na ASP stranicama. Tada bi u naslovu ove knjige pisalo samo "ASP, baze, Web aplikacije". O prednostima korištenja COM komponenti u srednjem sloju, biznis logici, govorili smo ranije u ovoj knjizi.

Dakle, između našeg ASP kôda u ASP stranici `ekartelo.asp` i baze podataka nalaze se čak dvije komponente. Jednoj komponenti, konkretno komponenti `eKarteloASPKupci` se obraća ASP stranica; ta komponenta kontaktira komponentu `eKarteloSQLKupci`, koja kontaktira bazu i vrati podatke tražitelju tj. komponenti `eKarteloASPKupci`, a ova podatke vrati ASP stranici.

Koja je uloga komponente `eKarteloASPPomocna`?

Budući da VBScript nije isto okruženje kao i Visual Basic od kojeg je napravljena komponenta `eKarteloASPKupci`, ne možemo kreirati VBScript kolekciju koju bi podržavalo okruženje Visual Basic i obrnuto. Zato primjenjujemo sljedeći trik. VBScript kôd može kontaktirati VB komponente. U VB komponentama se nalaze class forme. VBScript može napraviti instanciju objekta na bazi te class forme. Na class formama se nalaze metode. VBScript može napraviti instanciju objekta na rezultatu te metode. Ako jedna od metoda u class formi u okruženju VB kreira prazni objekt kolekciju, onda će i ASP vezan na tu metodu kreirati instanciju tog praznog objekta kolekcije. Tada će ASP dodati članove toj instanciji objekta kolekcije. Članovi će biti u našem slučaju `dio@e92.hr` i `dio`, e-mail i zaporka, podaci koje je unio posjetitelj na formi. Sada napunjena kolekcija s podacima iz forme, sukladna VB okruženju, odnosi podatke u komponentu `eKarteloASPKupci`, konkretnije class formu `Kupci`, konkretnije metodi `ProvjeriKupca`.

Dakle, nastavimo s našim kôdom. Instanciju roditelja, class forme `NovaKolekcija` imamo. To je `objPomocni`. U class formi `NovaKolekcija` postoji metoda koja se isto zove `NovaKolekcija`. Kreirajmo instanciju u ASP na osnovu te metode...

```
<% set objPodaciKupca = objPomocni.NovaKolekcija() %>
```

Toj instanciji objekta kolekcije, praznoj kolekciji, koja je u VBScriptu sada dohvatljiva pod imenom `objPodaciKupca`, VBScript će pomoću `Add` metode, koju podržava kolekcija napravljena u Visual Basic-u (a ne i kolekcija napravljena u VBScriptu), dodijeliti sljedeće članove...

```
<% objPodaciKupca.Add Session("email"),"email" %>
```

```
<% objPodaciKupca.Add Session("zaporka"), "zaporka" %>
```

"Item" (član) kolekcije je vrijednost dio@e92.hr a "key" Item-a je "email". Drugi je Item=dio, a key=zaporka. Prisjetimo se jedne od prednosti kolekcija ispred array-a. U array-u su članovi dohvatljivi samo preko indexa, a u kolekcijama preko jedinstvene oznake svakog člana kolekcije, "key-a". Kao u bazama podataka, gdje je svaki redak u tablici, slog, dohvatljiv preko svog jedinstvenog naziva u polju koje smo definirali kao neponovljivo (polje također dobije ikonu ključa, key-a).

Objekt kolekcija objPodaciKupca ima dva člana čiji su key-evi *email* i *zaporka*. Asp kôd, konkretno VBScript kôd, spreman je konačno kontaktirati prvu komponentu za vezu s bazom eKarteloASPKupci.

Čeka nas isti proces kao kada smo se spajali na komponentu eKarteloASPPomocna.

Slijedi najprije kreiranje instancije objekta veze s komponentom i konkretno class formom Kupci (objektom Kupci)...

```
<% set objKupci = Server.CreateObject("eKarteloASPKupci.Kupci") %>
```

...pa onda kreiranje instancije objekta objOdgovor koji će sadržavati povratne podatke metode ProvjeriKupca. Metoda ProvjeriKupca je jedna od tri metode u objektu (class formi) Kupci...

```
<% set objOdgovor = objKupci.ProvjeriKupca(objPodaciKupca) %>
```

Dogodilo se u istoj naredbi: inicijalizacija instancije objekta objOdgovor i stvarno oživljavanje tog objekta s podacima koje će mu vratiti metoda ProvjeriKupca. Podaci *email* i *zaporka* se prenose parametarski, a parametar je objekt *objPodaciKupca* naveden u zagradi metode koja se poziva, metode ProvjeriKupca. Metoda `ProvjeriKupca` će prihvatiti vrijednosti parametra ByValue (po vrijednosti), a ne ByRef (po referenciji). Ona će obaviti svoj dio posla s tim vrijednostima i rezultat vratiti ASP kôdu.

Kada kreirate komponentu sasvim je svejedno koja ćete imena dati varijablama s obzirom na imena koja ste dali varijablama, objektima u ASP kôdu. Asp stranica i komponenta dvije su potpuno izolirane tvrđave koje izvanredno komuniciraju. Asp kôd vidi "vrata" na tvrđavi komponenti i zna što može kroz ta vrata ubaciti u tvrđavu. Isto tako jako dobro zna što mu sve može ta tvrđava vratiti. I to je upravo nastavak ASP kôda na ASP stranici ekartelo.asp.

Ako je sve proteklo u redu, nakon ovog retka...

```
<% set objOdgovor = objKupci.ProvjeriKupca(objPodaciKupca) %>
```

...asp kôd zna postoji li kupac već u bazi ili ne. U nastavku su mu potrebni samo ti vraćeni podaci koji se nalaze u objektu objOdgovor. Ostali objekti, koji su omogućili dohvat podataka iz baze, više nisu potrebni, i ako ih eksplicitno ne razorimo, oni će živjeti i zauzimati resurse, vezati na sebe COM komponente sve do kraja života ove ASP stranice, a to je do kraja procesuiranja ove ASP stranice. Zato slijedi eksplicitno razaranje nepotrebnih objekata, odmah sada, kada više ne trebaju:

```
<% set objPodaciKupca = nothing %>
```

```
<% set objKupci = nothing %>
```

```
<% set objPomocni = nothing %>
```

Ovoga trenutka, znači, ASP stranica je prestala vezati na sebe sve komponente. Držala ih je samo onoliko koliko joj je bilo potrebno da dobije rezultat metode ProvjeriKupca. Kako smo morali kreirati mnogo objekata, zato ponovimo:

- objPomocni uspostavlja vezu s komponentom eKarteloASPPomocna,
- objPodaciKupca je instancija objekta čiji je sadržaj rezultat metode NovaKolekcija, djeteta te gornje veze,
- objKupci je instancija objekta veze s komponentom eKarteloASPkupci, class formom Kupci,
- objOdgovor je instancija objekta metode ProvjeriKupca (dijete gornje veze).

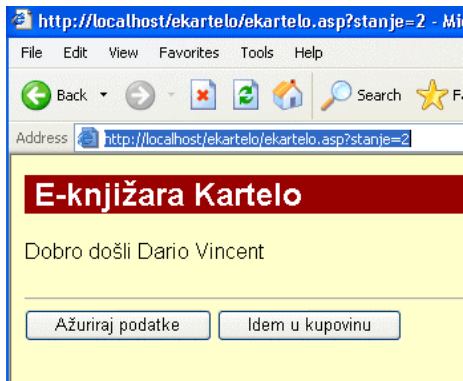
Objekata je puno jer su potrebna dva objekta kako bi se došlo do cilja u jednoj komponenti. Objekt roditelj (veza s komponentom, točnije s class formom u toj komponenti) i objekt dijete (objekt na osnovu metode u class formi).

Od četiri objekta, jedan još živi u ovom trenutku, a to je objOdgovor. Što je sve njemu vratila metoda ProvjeriKupca? objOdgovor je objekt kolekcija i ako pogledamo daljnji tijek ASP skripta, vidjet ćemo što je to metoda vratila.

Možemo napustiti ASP stranicu i pratiti dijagram kretanja podataka i procesuiranja u metodi ProvjeriKupca.

Ipak, postavimo sebi sljedeće pitanje. Što želimo dobiti od metode ProvjeriKupca? Koje podatke?

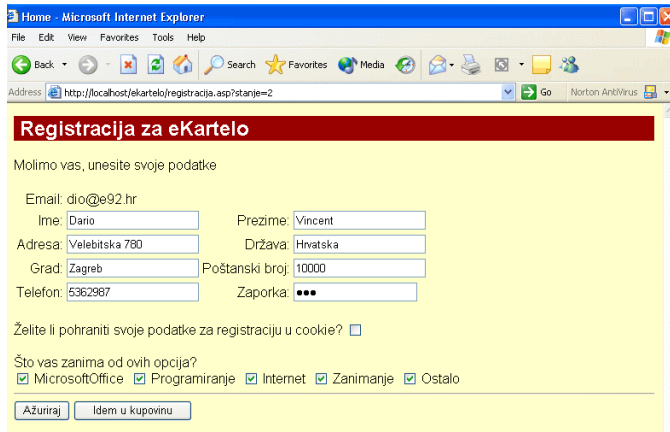
Pogledajmo još jednom formu koju ćemo dobiti kada metoda ProvjeriKupca ispravno odradi svoj posao:



Web stranica nas je dočekala s pozdravom, i to nam se sviđa. Metoda ProvjeriKupca nas je pronašla i vratila naše podatke. Koje podatke? Da li samo ime i prezime? Vratila je sve podatke iz našeg retka, jer će ti podaci biti potrebni ako pritisnemo gumb 'Ažuriraj podatke'.

Moglo je i drukčije. Ne donijeti sve podatke, nego ići po njih ako posjetitelj pritisne gumb Ažuriraj podatke. A to bi značilo još jednom kontaktirati komponente i bazu, što su dodatna opterećenja na resurse web aplikacije. Svi su podaci doneseni i nalaze se u objektu session, spremni - ako pritisnemo gumb Ažuriraj podatke - prijeći u formu web stranice koja će biti odaslana browser-u.

Pogledajmo tu formu. Pritisnimo gumb Ažuriraj podatke i evo ispunjene forme s našim podacima. Možemo promijeniti što želimo.



U Address polju browser-a je HTTP zahtjev
<http://localhost/ekartelo/registracija.asp?stanje=2>

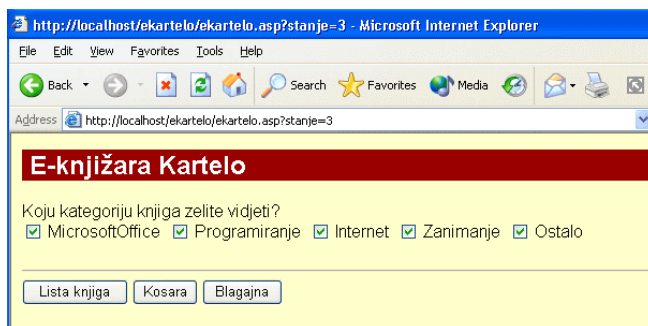
Zahtjev je kreiran VBScriptom za programiranje klijent strane...

```
onclick="window.document.location.href='registracija.asp?stanje=2' "
```

Prebačeni smo na novu ASP stranicu, registracija.asp. Nije kontaktirao bazu. Znači, podaci su u objektu Session. Dokaz je i to što su ti podaci dostupni i na novoj ASP stranici, a to mogu biti samo ako dolaze iz objekta Session. Isto tako, ako odaberemo gumb "Idem u kupovinu", aktivirat će se sljedeće skriptiranje klijent strane...

```
<input type="button" value="Idem u kupovinu"
onclick="window.document.location.href='ekartelo.asp?stanje=3' ">
```

Ostajemo na istoj ASP stranici, ekartelo.asp, stanje = 3...



Pogledajte View/Source ove web stranice:

```
<html>

<head>

<META HTTP-EQUIV="Pragma" CONTENT="no-cache">

<title></title>

<link rel="stylesheet" type="text/css"
href="glavni.css">

</head>

<body>

<table width=100% cellspacing=0 cellpadding=0 border=0>
<tr class="zaglavljestranice">
<th align=left>&nbsp;E-knji&#382ara Kartelo
</th>
</tr>
</table>

    <form action="ekartelo.asp?stanje=4" method="POST">

    <p>Koju kategoriju knjiga zelite vidjeti?</br>

    <input type="checkbox" checked name =
"MicrosoftOffice" value = "y" >
MicrosoftOffice&nbsp;&nbsp;&nbsp;<input type="checkbox"
checked name = "Programiranje" value = "y" >
Programiranje&nbsp;&nbsp;&nbsp;<input type="checkbox" checked
name = "Internet" value = "y" >
Internet&nbsp;&nbsp;&nbsp;<input type="checkbox" checked
name = "Zanimanje" value = "y" >
Zanimanje&nbsp;&nbsp;&nbsp;<input type="checkbox" checked
name = "Ostalo" value = "y" > Ostalo&nbsp;&nbsp;&nbsp;

    <p>
```

```

<hr>

<input type="submit" name="List" value="Lista knjiga">

<input type="button" value="Kosara"

onclick="window.document.location.href='pogleduKosaru.a
sp' ">

<input type="button" value="Blagajna"

onclick="window.document.location.href='blagajna.asp' ">

</form>

</form>

</body>

</html>

```

Bit će zanimljivo vidjeti skript kôd koji je od ovih podataka ZanimaSeZa iz tablice Kupci:

Kupci : Table			
	Zaporka	KoristiCookie	ZanimaSeZa
▶	dio	N	M,P,I,Z,O
	iva	N	M,P,I,Z,O
	mar	N	P
	valentini	Y	M,P,I,Z,O
	vlatka	N	M P I Z O

napravio smislenu HTML stranicu. Onaj string slova M,P,I,Z.O u polju ZanimaSeZa tablice Kupci oblikovao je uredno složen redak checkbox-a s odgovarajućim nazivima:

Koju kategoriju knjiga zelite vidjeti?
 MicrosoftOffice Programiranje Internet Zanimanje Ostalo

Nastavimo s analizom kôda na ASP stranici ekartelo.asp

Posljednji kôd koji smo analizirali na stranici ekartelo.asp bio je

```
<% set objPodaciKupca = nothing %>
```



```
<% set objKupci = nothing %>
<% set objPomocni = nothing %>
```

Slijedi kôd:

```
<% if objOdgovor("greska") then %>
<!--#include file=error.inc-->
<% else %>
<% if objOdgovor("postoji") and objOdgovor("zaporkatocna") then %>
<% set objPodaciKupca = objOdgovor("podacikupca") %>
```

Iz gornjega kôda vidimo da kolekcija objOdgovor, koji je kreiran na osnovu metode ProvjeriKupca, ima sljedeća četiri Item-a i njihova sljedeća četiri Key-a:

```
objOdgovor("greska").....False ili True
objOdgovor("postoji").....False ili True
objOdgovor("zaporkatocna")...False ili True
objOdgovor("podacikupca")....kolekcija - svi podaci kupca iz
tablice Kupci.
```

"Greska" može biti True ili False; vidjet ćemo u komponentama gdje je kreirana. Ako je True, poziva se error.inc koji se uklapa u ASP svake ASP stranice, pa je izdvojen kao ponavljajuća cjelina kôda u error.inc.

```
<% if objOdgovor("greska") then %>
<!--#include file=error.inc-->
```

Ako je greska = False, što znači da se nije dogodila, idemo dalje na naredbu else:

```
<% else %>
<% if objOdgovor("postoji") and objOdgovor("zaporkatocna") then %>
```

Ako je kupac dio@e92.hr pronađen u tablici Kupci, u komponentama je kreirana varijabla "postoji" i vrijednost iste je stavljena na True. E-mail posjetitelja je ujedno i jedinstveni ID kupca. Ako je pronađen e-mail, kupac postoji. Sada se provjerava je li točna zaporka. Ako je točna i zaporka, kreira se varijabla "zaporkatocna" i postavlja na vrijednost True. Ako e-maila nema u tablici Kupci, vratit će se...

```
postoji = false
```

...a ako postoji e-mail, a zaporka nije točna, vratit će se...

```
zaporkatocna = false
```

Zato slijedi IF...

```
<% if objOdgovor("postoji") and objOdgovor("zaporkatocna")
```

Ako kupac postoji i (AND) ako je zaporka točna, slijedi procesuiranje tijela strukture IF...End IF. Oba uvjeta u jednom retku moraju biti zadovoljena da bi se izvršile naredbe tijela IF. Dakle, kupac postoji, zaporka je točna, greške nema. Kreiramo objekt kolekciju koji će prihvatiti samo ITEM čiji je KEY "podacikupca". "podacikupca" su kolekcija čiji su ITEM-i podaci iz retka tablice Kupci.

```
    <% set objPodaciKupca = objOdgovor("podacikupca") %>
```

Budući da su nam podaci kupca potrebni i na drugim ASP stranicama, znači potrebni su nam cijelo vrijeme trajanja posjete, podatke kupca prenijet ćemo u objekt Session; nakon toga nam objekt objPodaciKupca više ne treba i eksplicitno ćemo ga odmah razoriti. Također, Session varijablu "posjetitelj" stavljamo na True, jer od sada pa nadalje posjetitelj postoji, pa neka to znaju sve ASP stranice koje budu zahvaćene sesijom (posjetom)....

```
<% Session("posjetitelj") = True %>
```

```
<% Session("email") = objPodaciKupca("email") %>
```

```
<% Session("ime") = objPodaciKupca("ime") %>
```

```
<% Session("prezime") = objPodaciKupca("prezime") %>
```

```
<% Session("adresa") = objPodaciKupca("adresa") %>
```

```
<% Session("grad") = objPodaciKupca("grad") %>
```

```
<% Session("postanskibroj") = objPodaciKupca("postanskibroj") %>
```

```
<% Session("drzava") = objPodaciKupca("drzava") %>
```

```
<% Session("telefon") = objPodaciKupca("telefon") %>
```

```
<% Session("koristicookie") = objPodaciKupca("koristicookie") %>
```

```
<% set Session("zanimaseza") = objPodaciKupca("zanimaseza") %>
```

```
<% set objPodaciKupca = nothing %>
```

Obratite pozornost na ovaj redak...

```
<% set Session("zanimaseza") = objPodaciKupca("zanimaseza") %>
```

Ključna riječ Set ovdje je korištena za kreaciju posebnog objekta Session, koji će imati samo jedan Item, i to onaj čiji je Key "zanimaseza". To je string M,P,I,Z,O iz kojega će trebati kreirati "parse" browser-a:

```
Koju kategoriju knjiga zelite vidjeti?  
 MicrosoftOffice  Programiranje  Internet  Zanimanje  Ostalo
```

Ovdje će biti dosta kôdiranja pa je lakše manipulirati s jednim objektom Session. koji sadrži samo taj string.

Ako iti jedno od "I" strana nije točno u strukturi...

```
<% if objOdgovor("postoji") and objOdgovor("zaporkatocna")
```

tj. vratilo se s vrijednošću FALSE, pogledajte što kaže prva sljedeća naredba ELSE. Broje se pokušaji posjetitelja i nakon tri neuspjela pokušaja...

```
<% else %>
```

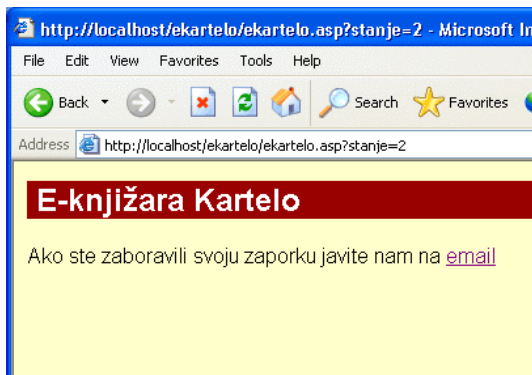
```
<% Session("posjetiteljBroj") = Session("posjetiteljBroj") + 1 %>
```

```
<% if Session("posjetiteljBroj") > 3 then %>
```

...asp šalje u browser sljedeću poruku...

```
<p>Ako ste zaboravili svoju zaporku javite nam na
```

```
<a href="mailto:skola-e-92@st.tel.hr">email</a></p>
```



...u suprotnome šalje ovu poruku...

```
<% else %>

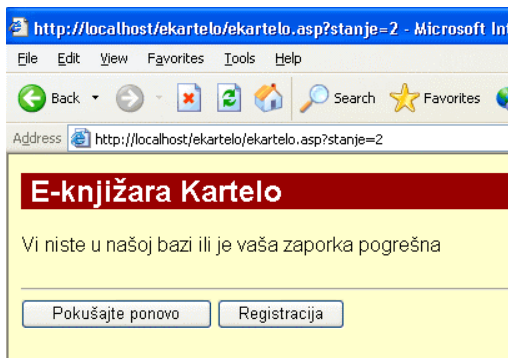
<p>Vi niste u na&#353oj bazi ili je va&#353a zaporka
pogre&#353na</p>

<hr>

<input type="button" value="Poku&#353ajte ponovo"
onclick="window.document.location.href='ekartelo.asp?stanje=1'">

<input type="button" value="Registracija"
onclick="window.document.location.href='registracija.asp?stanje=1'
">

<% end if %>
```



registracija.asp

Kada smo odabrali gumb 'Ažuriraj podatke', VBScript klijent strane (onclick...) nas je usmjerio na stranicu registracija.asp, pa smo uskoro na zaslonu imali sve naše podatke iz tablice Kupci Access baze ekartelo.mdb.

U Address polju browser-a je HTTP zahtjev:
<http://localhost/ekartelo/registracija.asp?stanje=2>

Taj zahtjev nam je vratio ovu web stranicu:


```

        <td align=right>Adresa:</td>
        <td><input type="text" size="20" name="adresa"
value="<% = Session("adresa") %>" ></td>
        <td align=right>Dr&#382ava:</td>
        <td><input type="text" size="20" name="drzava"
value="<% = Session("drzava") %>" ></td>
    </tr>
    <tr>
        <td align=right>Grad:</td>
        <td><input type="text" size="20" name="grad"
value="<% = Session("grad") %>" ></td>
        <td align=right>Po&#353tanski broj:</td>
        <td><input type="text" size="20" name="postanskibroj"
value="<% = Session("postanskibroj") %>" ></td>
    </tr>
    <tr>
        <td align=right>Telefon:</td>
        <td><input type="text" size="20" name="telefon"
value="<% = Session("telefon") %>" ></td>
        <td align=right>Zaporka:</td>
        <td><input type="password" size="20" name="zaporka"
value="<% = Session("zaporka") %>" ></td>
    </tr>
</table>
    <p>&#381elite li pohraniti svoje podatke za registraciju u
cookie?

```

```



```

Ovaj ASP kôd kreira checkbox-ove...

```

<p>&#352to vas zanima od ovih opcija?</br>
<% vPodaci = "" %>
<% for each objZanimaSeZa in Session("zanimaseza") %>
    <% vPodaci = vPodaci & "<input type=" & chr(34) &
"checkbox" & chr(34) %>
    <% if objZanimaSeZa("ukljuceno") then %>
        <% vPodaci = vPodaci & " checked " %>
    <% end if %>
    <% vPodaci = vPodaci & " name = " & chr(34) &
objZanimaSeZa("naziv") & chr(34) %>
    <% vPodaci = vPodaci & " value = ""Y"" > " %>
    <% vPodaci = vPodaci & objZanimaSeZa("naziv") &
"&nbsp;?&nbsp;?" %>
    <% next %>
<% Response.Write vPodaci %>
<hr>
<input type="submit" name="Update" value="A&#382uriraj">
    <input type="button" value="Idem u kupovinu"
onclick="window.document.location.href='OdabirKategorija.asp'">
</form>

```

Lekcija 13

Analiza kôda

komponenti

Da bi naše komponente radile s Transaction Serverom, moramo učiniti sljedeće:

1. Svakoj komponenti ugraditi referenciju za Microsoft Transaction Server Type Library.
2. Dodati sljedeći kôd svakom class modulu:

```
Implements ObjectControl
```

Ova naredba Visual Basic-u znači da će ova class forma podržavati sve metode sučelja ObjectControl. ObjectControl je objekt koji implementira COM specifikacije u okruženje.

3. Implementirati metode ObjectControl_CanBePooled i ObjectControl_Deactivate. Isti je kôd za sve class forme.
4. Implementirati metodu ObjectControl_Activate. Ovaj kôd nije isti za sve class forme. Komponente koje surađuju s ASP imaju unutar ove metode i kreiranje instancije objekta class forme komponente koja surađuje s bazom podataka. Vidi dolje.

Class forma Kupci

Implementacija objekta ObjectControl...

```
Implements ObjectControl
```


Slijede deklaracije nekoliko varijabli životnog vijeka koliko i class forma.

```
Private oSQLKupci As eKarteloSQLKupci.SQLKupci
```

```
Private oObjectContext As ObjectContext
```

Varijabla oSQLKupci će služiti za pristup objektu SQLKupci u našoj komponenti eKarteloSQLKupci.

Varijabla oObjectContext ovaj će modul povezivati s ugrađenim objektom ObjectContext koji podržava transakcije. "context" je naziv za skupinu informacija koje su specifične za svaki konkretni objekt. U tim informacijama su i informacije o okruženju u kojem radi objekt, kao i informacije o transakciji u koju je uključen taj objekt. Te "context-e" za svaki objekt u run-timeu (u radu) stvara ugrađeni objekt ObjectContext. Objekt ObjectContext povezuje ASP okruženje s Transaction Server okruženjem.

Slijedi podrška sljedećim metodama:

```
Private Function ObjectControl_CanBePooled() As Boolean
```

```
ObjectControl_CanBePooled = True
```

```
End Function
```

```
Private Sub ObjectControl_Deactivate()
```

```
Set oObjectContext = Nothing
```

```
Set oSQLKupci = Nothing
```

```
End Sub
```

```
Private Sub ObjectControl_Activate()
```

```
On Error GoTo OCActivateErr
```

```
Set oObjectContext = GetObjectContext()
```

```
Set oSQLKupci =  
oObjectContext.CreateInstance("eKarteloSQLKupci.SQLKupci")
```

```
Exit Sub
```

Ovaj kôd...

```
Set oSQLKupci =  
ObjectContext.CreateInstance("eKarteloSQLKupci.SQLKupci")
```

...postoji samo u komponentama koje pozivaju druge komponente. U našem slučaju komponenta "...ASPkupci" poziva komponentu "...SQLKupci". Zašto kreiranje objekta nismo obavili pomoću ASP objekta Server i metode Server.CreateObject? Zato što tada instancija objekta ne bi bila "pod kišobranom" objekta ObjectContext. Ovako napisana naredba, koja uključuje metodu CreateInstance objekta ObjectContext, implementira podršku objekta ObjectContext novokreiranoj instanciji.

A ovaj kôd će nam donijeti poruku greške...

```
OCActivateErr:  
    Call JaviGresku("OCActivate")  
End Sub
```

Funkcija JaviGresku("OCActivate") se nalazi u zajedničkom modulu svih komponenti: modZajednicki.bas

Gornji kôd je zajednički svim Visual Basic komponentama.

Slijedi kôd prve metode na ovoj class formi.

Deklaracija metode, da bi bila dohvatljiva i vanjskom kôdu...

```
Public Function AzurirajKupca(ByVal vPodaciKupca As Variant) As Variant
```

Obvezno hvatanje greške unutar procedure...

```
On Error GoTo AzurirajKupcaErr
```

Deklaracija varijable koja će prenijeti vanjskom kôdu rezultat metode...

```
Dim vOdgovor As Variant
```

Poziv metode oSQLKupci.AzuriranjeKupca(vPodaciKupca). Ovo je kontakt s komponentom eKarteloSQLKupci koja će kontaktirati bazu.

```
Set vOdgovor = oSQLKupci.AzuriranjeKupca(vPodaciKupca)
```

Ako aplikacija dođe na sljedeći redak...

```
oObjectContext.SetComplete
```

...znači da je prethodna metoda i veza s komponentom i bazom prošla u redu i da varijabla vOdgovor sadrži podatke koje će prenijeti ASP kôdu. Proces transakcije je uspješan i naredbom ...

```
oObjectContext.SetComplete
```

...taj se proces oslobađa. Komponenta oSQLKupci je obavila svoj posao i može napustiti proces transakcije koji je u tijeku, što se s ovom naredbom i postiže.

Ostala nam je varijabla vOdgovor, koja svoj sadržaj predaje vanjskom ASP kôdu ovom naredbom...

```
Set AzurirajKupca = vOdgovor
```

Slijedi standardni kraj ove funkcije...

```
Exit Function
```

```
AzurirajKupcaErr:
```

```
oObjectContext.SetAbort
```

```
Set AzurirajKupca = JaviGresku("AzurirajKupca")
```

```
End Function
```

Gornja konstrukcija kôda vrijedi i za sve ostale metode u svim class formama. Objekt Kupci ima još ove metode:

RegistrirajKupca

ProvjeriKupca

Naš ASP kôd na stranici ekartelo.asp je najprije pozvao metodu ProvjeriKupca; pogledajmo što je ta metoda učinila.

Metoda ProvjeriKupca

Stranica ekartelo.asp ima podatke iz forme, email i zaporku, koje je tipkao posjetitelj, spremne za slanje na provjeru u bazu. Asp stranice mogu i direktno preko ADO kontaktirati bazu. Naša web aplikacija se, međutim, sastoji od ASP stranica, komponenti i baze. Komponente su sloj između ASP stranica i baze. ASP stranice najprije pozivaju komponente koje u sebi kriju poslovnu logiku, i ja sam te komponente označio s "...ASP..." u imenu. Komponente "...ASP..."

kontaktiraju komponente "...SQL..." koje rade rutinske poslove nad bazom, ADD, DELETE, UPDATE, EDIT.

Asp stranica ekartelo.asp je pozvala "...ASP..." komponentu ovako...

```
<% set objKupci = Server.CreateObject("eKarteloASPKupci.Kupci") %>  
<% set objOdgovor = objKupci.ProvjeriKupca(objPodaciKupca) %>
```

Evo što se događalo u komponenti, u metodi ProvjeriKupca:

Metoda počinje s deklaracijom Public, što znači da će biti dohvatljiva kôdu izvan komponente...

```
Public Function ProvjeriKupca(ByVal vPodaciKupca As Variant) As  
Variant
```

ByVal znači da će se podaci prenijeti iz ASP kôda u metodu po vrijednosti, a ne po referenciji. Asp kôd je metodi predao kolekciju sa sljedećim Item-ima:

```
Item=dio@e92.hr; Key=email  
item=dio; Key=zaporka
```

To što su imena varijabli vPodaciKupca ista na ASP strani i sada u metodi ProvjeriKupca(ByVal vPodaciKupca As Variant) nije nikakvo pravilo kojega se moramo držati. Ta imena mogu biti potpuno različita jer sve što se dešava unutar komponente, iza prvog retka metode...

```
Public Function ProvjeriKupca(ByVal vPodaciKupca As Variant) As  
Variant
```

... interna je stvar komponente i njezine metode.

Slijedi premošćivanje greške i deklaracija varijabli...

```
On Error GoTo ProvjeriKupcaErr
```

```
Dim vZaporka2 As Variant
```

```
Dim vOdgovor As Variant
```

```
Dim vPodaciKupca2 As Variant
```

```
Dim vOdgovor2 As Variant
```

U varijablu `vZaporka2` pohranjuje se ono što je posjetitelj tipkao u polje forme zaporka, *dio*.

```
vZaporka2 = vPodaciKupca("zaporka")
```

Razlog je sljedeći. Metoda `ProvjeriKupca` će poslati podatke metodi `OdaberiKupca`, koja će te podatke pokušati pronaći u bazi. Metoda `OdaberiKupca` vraća pronađene podatke, cijeli red iz tablice, nazad. Metoda `ProvjeriKupca` sada mora usporediti podatke iz tablice s podacima koje je dobila iz baze. Provjerava samo zaporku, pa je zato zaporka iz forme morala ovdje biti pohranjena.

Iz kôda koji slijedi vidjet ćemo da metoda `ProvjeriKupca` prosljeđuje e-mail i zaporku prema objektu `SQLKupci` komponente `KarteloSQLKupci`, preciznije, prema metodi `OdaberiKupca`.

Za kontakt s metodom u komponenti potrebna su dva retka naredbe `Set...`

```
Set oSQLKupci =  
ObjectContext.CreateInstance("eKarteloSQLKupci.SQLKupci")
```

```
Set vOdgovor = oSQLKupci.OdaberiKupca(vPodaciKupca)
```

Prvi redak kreira instanciju objekta `SQLKupci`, i to u okruženju objekta `ObjectContext`, pomoću metode `CreateInstance`. Mogla se instancija kreirati i sa `Server.CreateObject`, ali tada ne bi mogla sudjelovati u procesima transakcije.

Drugi redak kreira instanciju objekta metode `OdaberiKupca` i odmah oživljava instanciju s povratnim podacima. Metodi su predani podaci email (*dio@e92.hr*) i zaporka (*dio*).

Ovdje ćemo i mi, zajedno s podacima, napustiti komponentu i prijeći u drugu komponentu. Idemo za podacima u metodu `OdaberiKupca`. Vratit ćemo se na ovo mjesto.

Metoda `eKarteloSQLKupci.SQLKupci.OdaberiKupca`

Metoda počinje deklaracijom `Public`, što je uvjet da bude dohvatljiva izvana.

```
Public Function OdaberiKupca(ByVal vPodaciKupca As Variant) As  
Variant
```

Prihvatit će podatke izvana kao variant. Konkretno, prihvatit će kolekciju koja se sastoji od:

```
Item=dio@e92.hr; Key=email  
Item=dio; Key=zaporka
```

Slijedi premošćivanje greške i deklaracija varijabli:

```
On Error GoTo OdaberiKupcaErr  
  
Dim vOdgovor As New Collection  
  
Dim vPodaciKupca2 As New Collection  
  
Dim vZanimaSeZa As Variant  
  
Dim vSql As Variant
```

Kreira se varijabla vSql, u koju se pohranjuje SQL upit...

```
Let vSql = "SELECT * FROM KUPCI " & _  
" WHERE email = '" & LCase(vPodaciKupca("email")) & "';"
```

Gornji SQL upit samo poziva na pregled (naredba SELECT) sva polja sloga iz tablice Kupci, gdje vrijedi uvjet E-mail = dio@e92.hr

Funkcija LCase vrši konverziju mogućih velikih znakova u mala.

A kako varijabla vSql zna ime baze podataka ekartelo.mdb, točnije DSN ime te baze i lokaciju na poslužitelju? Ona to ne zna jer u njoj to nije upisano. Za to je zadužen ovaj kôd...

```
Set oDb = oObjectContext.CreateInstance("ADODB.Connection")  
  
oDb.Open vSpojNaBazu  
  
Set oRs = oDb.Execute(vSql)
```

Ponovo je instancija kreirana pomoću metode CreateInstance i tako dobila pasku objektaObjectContext. Ovaj put se radi o instanciji jednog ugrađenog ASP ADO objekta. Objekta connection.

Objekt veze se otvara pomoću metode Open i podataka u konstanti vSpojNaBazu.

Pogledajte gdje smo definirali tu konstantu. U modul modZajednicki:

```
Public Const vSpojNaBazu As Variant = "DSN=EKARTELO;UID=sa;PWD=;"
```

Budući da je instancija objekta Recordset oRs kreirana na početku ove class forme...

```
Private oRs As ADODB.Recordset
```

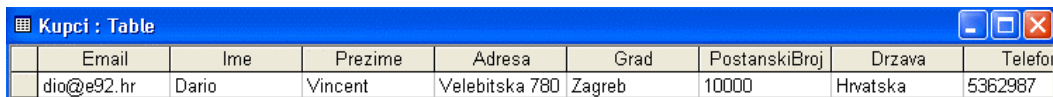
...možemo primijeniti metodu Execute i izvršiti SQL upit nad bazom...

```
Set oRs = oDb.Execute(vSql)
```

Ova naredba provjerava stanje u objektu oRs...

```
If Not oRs.EOF Then
```

...i znači sljedeće. Ako se kursor ne nalazi u EOF, tj. ako se kursor ne nalazi u "praznom retku", znači da je kupac pronađen u bazi i naš oRs se sastoji od jednog sloga. Konkretno, u njemu je ovaj slog ili redak iz tablice Kupci...



Email	Ime	Prezime	Adresa	Grad	PostanskiBroj	Drzava	Telefon
dio@e92.hr	Dario	Vincent	Velebitska 780	Zagreb	10000	Hrvatska	5362987

Objekt Recordset ima kolekciju Fields, u kojoj se ovog trenutka nalaze kopije podataka iz pronađenog retka.

Sada slijedi dosta rada s kolekcijama.

Prije svega, moramo osloboditi Recordset i tako osloboditi vezu s bazom, jer na bazu čekaju i drugi. Prije nego razorimo objekt oRs, sadržaj Recordseta prebacimo u varijablu kolekciju vPodaciKupca2 pomoću metode koju podržava Visual Basic kolekcija, metode Add...

```
vPodaciKupca2.Add Trim(oRs.Fields("Zaporka")), "zaporka"
```

```
vPodaciKupca2.Add CStr(oRs.Fields("email")), "email"
```

```
vPodaciKupca2.Add Trim(oRs.Fields("Ime")), "ime"
```

```
vPodaciKupca2.Add Trim(oRs.Fields("Prezime")), "prezime"
```

```
vPodaciKupca2.Add Trim(oRs.Fields("Adresa")), "adresa"
```

```
vPodaciKupca2.Add Trim(oRs.Fields("Grad")), "grad"
```

```
vPodaciKupca2.Add Trim(oRs.Fields("PostanskiBroj")),  
"postanskibroj"
```

```
vPodaciKupca2.Add Trim(oRs.Fields("Drzava")), "drzava"
```

```
vPodaciKupca2.Add Trim(oRs.Fields("Telefon")), "telefon"
```

```
vPodaciKupca2.Add Trim(oRs.Fields("KoristiCookie")),  
"koristicookie"
```

TRIM je funkcija nad stringom koja uklanja razmaknice. CSTR je funkcija koja vrši konverziju podataka u string. Ranije smo imali funkciju koja Lstr

```
Trace ("Izgradnja zanima se za:")
```

Subprocedura TRACE služi za *debugiranje*. Kako? Trace daje nalog ispisivanja podataka u tekstualni dokument ekartelo.txt, čiji je put c\ekartelo.txt.

Subprocedura Trace se nalazi u modulu modZajednicki i aktivira se svaki put kada je pozvana komponenta. Trace nam koristi u vremenu razvoja aplikacije. Kasnije Trace možemo obrisati ili samo deaktivirati stavljanjem apostrofa ispred.

```
Public Sub Trace(vPoruka As Variant)  
  
    Open "c:\ekartelo.txt" For Append Shared As #1  
  
    Print #1, Time & " " & vPoruka  
  
    Close #1  
  
End Sub
```

Naredba Trace poziva gornju subproceduru.

Na kraju je kreiran poseban objekt kolekcija od podataka u polju ZanimaSeZa...

```
Set vZanimaSeZa =  
IzgradiZanimaSeZa(Trim(oRs.Fields("ZanimaSeZa")))
```

```
vPodaciKupca2.Add vZanimaSeZa, "zanimaseza"
```

...i dodan je kolekciji vPodaciKupca2. Razlog za to je dosta manipulacije koja slijedi oko stringa M,P,I,Z,O u polju ZanimaSeZa tablice Kupci.

Ovog trenutka kreirana je kolekcija vPodaciKupca2 čiji su Item-i svi podaci iz retka tablice, a Key-i su dati po imenima polja, sve malim slovima. Svi su Item-i varijable stringova, samo je polje ZanimaSeZa pridruženo kao kolekcija. Znamo otprije da objekti kolekcije mogu u sebi sadržavati različite tipove podataka, pa tako i druge kolekcije.

Kupac je pronađen, redak iz tablice je pohranjen u kolekciju vPodaciKupca2. Sada programer gradi svoje varijable potrebne za obradu cjelovitog scenarija.

Na osnovu onoga što se dogodilo do sada opisanom kôdu, možemo kolekciji vOdgovor pomoću metode Add dodati Item=True i nazvati ga Key="postoji". U ovoj smo točki sigurni da postoji kupac čiji je E-mail dio@e92.hr. Međutim, mi još ne znamo je li i zaporka točna. Polje E-mail u tablici je deklarirano kao neponovljivo, tj. kao ključ tablice. Zato je nemoguće da ćemo u tablici zateći dva kupca s istim E-mailom. Činjenicu "Kupac postoji" ugradimo u varijablu postoji i pridodajmo tu varijablu kolekciji vOdgovor...

```
vOdgovor.Add True, Key:="postoji"
```

Pomoću metode Add pridodajmo kolekciji vOdgovor kolekciju vPodaciKupca2...

```
vOdgovor.Add vPodaciKupca2, "podacikupca"
```

U slučaju da je kursor bio u EOF u oRs, tj. da kupac nije pronađen u tablici Kupci, slijedilo bi ELSE...

```
Else
```

```
vOdgovor.Add False, Key:="postoji"
```

```
End If
```

...i varijabli *postoji* bi pridodali vrijednost False, i takvu je dodali kolekciji vOdgovor.

Brzo oslobodimo bazu koja nam više ne treba...

```
oDb.Close
```

...i kreirajmo varijablu *greska*. Toj varijabli pridodajmo vrijednost False jer, ako se nalazimo u ovoj točki, to je istina budući da se do sada greška nije pojavila.

Završeno je sudjelovanje ove metode u transakciji.ObjectContext može povući (pooled) njen kôd iz threada i tako osloboditi proces za druge threadove. To se događa u ovoj našoj naredbi...

```
ObjectContext.SetComplete
```

Funkcija (metoda) se izjednačava s kolekcijom vOdgovor, tj. pozivatelju metode OdaberiKupca se vraća kolekcija vOdgovor...

```

Set OdaberiKupca = vOdgovor
I na kraju metode standardno...

Exit Function

OdaberiKupcaErr:

oObjectContext.SetAbort

Set OdaberiKupca = JaviGresku("OdaberiKupca")

End Function

```

Vraćamo se metodi eKarteloASP Kupci.Kupci.ProvjeriKupca tačno tamo gdje smo je napustili

Gdje smo napustili metodu ProvjeriKupca? Napustili smo je u naredbi u kojoj poziva i predaje podatke *email* i *zaporka* metodi OdaberiKupca...

```
Set vOdgovor = oSQLKupci.OdaberiKupca(vPodaciKupca)
```

Najprije analiza mogućih vraćenih podataka

Prije nego što nastavimo, podsjetimo se što je metoda OdaberiKupca vratila. Što je bilo u kolekciji vOdgovor? Bolje je pitanje: koje su sve varijante vOdgovor moguće?

Varijanta 1:

Kupac nije pronađen. Tada je kursor u EOF, pa nakon ove naredbe...

```

If Not oRs.EOF Then

...proces prelazi na ELSE...

Else

vOdgovor.Add False, Key:="postoji"

End If

```

...i ide do...

```
vOdgovor.Add False, Key:="greska"
```

Rezultat navedenoga je sljedeći. Kolekcija vOdgovor se sastoji od ovih Item-a:

Item=False; Key=postoji

Item=False; Key=greska

Varijanta 2:

Kupac je pronađen. Tada je istina da kursor nije u EOF i počinje procesuiranje tijela strukture IF...THEN. Rezultat toga je:

```
vOdgovor.Add True, Key:="postoji"
```

```
vOdgovor.Add vPodaciKupca2, "podacikupca"
```

...čemu se kasnije pridružuje i...

```
vOdgovor.Add False, Key:="greska"
```

Kolekcija vOdgovor sada ima ove Item-e:

Item=True; Key=postoji

Item=False; Key=greska

Item="cijeli redak iz tablice"; Key=podacikupca

Varijanta 3:

Jasno, uvijek postoji i mogućnost greške u procesuiranju kôda metode, kada vrijedi...

```
OdaberiKupcaErr:
```

```
oObjectContext.SetAbort
```

```
Set OdaberiKupca = JaviGresku("OdaberiKupca")
```

```
End Function
```

Najprije će biti metoda isključena iz "contexta" transakcije naredbom...

```
oObjectContext.SetAbort
```

...pa se metoda izjednačava s rezultatom funkcije JaviGresku. Funkciji JaviGresku se prenosi u string "OdaberiKupca"...

Funkcija JaviGresku se odrađuje (nalazi se u modulu modZajednicki):

```
Public Function JaviGresku(ByVal vIme As Variant) As Variant

    Dim vGreskaInfo As Variant

    Dim vOdgovor As Collection

    vGreskaInfo = "Ekartelo greska sustava: " &
Hex(Err.Number) & " " & Err.Source & "->" & vIme & " "
& Err.Description

    Err.Clear

    Set vOdgovor = New Collection

    vOdgovor.Add True, Key:="greska"

    vOdgovor.Add vGreskaInfo, Key:="greskainfo"

    Trace (vGreskaInfo)

    Set JaviGresku = vOdgovor

End Function
```

Tu je opet redak koji poziva proceduru Trace...

```
Trace (vGreskaInfo)
```

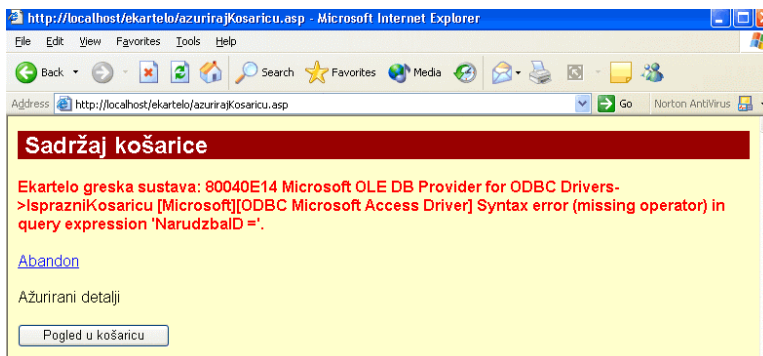
...koja služi samo u razvojnoj fazi aplikacije. Ovaj redak proizvodi onaj ispis greške u C:\ekartelo.txt. Ne zaboravite, prije stavljanja aplikacije u pravi rad, treba staviti apostrof ispred i tako deaktivirati ovaj redak pretvaranjem istoga u komentar.

I na kraju ova naredba...

```
Set JaviGresku = vOdgovor
```

vraća nam poruku...

```
vGreskaInfo = "Ekartelo greska sustava: " &  
Hex(Err.Number) & " " & Err.Source & "->" & vIme & " "  
& Err.Description
```



Ali još smo daleko od browser-a. Analizirali smo tri moguće varijante koje može metoda OdaberiKupca vratiti metodi ProvjeriKupca.

Nastavljamo s kôdom metode Provjeri kupca

Ovdje smo napustili metodu...

```
Set vOdgovor = oSQLKupci.OdaberiKupca(vPodaciKupca)
```

...a već ovdje imali smo podatke sukladne jednoj od gornje tri moguće kombinacije podataka. Od tri moguće varijante podataka, kôd mora otkriti koja je stigla. To čini pomoću višestrukih IF...THEN...ELSE struktura, pa je najbolje cijeli kôd pustiti na papir i olovkom označavati pojedine dijelove te strukture i do kojih rezultata nas one vode.

```
If Not vOdgovor("greska") Then  
    If vOdgovor("postoji") Then  
        Set vPodaciKupca2 = vOdgovor("PodaciKupca")  
        Set vOdgovor2 = New Collection  
        vOdgovor2.Add False, Key:="greska"  
        vOdgovor2.Add True, Key:="postoji"  
        vOdgovor2.Add vPodaciKupca2, Key:="podacikupca"
```

```

        If (Trim(vPodaciKupca2("zaporka")) Like
Trim(vZaporka2)) Then

            vOdgovor2.Add True, Key:="zaporkatocna"

        Else

            vOdgovor2.Add False, Key:="zaporkatocna"

        End If

        Set vOdgovor = vOdgovor2

    End If

End If

```

Varijanta 1:

Zadovoljena su oba kriterija...

```

If Not vOdgovor("greska") Then

    If vOdgovor("postoji") Then

```

...pa se izvršava tijelo unutrašnje IF...THEN strukture...

```

        Set vPodaciKupca2 = vOdgovor("PodaciKupca")

        Set vOdgovor2 = New Collection

        vOdgovor2.Add False, Key:="greska"

        vOdgovor2.Add True, Key:="postoji"

        vOdgovor2.Add vPodaciKupca2, Key:="podacikupca"

```

U gornjem kôdu je puno objekata u igri jer ne želimo vratiti ASP kôdu striktno ono što smo dobili. Naime, mi još ne znamo je li zaporka točna. To ćemo doznati sada, no to je već jedan novi Item, koji do sada nismo imali. On se ovdje *rađa*.

Zato smo najprije u posebnu kolekciju izolirali pristiglu kolekciju "podacikupca"...

```

        Set vPodaciKupca2 = vOdgovor("PodaciKupca")

```

...pa smo kreirali novu praznu kolekciju...

```
Set vOdgovor2 = New Collection
```

...kojoj smo dodali tri Itema i tri Keya...

```
vOdgovor2.Add False, Key:="greska"
```

```
vOdgovor2.Add True, Key:="postoji"
```

```
vOdgovor2.Add vPodaciKupca2, Key:="podacikupca"
```

...provjerili smo dvije zaporke. Onu što ju je tipkao kupac u formi i onu koja je došla iz tablice Kupci...

```
    If (Trim(vPodaciKupca2("zaporka")) Like  
Trim(vZaporka2)) Then
```

...i ako su iste, naša kolekcija je bogatija za još jedan Item...

```
        vOdgovor2.Add True, Key:="zaporkatocna"
```

...a ako nisu iste, naša je kolekcija također bogatija za jedan Item, samo mu je vrijednost False...

```
    Else
```

```
        vOdgovor2.Add False, Key:="zaporkatocna"
```

```
    End If
```

```
    Set vOdgovor = vOdgovor2
```

```
End If
```

```
End If
```

```
ObjectContext.SetComplete
```

```
Set ProvjeriKupca = vOdgovor
```

```
Exit Function
```

I ono što metoda ProvjeriKupca vraća ASP kôdu, u jednoj mogućoj varijanti je..

Item="cijeli redak tablice"; Key=podacikupca

Item=False; Key=greska

Item=True; Key=postoji

Item=True ili False; Key=zaporkatocna

Evo nas konačno nazad u ASP kodu stranice ekartelo.asp

Vratila nam se gore navedena varijanta podataka.

Asp kôd na stranici ekartelo.asp napustili smo ovdje...

```
<% set objOdgovor = objKupci.ProvjeriKupca(objPodaciKupca) %>
```

Nakon toga retka u ASP kôdu smo imali kolekciju koja je imala sljedeće članove:

```
Item="cijeli redak tablice"; Key=podacikupca
```

```
Item=False; Key=greska
```

```
Item=True; Key=postoji
```

```
Item=True ili False; Key=zaporkatocna
```

Tu nam je kolekciju vratila metoda ProvjeriKupca. Sada ASP kôd pomoću višestrukih IF...THEN...ELSE struktura mora otkriti koji su nam podaci stigli i prema tome usmjeriti nastavak procesuiranja.

Ako nam je Key=greska; Item=True...

```
<% if objOdgovor("greska") then %>
```

...tada se procesuirá kôd u datoteci error.inc...

```
<!--#include file=error.inc-->
```

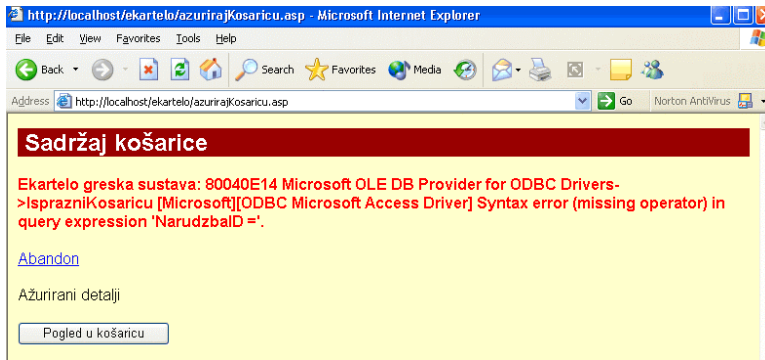
...i taj kôd izgleda ovako...

```
<% Response.Write "<p class=porukaogresci>" &  
objOdgovor("greskainfo") & "</p>" %>
```

```
<% Response.Write "<a href=abort.asp>Abandon</a>" %>
```

```
<% stanje = -1 %>
```

...što znači da će nam objekt Response poslati u browser ovu web stranicu...



Ako je pristigli Item=False; Key=greska, tj. greške nema, put je sljedeći...

```
<% else %>
```

```
<% if objOdgovor("postoji") and objOdgovor("zaporkatocna") then %>
```

...istina je da kupac postoji i da je zaporka točna, pa idemo s kôdom. Najprije kreiramo posebnu kolekciju samo za onaj redak iz tablice Kupci...

```
<% set objPodaciKupca = objOdgovor("podacikupca") %>
```

...pa konstatiramo u varijabli objekta Session...

```
<% Session("posjetitelj") = True %>
```

...prebacimo sve podatke kupca u objekt session kako bi bili na raspolaganju i drugim ASP stranicama koje posjeti kupac. Jer na mnogim stranicama kupac može preko gumba "Ažuriraj podatke" poželjeti vidjeti svoje podatke na zaslonu i promijeniti kategorije koje ga zanimaju, na primjer...

```
<% Session("email") = objPodaciKupca("email") %>
```

```
<% Session("ime") = objPodaciKupca("ime") %>
```

```
<% Session("prezime") = objPodaciKupca("prezime") %>
```

```
<% Session("adresa") = objPodaciKupca("adresa") %>
```

```
<% Session("grad") = objPodaciKupca("grad") %>
```

```
<% Session("postanskibroj") = objPodaciKupca("postanskibroj") %>
```

```
<% Session("drzava") = objPodaciKupca("drzava") %>
```

```
<% Session("telefon") = objPodaciKupca("telefon") %>
```

```
<% Session("koristicookie") = objPodaciKupca("koristicookie") %>
```

```
<% set Session("zanimaseza") = objPodaciKupca("zanimaseza") %>
```

...objekt objPodaciKupca je odigrao svoju ulogu prijenosa podataka u objekt Session, pa ga razaramo...

```
<% set objPodaciKupca = nothing %>
```

Slijedi poznata nam web stranica na našem browser-u, koja potječe od ovoga...

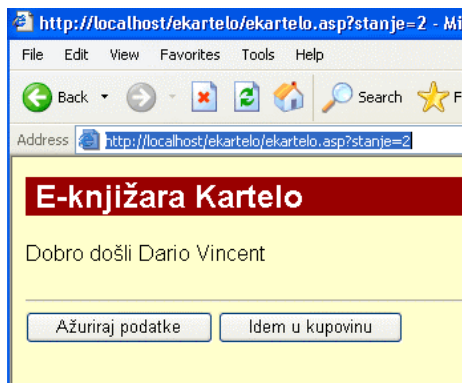
```
<p>Dobro do&#353li
```

```
<% Response.Write Session("ime") & " " & Session("prezime") %>
```

```
</p><hr>
```

```
<input type="button" value="A&#382uriraj podatke"
onclick="window.document.location.href='registracija.asp?stanje=2'
">
```

```
<input type="button" value="Idem u kupovinu"
onclick="window.document.location.href='ekartelo.asp?stanje=3' ">
```



Sažetak analize web aplikacije E-knjižara Kartelo

Na gornjem, živom profesionalnom projektu web aplikacije, kakvih se danas ponajviše vrti na web-u, naučili smo sljedeće.

1. Znanje programiranja u VBScriptu (ili kojem drugom jeziku) i Visual Basic-u (ili kojem drugom jeziku) će nam najviše trebati jer je najviše rada u tome

dijelu. Boolova algebra, AND i OR strukture, IF...THEN...ELSE strukture, manipulacija sa stringovima, mnogo rada s kolekcijama.

2. Nakon što smo danima zauzimali dnevni boravak s papirima s kôdom web aplikacije, zaključili smo da moramo imati cjelovitu sliku projekta prije nego što prijedemo na kodiranje pojedinih sekcija. A sve počinje pričom "što će raditi moja aplikacija korak po korak". Što ćemo uzeti za ID u tablici Kupci. Možda je dovoljno E-mail? Tako imamo i e-mail i ID u jednom.
3. Nakon iskustva s desktop aplikacijama, gdje smo stekli iskustva s Visual Basic-om, a i iskustva sa skriptom na klijent strani, pri pomisli na Web aplikacije najviše nas opterećuje pomisao da ćemo trebati puno znanja o operacijskom sustavu, o IIS, o Transaction Serveru, o Message Queueing Serveru, o procesima i threadovima, o DSN, ODBC itd.

Dogodilo se upravo suprotno. Naučili smo kako je jednostavno registrirati komponente, prijaviti im referencije potrebnih *library* (pa tako i one za Transaction podršku), pohraniti ih kao DLL komponente. Zatim nam je trebalo tek nekoliko redaka kôda u svakom class modulu za implementaciju ObjectControl iObjectContext i naše su komponente radile odlično.

4. Uloga MTS je ogromna za iskoristivost resursa web aplikacije. ObjectContext isključi objekt iz procesa transakcije kad taj objekt završi svoj dio posla. Ne žive sudionici transakcije koliko i sama transakcija. Ne. Žive samo koliko im je potrebno za obavljanje njihovog dijela posla. Jedna transakcija je proces u kojem sukcesivno sudjeluje više objekata. U našem primjeru jedna transakcija započne na asp stranici, onda se poziva objekt u jednoj komponenti, ovaj pozove objekt u drugoj komponenti, druga komponenta kontaktira bazu podataka i rezultat vraća prvoj komponenti, prva komponenta rezultat vraća asp stranici od koje je sve i počelo. To je jedna transakcija. Što radi ContextObjekt? Kad je class forma u komponenti pozvana, najprije se implementira MTS okruženje, konkretno ControlObject i ContextObject, preko ovog kôda:

```
Private oSQLKatalog As eKarteloSQLKatalog.SQLKatalog

Private oObjectContext As ObjectContext

Private Function ObjectControl_CanBePooled() As Boolean

    ObjectControl_CanBePooled = True
```

```

End Function

Private Sub ObjectControl_Deactivate()
    Set oObjectContext = Nothing
    Set oSQLKatalog = Nothing
End Sub

Private Sub ObjectControl_Activate()
    On Error GoTo OCActivateErr
    Set oObjectContext = GetObjectContext()
    Set oSQLKatalog =
oObjectContext.CreateInstance("eKarteloSQLKatalog.SQLKatalog")
    Exit Sub
OCActivateErr:
    Call JaviGresku("OCActivate")
End Sub

```

Tako je konkretni objekt i sve što se u tom objektu događa stavljeno pod kišobran MTS i objekta ControlObject. Kreiran je ObjectContext koji će sad voditi brigu u kojem thread-u će raditi komponenta. Kad objekt obavi svoj posao bez greške, konkretno, kad metoda obavi svoj posao, na kraju svake uspješno odrađene metode dogodit će se ovaj kôd:

```

oObjectContext.SetComplete
Set VratiKatalogListuPoZanimaseza = vOdgovor
Exit Function

```

Nakon što je class forma svoj posao obavila sljedi deaktiviranje MTS ControlObjecta i razaranje objekta ObjectContext u metodi ObjectControl_Deactivate()...

```

Private Sub ObjectControl_Deactivate()
    Set oObjectContext = Nothing
    Set oSQLKatalog = Nothing
End Sub

```

Međutim, ControlObject nije razoren. Samo je deaktiviran. Kao takav on će se puno brže odazvati (ponovo aktivirati) na sljedeći poziv i stvoriti novi objekt ObjectContext samo za taj poziv. Razara se samo konkretni objekt ObjectContext.

To je push-pooled način rada i to omogućava metoda

```
ObjectControl_CanBePooled()...
```

```
Private Function ObjectControl_CanBePooled() As Boolean
```

```
    ObjectControl_CanBePooled = True
```

```
End Function
```

5. Microsoftova filozofija se gradi na ideji: osloboditi programera što je moguće više pisanja kôda i ostaviti mu više vremena za ono što računalo ne može napraviti. Računalo ne može odlučiti umjesto nas da li će web aplikaciju raditi samo s ASP i bazom ili s ASP, COM komponentama i bazom. Ne zna koliko ćemo mi funkcija dati našoj web aplikaciji i koliko će ta aplikacija biti okrenuta kupcu ili *kupcocentrična*. Koliko ergonomična. I ono najvažnije "Koliko će prodati". Onaj famozni One-Click na Amazon.com-u nije izmislilo računalo nego Jeff Bezos. Patentirao je to i sada su samo radi toga svi iza njega jako, jako daleko.

Lekcija 14

Virtualno učilište

Kartelo

U ovoj lekciji je dat cjeloviti asp kôd web aplikacije Virtualno učilište Kartelo. Cijela web aplikacija se sastoji od asp stranica i Access baze. Što se slojeva tiče, i ova baza ima tri sloja: prezentacijski, poslovna logika i baza podataka. Međutim, u ovoj aplikaciji nema jasne granice između prezentacijskog sloja i poslovne logike nego su ta dva sloja fizički pomješana u VBScript kôdu asp stranica.

U daljnjem razvoju aplikacije, ta pomješanost se nastavlja, a gotovo je nemoguće izolirati neke dijelove za uporabu u drugim projektima.

Jedinu prednost uporabe samo ASP tehnologije vidim u kraćim aplikacijama tipa knjige gostiju, ankete, kalendari, funkcije, kvizovi, baner-rotatori, ugrađene ASP komponente, gdje vam nije potrebna pomoć administratora poslužitelja (nema se što registrirati). Ako uz to koristite i Access bazu onda administrator ne mora niti prijavljivati bazu u DSN poslužitelja. Ukoliko očekujete više od deset konkurentnih (istovremenih) posjetitelja, imajte na umu da to Access ne može odraditi. Access nije napravljen za platformu klijent-poslužitelj, već za desktop aplikacije. SQL Server je pravi alat za relacijske baze za tehnologiju klijent-poslužitelj.

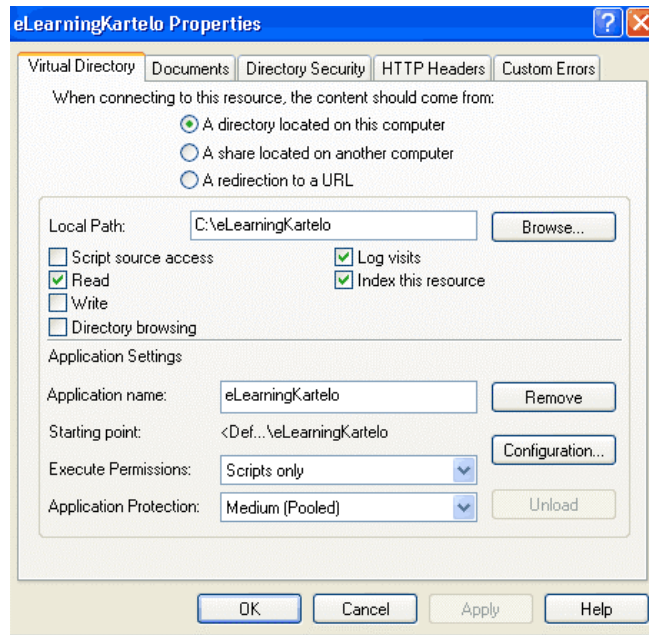
Instalacija aplikacije

1. Skinite programski kôd s Interneta, www.e92.hr/knjigaasp.zip.
2. Pokrenite IIS (Start/Control Panel/Administrative Tools/Internet Information Services).

3. Pritisnite desnu tipku miša na Default Web Site pa u skraćenom izborniku odaberite New/Virtual Directory/Next/U Alias tipkajte **eLearningKartelo**/Next/Browse/Označite mapu eLearningKartelo koju ste skinuli s Interneta u datoteci knjigaasp.zip i pohranili na disk svog računala.



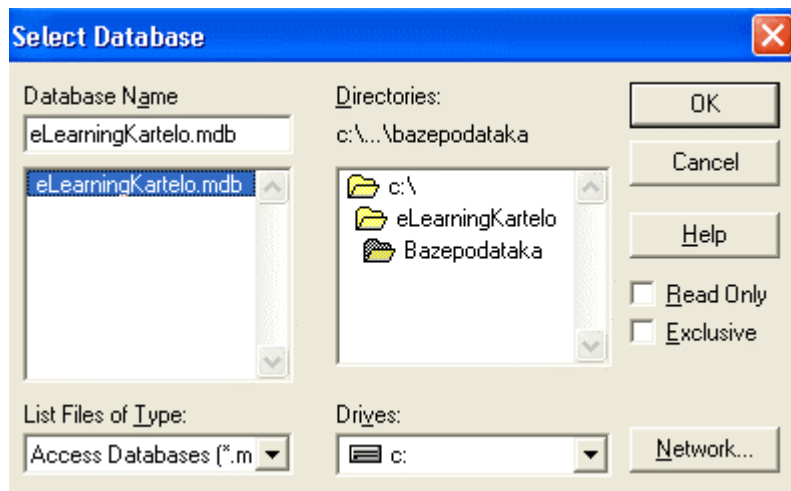
4. OK.
5. Next.
6. Uključite opcije Read i Run Scripts/Next.
7. Finish.
8. Pritisnite desnu tipku miša na virtualni direktorij eLearningKartelo, pa odaberite Properties. Tipkajte u Application name: eLearningKartelo. Podesite ostalo kao na slici.



9. OK.

Prijavite Access bazu u DSN

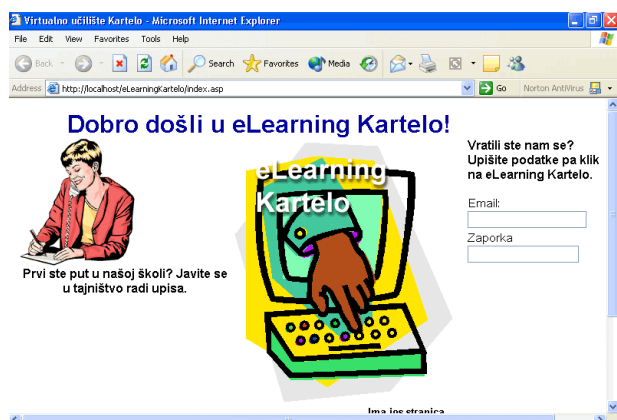
1. Start/Control Panel/Administrative Tools/Data Sources/System DSN/Add/Driver do Microsoft Access/OK/U Data Source Name tipkajte eLearningKartelo/Select/Označite eLearningKartelo.mdb koju ste skinuli s Interneta u knjigaasp.zip/



2. OK. OK.

Pokrenite eLearning Kartelo

1. Otvorite IIS.
2. Otvorite eLearningKartelo.
3. Pritisnite desnu tipku miša na index.asp/Browse.



Uspoređujte View/Source s asp kôdom u nastavku.

index.asp

```
<%@ Language=VBScript %>

<%

if isempty(Request.Form("Email")) then

    'prvi ulaz u eLearning

    Poruka = "Vratili ste nam se? Upi&#353ite podatke pa klik na
eLearning Kartelo."

else

    'provjera postoji li korisnik i da li se dobro logirao

    set conn = server.createobject ("adodb.connection")

    conn.open "eLearningKartelo"

    set rsUcenik = conn.execute ("SELECT UcenikID FROM
eKarteloUcenici where Email = '"_

        & Request.Form("Email") & "' and Zaporka = '" _

        & Request.Form("Zaporka") & "'")

    if rsUcenik.EOF then

        'neuspjelo logiranje

        Poruka = "U&#269enik nije prona&#273en. Ako ste novi
u&#269enik, javite se u tajni&#353tvo."

    else

        'uspjelo logiranje

        Session("UcenikID") = rsUcenik("UcenikID")

        Response.Redirect "./asp/ucenik_izbornik.asp"

    end if

end if

%>

<HTML>
```

```

<HEAD>
<TITLE>Virtualno u&#269ili&#353te Kartelo</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">
<FORM ACTION="./index.asp" METHOD=POST>
  <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=713>
    <TR VALIGN="top" ALIGN="left">
      <TD></TD>
      <TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Dobro do&#353li u eLearning
Kartelo!</B></FONT></TD>
      <TD></TD>
    </TR>
  </TABLE>
<TABLE CELLSPACING=0 CELLPADDING=0 BORDER=0 WIDTH=713>
<TR VALIGN="top" ALIGN="left">
<TD>
  <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=284>
    <TR VALIGN="top" ALIGN="left">
      <TD HEIGHT =131></TD>
      <TD WIDTH=274 COLSPAN=2 ALIGN="left" VALIGN="top"><A
href="./asp/novi_ucenik.asp"><IMG HEIGHT=167 WIDTH=150
src="./sl/l.gif" border=0 alt="Prvi dan" ></A></TD>
    </TR>
    <TR VALIGN="top" ALIGN="left">
      <TD></TD>

```

```

        <TD WIDTH=273><P ALIGN="CENTER"><B><FONT
FACE="Arial,Helvetica">Prvi ste put u na&#353oj &#353koli? Javite
se u tajni&#353tvo radi upisa.</B></FONT></TD>

        <TD></TD>

    </TR>

    <TR VALIGN="top" ALIGN="left">

        <TD COLSPAN=3 HEIGHT =303></TD>

    </TR>

</TABLE>

</TD>

<TD>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=429>

<TR VALIGN="top" ALIGN="left">

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=188>

        <TD HEIGHT =270></TD>

        <TD WIDTH=171><INPUT TYPE="image" NAME="" SRC="./s1/2.gif"
></TD>

    </TR>

</TABLE>

</TD>

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=241>

        <TR VALIGN="top" ALIGN="left">

            <TD></TD>

            <TD WIDTH=180 COLSPAN=3><P><B><FONT FACE="Arial,Helvetica"><%
Response.Write Poruka %></B></FONT></TD>

        </TR>

    </TABLE>

```

```

</TR>

<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=5 HEIGHT =16></TD>
</TR>

<TR VALIGN="top" ALIGN="left">

<TD></TD>

<TD WIDTH=176 COLSPAN=2><P>Email:</TD>

<TD COLSPAN=2></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =22></TD>

<TD WIDTH=160><INPUT TYPE="text" NAME="Email" VALUE="" SIZE=20
MAXLENGTH=50 ></TD>

<TD COLSPAN=3></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD></TD>

<TD WIDTH=176 COLSPAN=2><P>Zaporka</TD>

<TD COLSPAN=2></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =22></TD>

<TD WIDTH=160><INPUT TYPE="password" NAME="Zaporka" VALUE=""
SIZE=20" MAXLENGTH=50 ></TD>

<TD COLSPAN=3></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

```

```

        <TD COLSPAN=5 HEIGHT =116></TD>

    </TR>

</TABLE>

</TD>

</TR>

</TABLE>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=429>

<TR VALIGN="top" ALIGN="left">

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=274>

        <TR VALIGN="top" ALIGN="left">

            <TD></TD>

            <TD WIDTH=146><P ALIGN="RIGHT"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">Ima jos stranica </B></FONT><A
HREF="mailto:skola-e-92@st.tel.hr"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">eKartelo</B></FONT></A><FONT SIZE="-1"
FACE="Arial,Helvetica"><B><br>Online users:
<%=Application("NumberofVisitors")%></B></FONT></TD>

        </TR>

        <TR VALIGN="top" ALIGN="left">

            <TD COLSPAN=2 HEIGHT =73></TD>

        </TR>

    </TABLE>

</TD>

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=155>

        <TR VALIGN="top" ALIGN="left">

```

```

    <TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
    HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
    SRC="./sl/5.gif" BORDER=0 ALT="Slika" ></A></TD>

    <TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =65></TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</FORM></BODY>

</HTML>

```

Global.asa

```

<SCRIPT RUNAT=Server LANGUAGE=VBScript>

Sub Session_OnStart

Application("NumberofVisitors") = Application("NumberofVisitors")

End Sub

Sub Session_OnEnd

Application("NumberofVisitors") = Application("NumberofVisitors")
+ 1

End Sub

```

```
Sub Application_OnStart
Application("NumberOfVisitors") = 0
End Sub
</SCRIPT>
```

izvjesce.asp

```
<%@ Language=VBScript %>
<%
if isempty(Session("UcenikID")) then
    Response.Redirect "../index.asp"
end if

set conn = server.createobject ("adodb.connection")
conn.open "eLearningKartelo"

set rsUcenikZanimanja = conn.execute ("SELECT
eKarteloZanimanja.ZanimanjeNaziv, " _

    & "eKarteloUcenikZanimanja.Stanje,
eKarteloUcenikZanimanja.StanjeDatum " _

    & "FROM eKarteloUcenikZanimanja INNER JOIN eKarteloZanimanja ON
" _

    & "eKarteloUcenikZanimanja.ZanimanjeID =
eKarteloZanimanja.ZanimanjeID " _

    & "WHERE eKarteloUcenikZanimanja.UcenikID = " &
Session("UcenikID"))
%>

<HTML>
<HEAD>
<TITLE>izvjesce</TITLE>
</HEAD>
```



```
<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD></TD>
```

```
<TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Evo va&#353eg
izvje&#353&#263a</B></FONT></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD HEIGHT =109></TD>
```

```
<TD WIDTH=595>
```

```
<TABLE BORDER=1 CELLSPACING=3 CELLPADDING=1 >
```

```
<%
```

```
Do Until rsUcenikZanimanja.EOF
```

```
%>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD WIDTH=49 ALIGN="center" VALIGN="middle"><IMG HEIGHT=31
WIDTH=31 SRC="..s1/3.gif" BORDER=0 ></TD>
```

```
<TD WIDTH=179>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=5 WIDTH=179>
```

```
<TR>
```

```
<TD><P><B><FONT FACE="Arial,Helvetica"><% Response.Write
rsUcenikZanimanja("ZanimanjeNaziv") %></B></FONT></TD>
```

```
</TR>
```

```

</TABLE></TD>

    <TD WIDTH=171>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=5 WIDTH=171>

    <TR>

        <TD><P><B><FONT FACE="Arial,Helvetica"><% Response.Write
rsUcenikZanimanja("Stanje") %></B></FONT></TD>

    </TR>

</TABLE></TD>

    <TD WIDTH=171 HEIGHT=47>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=5 WIDTH=171>

    <TR>

        <TD><P><B><FONT FACE="Arial,Helvetica"><% Response.Write
rsUcenikZanimanja("StanjeDatum") %></B></FONT></TD>

    </TR>

</TABLE></TD>

</TR>

<%
    rsUcenikZanimanja.MoveNext
loop
%>

</TABLE>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>

<TR VALIGN="top" ALIGN="left">

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=173>

        <TR VALIGN="top" ALIGN="left">

            <TD HEIGHT =96></TD>

```

```

    <TD WIDTH=150 ALIGN="left" VALIGN="top"><A
    HREF="../asp/ucenik_izbornik.asp"><IMG HEIGHT=96 WIDTH=150
    SRC="../sl/4.gif" BORDER=0 ALT="Nazad na ulaz" ></A></TD>

    </TR>

    <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=2 HEIGHT =276></TD>

    </TR>

</TABLE>

</TD>

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=380>

    <TR VALIGN="top" ALIGN="left">

    <TD></TD>

    <TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT"><B><FONT SIZE="-1"
    FACE="Arial,Helvetica">Ima jo&#353 </B></FONT><A
    HREF="mailto:skola-e-92@st.tel.hr"><B><FONT SIZE="-1"
    FACE="Arial,Helvetica">eKartelo</B></FONT></A><B><FONT SIZE="-1"
    FACE="Arial,Helvetica"></B></FONT></TD>

    </TR>

    <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=2 HEIGHT =174></TD>

    </TR>

</TABLE>

</TD>

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=147>

    <TR VALIGN="top" ALIGN="left">

    <TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
    HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
    SRC="../sl/5.gif" BORDER=0 ALT="skola-e-92@st.tel.hr" ></A></TD>

```

```

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =166></TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</BODY>

</HTML>

```

Kreniiodustani.asp

```

<%@ Language=VBScript %>

<%

if isempty(Session("UcenikID")) then

    Response.Redirect "../index.asp"

end if

if not isempty(Request.QueryString("ZanimanjeID")) then

    'zelite se ukljuciti u zanimanje

    set conn = server.createobject ("adodb.connection")

    conn.open "eLearningKartelo"

    set rsZanimanjeUTijeku = conn.execute ("SELECT
UcenikZanimanjeID " _

    & " from eKarteloUcenikZanimanja where UcenikID = " _

    & session("UcenikID") & " and ZanimanjeID = " _

```

```

    & Request.QueryString("ZanimanjeID") & " and Stanje =
'UTijeku'")

    if rsZanimanjeUTijeku.EOF then

        set rsZanimanjeUTijeku = conn.execute ("insert Into
eKarteloUcenikZanimanja (UcenikID, ZanimanjeID, " _

            & " Stanje, Stanjedatum) values (" & Session("UcenikID") &
", " & _

            Request.QueryString("ZanimanjeID") & ", 'UTijeku', '" & Date
& "'")")

        end if

        Response.Redirect "../asp/ucenik_izbornik.asp"

elseif not isempty(Request.Form("PrekinutiZanimanje")) then

    'zelite prekinuti zanimanje

    set conn = server.createobject ("adodb.connection")

    conn.open "eLearningKartelo"

    set rsAktivnaZanimanja = conn.Execute ("Delete from
eKarteloUcenikZanimanja " _

        & " where UcenikID = " & session("UcenikID") & " and
ZanimanjeID = " _

        & Request.Form("PrekinutiZanimanje") & " and Stanje =
'UTijeku'")

    Response.Redirect "../asp/ucenik_izbornik.asp"

else

    set conn = server.createobject ("adodb.connection")

    conn.open "eLearningKartelo"

    set rsZanimanja = conn.execute ("SELECT ZanimanjeID,
ZanimanjeNaziv, ZanimanjeOpis FROM eKarteloZanimanja")

    set rsAktivnaZanimanja = conn.Execute ("SELECT
eKarteloUcenikZanimanja.UcenikID, " _

```

```

        & "eKarteloUcenikZanimanja.Stanje,
eKarteloUcenikZanimanja.ZanimanjeID,
eKarteloZanimanja.ZanimanjeNaziv " _

    & "FROM eKarteloUcenikZanimanja INNER JOIN eKarteloZanimanja
ON " _

        & "eKarteloUcenikZanimanja.ZanimanjeID =
eKarteloZanimanja.ZanimanjeID " _

        & "WHERE (eKarteloUcenikZanimanja.UcenikID = " &
Session("UcenikID") _

        & " AND eKarteloUcenikZanimanja.Stanje = 'UTijeku'))"
end if

%>
<HTML>
<HEAD>
<TITLE>Nastaviti ili prekinuti zanimanje</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">
<FORM ACTION=" ../asp/kreniiliodustani.asp" METHOD=POST>
    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
        <TR VALIGN="top" ALIGN="left">
            <TD></TD>
            <TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Odaberite
zanimanje</B></FONT></TD>
        <TD></TD>
    </TR>
</TABLE>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
    <TR VALIGN="top" ALIGN="left">

```

```

<TD COLSPAN=2 HEIGHT =125></TD>

<TD WIDTH=519>

    <TABLE BORDER=1 CELLSPACING=3 CELLPADDING=1 >

<%
do until rsZanimanja.EOF

%>

    <TR VALIGN="top" ALIGN="left">

        <TD WIDTH=57 ALIGN="center" VALIGN="middle"><A
        HREF=" ../asp/kreniiliiodustani.asp?ZanimanjeID=<% Response.Write
        rsZanimanja("ZanimanjeID") %>"><IMG HEIGHT=52 WIDTH=56
        SRC=" ../sl/7.gif" BORDER=0 ALT="Kre&#263em" ></A></TD>

        <TD WIDTH=447 HEIGHT=57><P><B><FONT
        FACE="Arial,Helvetica"><% Response.Write
        rsZanimanja("ZanimanjeNaziv") %> - <% Response.Write
        rsZanimanja("ZanimanjeOpis") %></B></FONT></TD>

    </TR>

<%
rsZanimanja.MoveNext

loop

%>

    </TABLE></TD>

<TD COLSPAN=2></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=5 HEIGHT =22></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =35></TD>

```

```
<TD WIDTH=574 COLSPAN=3 ALIGN="left" VALIGN="top"><IMG  
HEIGHT=35 WIDTH=574 SRC="../../../sl/11.gif" BORDER=0 ALT="Slika"  
></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD></TD>
```

```
<TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"  
SIZE="+3" FACE="Arial,Helvetica">Ili odustanite od  
zanimanja...</B></FONT></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=276>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD COLSPAN=2></TD>
```

```
<TD WIDTH=125><P>Naziv zanimanja</TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD COLSPAN=3 HEIGHT =57></TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD HEIGHT =96></TD>
```



```

    <TD WIDTH=251 COLSPAN=2 ALIGN="left" VALIGN="top"><A
    HREF=" ../asp/ucenik_izbornik.asp"><IMG HEIGHT=96 WIDTH=150
    SRC=" ../sl/4.gif" BORDER=0 ALT="Return to Homeroom" ></A></TD>

    </TR>

    <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=3 HEIGHT =179></TD>

    </TR>

</TABLE>

</TD>

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=279>

    <TR VALIGN="top" ALIGN="left">

    <TD HEIGHT =24></TD>

    <TD WIDTH=130 COLSPAN=2><SELECT NAME="PrekinutiZanimanje" >

    <%

    Do Until rsAktivnaZanimanja.EOF

    %>

    <OPTION VALUE="<% Response.Write rsAktivnaZanimanja("ZanimanjeID")
    %>"><% Response.Write rsAktivnaZanimanja("ZanimanjeNaziv")
    %></OPTION>

    <%

    rsAktivnaZanimanja.MoveNext

    loop

    %>

    </SELECT></TD>

    <TD></TD>

    <TD WIDTH=54><INPUT TYPE="submit" NAME="" VALUE="Odustajem"
    ></TD>

```

```

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=6 HEIGHT =217></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2></TD>

<TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">Ima jo&#353 </B></FONT><A
HREF="mailto:skola-e-92@st.tel.hr"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">eKartelo</B></FONT></A><B><FONT SIZE="-1"
FACE="Arial,Helvetica"></B></FONT></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=6 HEIGHT =77></TD>

</TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=145>

<TR VALIGN="top" ALIGN="left">

<TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
SRC="..s1/5.gif" BORDER=0 ALT="skola-e-92@st.tel.hr" ></A></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =69></TD>

</TR>

```

```
</TABLE>
</TD>
</TR>
</TABLE>
</FORM></BODY>
</HTML>
```

novi_ucenik.asp

```
<%@ Language=VBScript %>
<%
if isempty(Request.Form("Submit")) then
    'Prvi ste put ovdje
    Poruka = "Potrebno je ispuniti sva polja."
else
    if isempty(Request.Form("Ime")) or Request.Form("Ime") = ""
then
        Poruka = "Niste unijeli ime."
    elseif isempty(Request.Form("Email")) or Request.Form("Email")
= "" then
        Poruka = "Niste unijeli email."
    elseif isempty(Request.Form("Zaporka")) or
Request.Form("Zaporka") = "" then
        Poruka = "Niste unijeli zaporku."
    elseif isempty(Request.Form("PonoviZaporku")) or
Request.Form("PonoviZaporku") = "" then
        Poruka = "Niste povo unijeli zaporku."
    elseif Request.Form("Zaporka") <> Request.Form("PonoviZaporku")
then
```

```

    Poruka = "Nisu isti unosi zaporke!"

else

    set conn = server.createobject ("adodb.connection")

    conn.open "eLearningKartelo", "sa", ""

    set rsUcenik = conn.execute ("SELECT UcenikID FROM
eKarteloUcenici where Email = '"_
        & Request.Form("Email") & "'")

    if rsUcenik.EOF then

        'tog ucenika nema u bazi pa se moze prijaviti novi

        set rsUcenik = conn.execute ("insert Into eKarteloUcenici
(Ime, Email," _
            & " Zaporka) values (' & Request.Form("Ime") & "', '" &
            _
            Request.Form("Email") & "', '" & Request.Form("Zaporka")
& "')")

        set rsUcenik = conn.execute ("SELECT UcenikID FROM
eKarteloUcenici where Email = '"_
            & Request.Form("Email") & "' and Zaporka = '" _
            & Request.Form("Zaporka") & "'")

        Session("UcenikID") = rsUcenik("UcenikID")

        Response.Redirect "../asp/ucenik_izbornik.asp"

    else

        'login je zauzet

        Poruka = "Email koji ste upisali je ve&#263 u uporabi."

    end if

end if

end if

%>

```

```

<HTML>

<HEAD>

<TITLE>Novi u&#269;enik</TITLE>

</HEAD>

<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">

<FORM ACTION="../asp/novi_ucenik.asp" METHOD=POST>

  <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>

    <TR VALIGN="top" ALIGN="left">

      <TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Recite nam tko ste vi pa
pritisnite Dobro. <% Response.Write Poruka %></B></FONT></TD>

    </TR>

  </TABLE>

<TABLE CELLSPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>

<TR VALIGN="top" ALIGN="left">

<TD>

  <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=170>

    <TR VALIGN="top" ALIGN="left">

      <TD HEIGHT =99></TD>

      <TD WIDTH=99 ALIGN="left" VALIGN="top"><IMG HEIGHT=99 WIDTH=99
SRC="../sl/6.gif" BORDER=0 ALT="Slika studenta :)" ></TD>

    </TR>

    <TR VALIGN="top" ALIGN="left">

      <TD COLSPAN=2 HEIGHT =383></TD>

    </TR>

  </TABLE>

</TD>

```

```

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=151>

  <TR VALIGN="top" ALIGN="left">

    <TD></TD>

    <TD WIDTH=125 COLSPAN=2><P>Ime</TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=3 HEIGHT =6></TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD></TD>

    <TD WIDTH=125 COLSPAN=2><P>Email</TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=3 HEIGHT =6></TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD></TD>

    <TD WIDTH=125 COLSPAN=2><P>Zaporka</TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=3 HEIGHT =6></TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD></TD>

    <TD WIDTH=125 COLSPAN=2><P>Ponovo zaporka</TD>

```

```

</TR>

<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=3 HEIGHT =6></TD>
</TR>

<TR VALIGN="top" ALIGN="left">
<TD HEIGHT =24></TD>

<TD WIDTH=36><INPUT TYPE="submit" NAME="Submit" VALUE="Dobro"
></TD>

<TD></TD>
</TR>

<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=3 HEIGHT =358></TD>
</TR>
</TABLE>
</TD>
<TD>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=229>
<TR VALIGN="top" ALIGN="left">
<TD HEIGHT =22></TD>
<TD WIDTH=160 COLSPAN=2><INPUT TYPE="text" NAME="Ime" VALUE=""
SIZE=20 MAXLENGTH=50 ></TD>
<TD></TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=4 HEIGHT =3></TD>
</TR>
<TR VALIGN="top" ALIGN="left">

```

```

<TD HEIGHT =22></TD>

<TD WIDTH=160 COLSPAN=2><INPUT TYPE="text" NAME="Email"
VALUE="" SIZE=20 MAXLENGTH=50 ></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=4 HEIGHT =3></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =22></TD>

<TD WIDTH=160 COLSPAN=2><INPUT TYPE="password" NAME="Zaporka"
VALUE="" SIZE=20 MAXLENGTH=50 ></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=4 HEIGHT =3></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =22></TD>

<TD WIDTH=160 COLSPAN=2><INPUT TYPE="password"
NAME="PonoviZaporku" VALUE="" SIZE=20 MAXLENGTH=50 ></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=4 HEIGHT =165></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

```



```

<TD COLSPAN=2></TD>

  <TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">Ima jos </B></FONT><A HREF="mailto:skola-e-
92@st.tel.hr"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">eKartelo</B></FONT></A><B><FONT SIZE="-1"
FACE="Arial,Helvetica"></B></FONT></TD>

  </TR>

</TABLE>

</TD>

<TD>

  <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=150>

    <TR VALIGN="top" ALIGN="left">

      <TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
      HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
      SRC="../../../s1/5.gif" BORDER=0 ALT="skola-e-92@st.tel.hr" ></A></TD>

      <TD></TD>

    </TR>

  </TABLE>

</TABLE>

</FORM></BODY>

</HTML>

```

promjena_podataka.asp

```

<%@ Language=VBScript %>

<%

if isempty(Session("UcenikID")) then

  Response.Redirect "../index.asp"

end if

```

```

if isempty(Request.Form("Submit")) then

    'prvi put u ucilistu

    set conn = server.createobject ("adodb.connection")

    conn.open "eLearningKartelo"

    set rsUcenik = conn.execute ("SELECT Ime, Email, Zaporka FROM
eKarteloUcenici where UcenikID = " & Session("UcenikID"))

    Poruka = "Promjenite svoje podatke i pritisnite Dobro."

    Ime = rsUcenik("Ime")

    Email = rsUcenik("Email")

    Zaporka = rsUcenik("Zaporka")

elseif isempty(Request.Form("Ime")) or Request.Form("Ime") = ""
then

    Poruka = "Niste unijeli ime."

    Ime = Request.Form("Ime")

    Email = Request.Form("Email")

    Zaporka = Request.Form("Zaporka")

elseif isempty(Request.Form("Email")) or Request.Form("Email") =
"" then

    Poruka = "Niste unijeli email."

    Ime = Request.Form("Ime")

    Email = Request.Form("Email")

    Zaporka = Request.Form("Zaporka")

elseif isempty(Request.Form("Zaporka")) or Request.Form("Zaporka")
= "" then

    Poruka = "Niste unijeli zaporku."

    Ime = Request.Form("Ime")

    Email = Request.Form("Email")

    Zaporka = Request.Form("Zaporka")

```

```

elseif isempty(Request.Form("PonovoZaporka")) or
Request.Form("PonovoZaporka") = "" then

    Poruka = "Niste ponovili zaporku."

    Ime = Request.Form("Ime")

    Email = Request.Form("Email")

    Zaporka = Request.Form("Zaporka")

elseif Request.Form("Zaporka") <> Request.Form("PonovoZaporka")
then
    Poruka = "Zaporke se ne sla&#382u!"

    Ime = Request.Form("Ime")

    Email = Request.Form("Email")

    Zaporka = Request.Form("Zaporka")

else

    set conn = server.createobject ("adodb.connection")

    conn.open "eLearningKartelo"

    set rsUcenik = conn.execute ("update eKarteloUcenici set Ime =
'" _
    & Request.Form("Ime") & "', Email = '" _
    & Request.Form("Email") & "', Zaporka = '" _
    & Request.Form("Zaporka") & "' where UcenikID = " _
    & Session("UcenikID"))

    Response.Redirect "../asp/ucenik_izbornik.asp"

end if

%>

<HTML>

<HEAD>

```

```
<TITLE>Promjena podataka</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"  
TEXT="#000000">
```

```
<FORM ACTION="" METHOD=POST>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD></TD>
```

```
<TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"  
SIZE="+3" FACE="Arial,Helvetica"><% Response.Write Poruka  
%></B></FONT></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD>
```

```
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=325>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=174>
```

```

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =99></TD>

<TD WIDTH=99 ALIGN="left" VALIGN="top"><IMG HEIGHT=120
WIDTH=200 SRC="../../sl/6.gif" BORDER=0 ALT="Slika u&#269enika :)"
></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =25></TD>

</TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=151>

<TR VALIGN="top" ALIGN="left">

<TD></TD>

<TD WIDTH=125 COLSPAN=2><P>Ime</TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=3 HEIGHT =6></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD></TD>

<TD WIDTH=125 COLSPAN=2><P>Email</TD>

</TR>

```

```

<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=3 HEIGHT =6></TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD></TD>
<TD WIDTH=125 COLSPAN=2><P>Zaporka</TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=3 HEIGHT =6></TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD></TD>
<TD WIDTH=125 COLSPAN=2><P>Ponovo zaporka</TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=3 HEIGHT =6></TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD HEIGHT =24></TD>
<TD WIDTH=36><INPUT TYPE="submit" NAME="Submit" VALUE="Dobro"
></TD>
<TD></TD>
</TR>
</TABLE>

```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=325>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD HEIGHT =96></TD>
```

```
<TD WIDTH=150 ALIGN="left" VALIGN="top"><A  
HREF=" ../asp/ucenik_izbornik.asp"><IMG HEIGHT=96 WIDTH=150  
SRC=" ../sl/4.gif" BORDER=0 ALT="Povratak na ulaz" ></A></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD COLSPAN=3 HEIGHT =229></TD>
```

```
</TR>
```

```
</TABLE>
```

```
</TD>
```

```
<TD>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=229>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD HEIGHT =22></TD>
```

```
<TD WIDTH=160 COLSPAN=2><INPUT TYPE="text" NAME="Ime" VALUE="<%  
Response.Write Ime %>" SIZE=20 MAXLENGTH=50 ></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```

<TD COLSPAN=4 HEIGHT =3></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =22></TD>

<TD WIDTH=160 COLSPAN=2><INPUT TYPE="text" NAME="Email"
VALUE="<% Response.Write Email %>" SIZE=20 MAXLENGTH=50 ></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=4 HEIGHT =3></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =22></TD>

<TD WIDTH=160 COLSPAN=2><INPUT TYPE="password" NAME="Zaporka"
VALUE="<% Response.Write Zaporka %>" SIZE=20 MAXLENGTH=50 ></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=4 HEIGHT =3></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =22></TD>

<TD WIDTH=160 COLSPAN=2><INPUT TYPE="password"
NAME="PonovoZaporka" VALUE="<% Response.Write Zaporka %>" SIZE=20"
MAXLENGTH=50 ></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

```



```

<TD COLSPAN=4 HEIGHT =249></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2></TD>

<TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">Ima jo&#353 </B></FONT><A
HREF="mailto:skola-e-92@st.tel.hr"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">eKartelo</B></FONT></A><B><FONT SIZE="-1"
FACE="Arial,Helvetica"></B></FONT></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=4 HEIGHT =97></TD>

</TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=146>

<TR VALIGN="top" ALIGN="left">

<TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
SRC=" ../sl/5.gif" BORDER=0 ALT="skola-e-92@st.tel.hr" ></A></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =89></TD>

</TR>

</TABLE>

```

```
</TD>
</TR>
</TABLE>
```

```
</FORM></BODY>
```

```
</HTML>
```

putlekcija.asp

```
<%@ Language=VBScript %>
<%
if isempty(Session("UcenikID")) then
    Response.Redirect "../index.asp"
end if

if isempty(Request.QueryString("LekcijaID")) then
    Response.Redirect "../asp/ucenik_izbornik.asp"
end if

set conn = server.createobject ("adodb.connection")
conn.open "eLearningKartelo"

set rsLekcijaUzeta = conn.Execute("SELECT UcenikLekcijaID " _
    & " from eKarteloUcenikLekcije where LekcijaID = " _
    & Request.QueryString("LekcijaID") & " and UcenikID = " _
    & Session("UcenikID"))

if rsLekcijaUzeta.EOF then

    conn.Execute "insert Into eKarteloUcenikLekcije (LekcijaID,
UcenikID) " _
```

```

        & "values (" & Request.QueryString("LekcijaID") & ", " _
        & Session("UcenikID") & ")"
end if

set rsLekcijaURL = conn.Execute("select LekcijaURL " _
    & "from eKarteloLekcije where LekcijaID = " &
Request.QueryString("LekcijaID"))

Response.Redirect rsLekcijaURL("LekcijaURL")

%>

```

razred.asp

```

<%@ Language=VBScript %>

<%

if isempty(Session("UcenikID")) then

    Response.Redirect "../index.asp"

end if

if isempty(Request.QueryString("ZanimanjeID")) then

    Response.Redirect "../asp/ucenik_izbornik.asp"

end if

Response.Expires = 0

set conn = server.createobject ("adodb.connection")

conn.open "eLearningKartelo"

set rsLekcije = conn.Execute ("SELECT LekcijaID, LekcijaNaziv, " _
    & "LekcijaOpis FROM eKarteloLekcije " _
    & "WHERE ZanimanjeID = " & Request.QueryString("ZanimanjeID"))

set rsZadaci = conn.Execute ("SELECT ZadatakID, ZadatakNaziv, " _

```

```

    & "ZadatakOpis, Zadatak FROM eKarteloZadaci " _
    & "WHERE ZanimanjeID = " & Request.QueryString("ZanimanjeID"))
set rsTestovi = conn.Execute ("SELECT TestID, TestNaziv, " _
    & "TestOpis FROM eKarteloTestovi " _
    & "WHERE ZanimanjeID = " & Request.QueryString("ZanimanjeID"))
set rsZanimanja = conn.Execute("SELECT ZanimanjeNaziv from
eKarteloZanimanja " _
    & " WHERE ZanimanjeID = " & Request.QueryString("ZanimanjeID"))

```

```
%>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Razred</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Razred: <% Response.write
rsZanimanja("ZanimanjeNaziv") %></B></FONT></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE CELLSPACING=0 CELLPADDING=0 BORDER=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=230>
```

```
<TR VALIGN="top" ALIGN="left">
```

```

<TD HEIGHT =88></TD>

<TD WIDTH=180 ALIGN="left" VALIGN="top"><IMG HEIGHT=132
WIDTH=108 SRC="../../sl/10.gif" BORDER=0 ALT="Slika" ></TD>

</TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=470>

<TR VALIGN="top" ALIGN="left">

<TD></TD>

<TD WIDTH=325><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Lekcije</B></FONT></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=3 HEIGHT =51></TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =84></TD>

<TD WIDTH=525>

<TABLE BORDER=1 CELLSPACING=3 CELLPADDING=1 >

```

```

<%
Do until rsLekcije.EOF

PrivLekcija = rsLekcije("LekcijaOpis")

set rsLekcijeUzete = conn.Execute("SELECT UcenikLekcijaID " _
& " from eKarteloUcenikLekcije where LekcijaID = " _
& rsLekcije("LekcijaID") & " and UcenikID = " _
& Session("UcenikID"))

if rsLekcijeUzete.EOF then

IkonaPut = "../sl/3.gif"

else

IkonaPut = "../sl/3check.gif"

end if

%>

<TR VALIGN="top" ALIGN="left">

<TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
HREF="../asp/putlekcija.asp?LekcijaID=<% Response.write
rsLekcije("LekcijaID") %>"><IMG HEIGHT=28 WIDTH=35 SRC="<%
Response.Write IkonaPut %>" BORDER=0 ></A></TD>

<TD WIDTH=466 HEIGHT=34>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH=466>

<TR>

<TD><P><B><FONT FACE="Arial,Helvetica"><% Response.write
rsLekcije("LekcijaNaziv") %> - <% Response.write PrivLekcija
%></B></FONT></TD>

</TR>

</TABLE></TD>

</TR>

<%

```

```

        rsLekcije.MoveNext
loop
%>
        </TABLE></TD>

        <TD></TD>

        </TR>

</TABLE>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>
<TR VALIGN="top" ALIGN="left">
<TD>

        <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=375>
        <TR VALIGN="top" ALIGN="left">
        <TD></TD>

        <TD WIDTH=325><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Zadaci</B></FONT></TD>

        </TR>

        <TR VALIGN="top" ALIGN="left">
        <TD COLSPAN=2 HEIGHT =61></TD>

        </TR>

</TABLE>

</TD>

<TD>

        <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=325>

```

```

<TR VALIGN="top" ALIGN="left">
  <TD HEIGHT =98></TD>
  <TD WIDTH=180 ALIGN="left" VALIGN="top"><IMG HEIGHT=98
WIDTH=180 SRC="../../sl/10.gif" BORDER=0 ALT="Picture" ></TD>
  <TD></TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>

```

```

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
  <TR VALIGN="top" ALIGN="left">
    <TD HEIGHT =94></TD>
    <TD WIDTH=525>
      <TABLE BORDER=1 CELLSPACING=3 CELLPADDING=1 >
        <%

```

```

Do until rsZadaci.EOF

```

```

  PrivZadatak = rsZadaci("ZadatakOpis")

```

```

  set rsZadatakUzet = conn.Execute("SELECT UcenikZadatakID " _
    & " from eKarteloUcenikZadaci where ZadatakID = " _
    & rsZadaci("ZadatakID") & " and UcenikID = " _
    & Session("UcenikID"))

```

```

  if rsZadatakUzet.EOF then

```

```

    IkonaPut = "../../sl/3.gif"

```

```

  else

```



```

        IkonaPut = "../sl/3check.gif"

    end if

%>

    <TR VALIGN="top" ALIGN="left">

        <TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
        HREF="../asp/zadatak.asp?ZadatakID=<% Response.Write
        rsZadaci("ZadatakID") %>"><IMG HEIGHT=28 WIDTH=35 SRC="<%
        Response.Write IkonaPut %>" BORDER=0 ></A></TD>

        <TD WIDTH=466 HEIGHT=44>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH=466>

        <TR>

            <TD><P><B><FONT FACE="Arial,Helvetica"><% Response.Write
            rsZadaci("ZadatakNaziv") %> - <% Response.Write PrivZadatak
            %></B></FONT></TD>

        </TR>

    </TABLE></TD>

    </TR>

%<

    rsZadaci.Movenext

loop

%>

    </TABLE></TD>

    <TD></TD>

    </TR>

    </TABLE>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>

<TR VALIGN="top" ALIGN="left">

<TD>

```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=238>
  <TR VALIGN="top" ALIGN="left">
    <TD HEIGHT =151></TD>
    <TD WIDTH=188 ALIGN="left" VALIGN="top"><IMG HEIGHT=151
WIDTH=188 SRC=" ../sl/10.gif" BORDER=0 ALT="Slika" ></TD>
  </TR>
</TABLE>
</TD>
<TD>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=462>
  <TR VALIGN="top" ALIGN="left">
    <TD></TD>
    <TD WIDTH=325><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Testovi</B></FONT></TD>
  </TR>
  <TR VALIGN="top" ALIGN="left">
    <TD COLSPAN=3 HEIGHT =115></TD>
  </TR>
</TABLE>
</TD>
</TR>
</TABLE>
```

```

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
  <TR VALIGN="top" ALIGN="left">
    <TD HEIGHT =84></TD>
    <TD WIDTH=525>
      <TABLE BORDER=1 CELLSPACING=3 CELLPADDING=1 >
<%
Do until rsTestovi.EOF
  PrivTest = rsTestovi("TestOpis")

  set rsTestUspjeh = Conn.Execute("select Stanje from
eKarteloUcenikTestovi " _
  & "where TestID = " & rsTestovi("TestID") & " and UcenikID = "
-
  & session("UcenikID"))

  if rsTestUspjeh.EOF then
    IkonaPut = "../sl/3.gif"
  elseif rsTestUspjeh("Stanje") = "USPJEH" then
    IkonaPut = "../sl/3check.gif"
  else
    IkonaPut = "../sl/3x.jpg"
  end if

%>

  <TR VALIGN="top" ALIGN="left">
    <TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
  HREF="../asp/test.asp?TestID=<% Response.Write rsTestovi("TestID")
  %>"><IMG HEIGHT=28 WIDTH=35 SRC="<% Response.Write IkonaPut %>"
  BORDER=0 ></A></TD>

```

```

        <TD WIDTH=466 HEIGHT=34>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH=466>

    <TR>

        <TD><P><B><FONT FACE="Arial,Helvetica"><% Response.Write
rsTestovi("TestNaziv") %> - <% Response.Write PrivTest
%></B></FONT></TD>

    </TR>

</TABLE></TD>

</TR>

<%
    rsTestovi.MoveNext
Loop
%>

    </TABLE></TD>

<TD></TD>

</TR>

</TABLE>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>
<TR VALIGN="top" ALIGN="left">
<TD>
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=567>
<TR VALIGN="top" ALIGN="left">
<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=110>
    <TR VALIGN="top" ALIGN="left">
    <TD HEIGHT =28></TD>

```



```

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=567>

  <TR VALIGN="top" ALIGN="left">

    <TD HEIGHT =144></TD>

    <TD WIDTH=346 ROWSPAN=2 ALIGN="left" VALIGN="top"><A
HREF="../asp/ucenik_izbornik.asp"><IMG HEIGHT=96 WIDTH=150
SRC="../sl/4.gif" BORDER=0 ALT="Ulaz" ></A></TD>

  <TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

  <TD></TD>

  <TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">Ima jo&#353 </B></FONT><A
HREF="mailto:skola-e-92@st.tel.hr"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">eKartelo</B></FONT></A><B><FONT SIZE="-1"
FACE="Arial,Helvetica"></B></FONT></TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=3 HEIGHT =24></TD>

  </TR>

</TABLE>

</TD>

<TD>

  <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=133>

    <TR VALIGN="top" ALIGN="left">

      <TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
SRC="../sl/5.gif" BORDER=0 ALT="skola-e-92@st.tel.hr" ></A></TD>

    <TD></TD>

```



```
</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =16></TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</BODY>

</HTML>
```

razredi.asp

```
<%@ Language=VBScript %>

<%

if isempty(Session("UcenikID")) then

    Response.Redirect "../index.asp"

end if

set conn = server.createobject ("adodb.connection")

conn.open "eLearningKartelo"

set rsAktivnaZanimanja = conn.Execute ("SELECT
eKarteloUcenikZanimanja.UcenikID,
eKarteloUcenikZanimanja.ZanimanjeID, " _

    & "eKarteloZanimanja.ZanimanjeNaziv,
eKarteloZanimanja.ZanimanjeOpis " _

    & "FROM eKarteloUcenikZanimanja INNER JOIN eKarteloZanimanja ON
" _
```

```

    & "eKarteloUcenikZanimanja.ZanimanjeID =
eKarteloZanimanja.ZanimanjeID " _

    & "WHERE (eKarteloUcenikZanimanja.UcenikID = " &
Session("UcenikID") _

    & " AND eKarteloUcenikZanimanja.Stanje = 'UTijeku')")

%>

<HTML>

<HEAD>

<TITLE>Razredi</TITLE>

</HEAD>

<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>

        <TR VALIGN="top" ALIGN="left">

            <TD WIDTH=14><IMG
SRC="../../assets/auto_generated_images/img_81e4140.gif" WIDTH =14
HEIGHT =1 BORDER=0></TD>

            <TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Odaberite razred u kojem
&#382elite u&#269iti</B></FONT></TD>

        </TR>

    </TABLE>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>

        <TR VALIGN="top" ALIGN="left">

            <TD HEIGHT =125></TD>

            <TD WIDTH=519>

                <TABLE BORDER=1 CELLSPACING=3 CELLPADDING=1 >

```

```

do until rsAktivnaZanimanja.eof

%>

    <TR VALIGN="top" ALIGN="left">

        <TD WIDTH=57 ALIGN="center" VALIGN="middle"><A
        HREF="../asp/razred.asp?ZanimanjeID=<% response.write
        rsAktivnaZanimanja("ZanimanjeID") %>"><IMG HEIGHT=57 WIDTH=50
        SRC="../sl/8.gif" BORDER=0 ALT = "U&#273ite u razred" ></A></TD>

        <TD WIDTH=447 HEIGHT=57><P><B><FONT
        FACE="Arial,Helvetica"><% response.write
        rsAktivnaZanimanja("ZanimanjeNaziv") %> - <% response.write
        rsAktivnaZanimanja("ZanimanjeOpis") %></B></FONT></TD>

    </TR>

<%
    rsAktivnaZanimanja.Movenext
loop
%>

    </TABLE></TD>

<TD></TD>

</TR>

</TABLE>

<TABLE CELLSPACING=0 CELLPADDING=0 BORDER=0 WIDTH=700>

<TR VALIGN="top" ALIGN="left">

<TD>

    <TABLE BORDER=0 CELLPACING=0 CELLPADDING=0 WIDTH=176>

    <TR VALIGN="top" ALIGN="left">

    <TD HEIGHT =96></TD>

```

```

        <TD WIDTH=150 ALIGN="left" VALIGN="top"><A
        HREF="../asp/ucenik_izbornik.asp"><IMG HEIGHT=96 WIDTH=150
        SRC="../sl/4.gif" BORDER=0 ALT="Povratak na ulaz" ></A></TD>

        </TR>

        <TR VALIGN="top" ALIGN="left">

        <TD COLSPAN=2 HEIGHT =303></TD>

        </TR>

    </TABLE>

</TD>

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=380>

        <TR VALIGN="top" ALIGN="left">

        <TD></TD>

        <TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT"><B><FONT SIZE="-1"
        FACE="Arial,Helvetica">Ima jo&#353 </B></FONT><A
        HREF="mailto:skola-e-92@st.tel.hr"><B><FONT SIZE="-1"
        FACE="Arial,Helvetica">eKartelo</B></FONT></A><B><FONT SIZE="-1"
        FACE="Arial,Helvetica"></B></FONT></TD>

        </TR>

        <TR VALIGN="top" ALIGN="left">

        <TD COLSPAN=2 HEIGHT =201></TD>

        </TR>

    </TABLE>

</TD>

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=144>

```

```

<TR VALIGN="top" ALIGN="left">

  <TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
  HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
  SRC=" ../sl/5.gif" BORDER=0 ALT="skola-e-92@st.tel.hr" ></A></TD>

  <TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =193></TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</BODY>

</HTML>

```

test.asp

```

<%@ Language=VBScript %>

<%

if isempty(Session("UcenikID")) then

    Response.Redirect "../index.asp"

end if

if isempty(Request.QueryString("TestID")) then

    Response.Redirect "../asp/ucenik_izbornik.asp"

end if

```

```

set conn = server.createobject ("adodb.connection")

conn.open "eLearningKartelo"

set rsTest = conn.Execute ("SELECT
eKarteloZanimanja.ZanimanjeNaziv, " _

    & "eKarteloTestovi.TestNaziv " _

    & "FROM eKarteloTestovi INNER JOIN eKarteloZanimanja ON " _

    & "eKarteloTestovi.ZanimanjeID = eKarteloZanimanja.ZanimanjeID
" _

    & "WHERE eKarteloTestovi.TestID = " &
Request.QueryString("TestID"))

```

```

set rsTestPitanja = conn.Execute("select TestPitanjeID, " _

    & "TestPitanje from eKarteloTestPitanja where TestID = " _

    & Request.QueryString("TestID"))

```

```
%>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Test</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">
```

```
<FORM ACTION="../asp/testrezultat.asp" METHOD=POST>
```

```
<INPUT TYPE=HIDDEN NAME="TestID" VALUE = "<% Response.Write
Request.QueryString("TestID") %>">
```

```
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD>
```

```

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=188>

  <TR VALIGN="top" ALIGN="left">

    <TD WIDTH=188 HEIGHT =151 ALIGN="left" VALIGN="top"><IMG
HEIGHT=151 WIDTH=188 SRC="../sl/10.gif" BORDER=0 ALT="Picture"
></TD>

  </TR>

</TABLE>

</TD>

<TD>

  <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=512>

    <TR VALIGN="top" ALIGN="left">

      <TD WIDTH=392><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Zanimanje <% Response.Write
rsTest("ZanimanjeNaziv") %>: Test <% Response.Write
rsTest("TestNaziv") %></B></FONT></TD>

    </TR>

    <TR VALIGN="top" ALIGN="left">

      <TD COLSPAN=3 HEIGHT =77></TD>

    </TR>

  </TABLE>

</TD>

</TR>

</TABLE>

<%

Do Until rsTestPitanja.EOF

  PitanjeBroj = PitanjeBroj + 1

  set rsOdgovori = conn.Execute("select TestOdgovor from
eKarteloTestOdgovori " _

```

```

        & "where TestPitanjeID = " & rsTestPitanja("TestPitanjeID"))
%>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
  <TR VALIGN="top" ALIGN="left">
    <TD></TD>

    <TD WIDTH=589><P><B>Pitanje <% response.write PitanjeBroj %>:
</B><% Response.write rsTestPitanja("TestPitanje") %></TD>

    <TD></TD>
  </TR>
</TABLE>

<TABLE CELLSPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>
<TR VALIGN="top" ALIGN="left">
<TD>

  <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=131>
    <TR VALIGN="top" ALIGN="left">
      <TD></TD>

      <TD WIDTH=106><P><B>Odgovor</B></TD>

    </TR>

    <TR VALIGN="top" ALIGN="left">
      <TD COLSPAN=2 HEIGHT =5></TD>
    </TR>
  </TABLE>
</TD>

```



```

<TD>

    <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=569>

        <TR VALIGN="top" ALIGN="left">

            <TD HEIGHT =24></TD>

            <TD WIDTH=215><SELECT NAME="<% Response.Write
rsTestPitanja("TestPitanjeID") %>" >

                <%

                    Do Until rsOdgovori.EOF

                %>

                    <OPTION VALUE="<% Response.write rsOdgovori("TestOdgovor")
%>"><% Response.write rsOdgovori("TestOdgovor") %></OPTION>

                <%

                    rsOdgovori.MoveNext

                Loop

                %>

            </SELECT></TD>

            <TD></TD>

        </TR>

    </TABLE>

</TD>

</TR>

</TABLE>

<%

    rsTestPitanja.MoveNext

loop

```

```

%>
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>
<TR VALIGN="top" ALIGN="left">
<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=175>
<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=4 HEIGHT =31></TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD HEIGHT =24></TD>
<TD WIDTH=141 COLSPAN=2><INPUT TYPE="submit" NAME="TestIt"
VALUE="Ocjenite moj test" ></TD>
<TD></TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=4 HEIGHT =76></TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD HEIGHT =96></TD>
<TD WIDTH=150 COLSPAN=3 ALIGN="left" VALIGN="top"><A HREF =
"../asp/ucenik_izbornik.asp" ><IMG HEIGHT=96 WIDTH=150
SRC="../sl/4.gif" BORDER=0 ALT="Ulaz" ></A></TD>
</TR>
<TR VALIGN="top" ALIGN="left">
<TD COLSPAN=4 HEIGHT =329></TD>

```

```

</TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=342>

  <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=3 HEIGHT =270></TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD></TD>

    <TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">Ima jos </B></FONT><A HREF="mailto:skola-e-
92@st.tel.hr"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">eKartelo</B></FONT></A><B><FONT SIZE="-1"
FACE="Arial,Helvetica"></B></FONT></TD>

  </TR>

  <TR VALIGN="top" ALIGN="left">

    <TD COLSPAN=3 HEIGHT =249></TD>

  </TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=183>

  <TR VALIGN="top" ALIGN="left">

```

```
<TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
SRC=" ../sl/5.gif" 4 BORDER=0 ALT="skola-e-92@st.tel.hr"
></A></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =241></TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</FORM></BODY>

</HTML>
```

testrezultat.asp

```
<%@ Language=VBScript %>

<%

if isempty(Session("UcenikID")) then

    Response.Redirect "../index.asp"

end if

if isempty(Request.Form("TestID")) then

    Response.Redirect "../index.asp"

end if
```

```

set conn = server.createobject ("adodb.connection")

conn.open "eLearningKartelo"

set rsTest = conn.execute ("SELECT ZanimanjeID, TestNaziv,
DovoljnoZaProci FROM " _

    & "eKarteloTestovi where TestID = " & Request.Form("TestID"))

BrojTocnih = 0

for each Pitanje in Request.Form

    if Pitanje <> "TestIt" and Pitanje <> "TestID" then

        set RSPitanje = conn.Execute("select TestOdgovor from " _

            & "eKarteloTestPitanja where TestPitanjeID = " _

            & Pitanje)

        if Request.Form(Pitanje) = RSPitanje("TestOdgovor") then

            BrojTocnih = BrojTocnih + 1

        end if

    end if

end if

next

conn.Execute "delete from eKarteloUcenikTestovi WHERE UcenikID = "
_

    & session("UcenikID") & " and TestID = " _

    & Request.Form("TestID")

If BrojTocnih < rsTest("DovoljnoZaProci") then

    Poruka = "Odgovorili ste na " & BrojTocnih _

    & " pitanja to&#269no. Za pro&#263i vam treba " _

    & rsTest("DovoljnoZaProci") & " to&#269nih odgovora. Pali ste."

```

```

    conn.Execute "insert Into eKarteloUcenikTestovi (TestID,
UcenikID, Stanje) " _

    & "values (" & Request.Form("TestID") & ", " &
session("UcenikID") _

    & ", 'Neuspjeh')"

else

    Poruka = "Stekli ste " & BrojTocnih _

    & " to&#269nih odgovora. Potrebno je " _

    & rsTest("DovoljnoZaProci") & " to&#269nih. Pro&#353li ste."

    conn.Execute "insert Into eKarteloUcenikTestovi (TestID,
UcenikID, Stanje) " _

    & "values (" & Request.Form("TestID") & ", " &
session("UcenikID") _

    & ", 'USPJEH')"

end if

```

'Test za kraj razreda

ZanimanjeZavrшено = "Yes"

```

set rsSviTestovi = conn.Execute("select TestID from
eKarteloTestovi " _

```

```

    & "where ZanimanjeID = " & rsTest("ZanimanjeID"))

```

do until rsSviTestovi.EOF

```

    set rsTestUSPJEH = Conn.Execute("select TestID from
eKarteloUcenikTestovi " _

```

```

    & "where TestID = " & rsSviTestovi("TestID") & " and UcenikID =
" _

```

```

    & session("UcenikID") & " and Stanje = 'USPJEH'"

```

```

    if rsTestUSPJEH.EOF then

```

```

        ZanimanjeZavrsono = "No"

    end if

    rsSviTestovi.MoveNext

loop

if ZanimanjeZavrsono = "Yes" then

    Poruka = Poruka & " Zanimanje je zavrsono!"

    conn.Execute "update eKarteloUcenikZanimanja set " _
    & "StanjeDatum = '" & Date & "', Stanje = 'Zavrsono' " _
    & "where UcenikID = " & Session("UcenikID") & " and " _
    & " ZanimanjeID = " & rsTest("ZanimanjeID")

end if

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

<HTML>

<HEAD>

<TITLE>Izbornik u&#269enika</TITLE>

</HEAD>

<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>

<TR VALIGN="top" ALIGN="left">

```

```

<TD></TD>

<TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+2" FACE="Arial,Helvetica">Test: <% Response.Write
rsTest("TestNaziv") %>.<BR><% Response.Write Poruka
%></B></FONT></TD>

<TD></TD>

</TR>

</TABLE>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>

<TR VALIGN="top" ALIGN="left">

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=201>

<TR VALIGN="top" ALIGN="left">

<TD HEIGHT =96></TD>

<TD WIDTH=150 ALIGN="left" VALIGN="top"><IMG HEIGHT=96
WIDTH=150 SRC="../../sl/4.gif" BORDER=0 ALT="Ulaz" ></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =357></TD>

</TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=499>

<TR VALIGN="top" ALIGN="left">

```



```

<TD HEIGHT =164></TD>

<TD WIDTH=376>

    <TABLE BORDER=1 CELLSPACING=3 CELLPADDING=1 >

        <TR VALIGN="top" ALIGN="left">

            <TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
    HREF="../asp/promjena_podataka.asp"><IMG HEIGHT=28 WIDTH=35
    SRC="../s1/3.gif" BORDER=0 ></A></TD>

            <TD WIDTH=317 HEIGHT=34>

                <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=9 WIDTH=317>

                    <TR>

                        <TD><P><B><FONT FACE="Arial,Helvetica">Promjenite svoje
    podatke.</B></FONT></TD>

                    </TR>

                </TABLE></TD>

            </TR>

        <TR VALIGN="top" ALIGN="left">

            <TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
    HREF="../asp/kreniiodustani.asp"><IMG HEIGHT=28 WIDTH=35
    SRC="../s1/3.gif" BORDER=0 ></A></TD>

            <TD WIDTH=317 HEIGHT=35>

                <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=9 WIDTH=317>

                    <TR>

                        <TD><P><B><FONT FACE="Arial,Helvetica">Pokreni ili zaustavi
    zanimanje</B></FONT></TD>

                    </TR>

                </TABLE></TD>

            </TR>

        <TR VALIGN="top" ALIGN="left">

```

```
<TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
HREF="../asp/razredi.asp"><IMG HEIGHT=28 WIDTH=35
SRC="../sl/3.gif" BORDER=0 ></A></TD>
```

```
<TD WIDTH=317 HEIGHT=36>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=9 WIDTH=317>
```

```
<TR>
```

```
<TD><P><B><FONT FACE="Arial,Helvetica">Idi u
razred</B></FONT></TD>
```

```
</TR>
```

```
</TABLE></TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
HREF="../asp/izvjesce.asp"><IMG HEIGHT=28 WIDTH=35
SRC="../sl/3.gif" p BORDER=0 ></A></TD>
```

```
<TD WIDTH=317 HEIGHT=34>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=9 WIDTH=317>
```

```
<TR>
```

```
<TD><P><B><FONT FACE="Arial,Helvetica">Izvje&#353&#263e mog
rada</B></FONT></TD>
```

```
</TR>
```

```
</TABLE></TD>
```

```
</TR>
```

```
</TABLE></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=499>
```

```
<TR VALIGN="top" ALIGN="left">
```

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=332>

<TR VALIGN="top" ALIGN="left">

<TD></TD>

<TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT">Ima još eKartelo</TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =175></TD>

</TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=167>

<TR VALIGN="top" ALIGN="left">

<TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =167></TD>

</TR>

```
</TABLE>
</TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

ucenik_izbornik.asp

```
<%@ Language=VBScript %>
<%
if isempty(Session("UcenikID")) then
    Response.Redirect "../index.asp"
end if

set conn = server.createobject ("adodb.connection")
conn.open "eLearningKartelo"

set rsUcenik = conn.execute ("SELECT Ime FROM eKarteloUcenici
where UcenikID = " & Session("UcenikID"))

%>

<HTML>
<HEAD>
<TITLE>Izbornik u&#269;enika</TITLE>
</HEAD>
```

```
<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD></TD>
```

```
<TD WIDTH=633><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Dobro nam se vratili <%
Response.Write rsUcenik("Ime") %>.<BR>&#352to &#382elite
raditi?</B></FONT></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=700>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=201>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD HEIGHT =96></TD>
```

```
<TD WIDTH=150 ALIGN="left" VALIGN="top"><IMG HEIGHT=96
WIDTH=150 SRC=" ../sl/4.gif" BORDER=0 ALT="Ulazna prostorija"
></TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD COLSPAN=2 HEIGHT =357></TD>
```

```
</TR>
```

```

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=499>

  <TR VALIGN="top" ALIGN="left">

    <TD HEIGHT =164></TD>

    <TD WIDTH=376>

      <TABLE BORDER=1 CELLSPACING=3 CELLPADDING=1 >

        <TR VALIGN="top" ALIGN="left">

          <TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
HREF=" ../asp/promjena_podataka.asp"><IMG HEIGHT=28 WIDTH=35
SRC=" ../sl/3.gif" BORDER=0 ></A></TD>

          <TD WIDTH=317 HEIGHT=34>

            <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=9 WIDTH=317>

              <TR>

                <TD><P><B><FONT FACE="Arial,Helvetica">Promjena podataka
u&#269enika.</B></FONT></TD>

              </TR>

            </TABLE></TD>

          </TR>

        </TABLE></TD>

      </TR>

    <TR VALIGN="top" ALIGN="left">

      <TD WIDTH=44 ALIGN="center" VALIGN="middle"><A
HREF=" ../asp/kreniliodustani.asp"><IMG HEIGHT=28 WIDTH=35
SRC=" ../sl/3.gif" BORDER=0 ></A></TD>

      <TD WIDTH=317 HEIGHT=35>

        <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=9 WIDTH=317>

          <TR>

```

```
<TD><P><B><FONT FACE="Arial,Helvetica">Upi&#353ite se ili  
ispi&#353ite iz zanimanja</B></FONT></TD>
```

```
</TR>
```

```
</TABLE></TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD WIDTH=44 ALIGN="center" VALIGN="middle"><A  
HREF=" ../asp/razredi.asp"><IMG HEIGHT=28 WIDTH=35  
SRC=" ../sl/3.gif" BORDER=0 ></A></TD>
```

```
<TD WIDTH=317 HEIGHT=36>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=9 WIDTH=317>
```

```
<TR>
```

```
<TD><P><B><FONT FACE="Arial,Helvetica">Idem u  
razred</B></FONT></TD>
```

```
</TR>
```

```
</TABLE></TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD WIDTH=44 ALIGN="center" VALIGN="middle"><A  
HREF=" ../asp/izvjesce.asp"><IMG HEIGHT=28 WIDTH=35  
SRC=" ../sl/3.gif" BORDER=0 ></A></TD>
```

```
<TD WIDTH=317 HEIGHT=34>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=9 WIDTH=317>
```

```
<TR>
```

```
<TD><P><B><FONT FACE="Arial,Helvetica">Izvj&#353&#263e o  
mom radu</B></FONT></TD>
```

```
</TR>
```

```
</TABLE></TD>
```

```
</TR>
```

```

        </TABLE></TD>

<TD></TD>

</TR>

</TABLE>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=499>

<TR VALIGN="top" ALIGN="left">

<TD>

        <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=332>

        <TR VALIGN="top" ALIGN="left">

        <TD></TD>

        <TD WIDTH=146><P ALIGN="RIGHT"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">Ostali aktivni site-ovi</B></FONT><A
HREF="mailto:skola-e-92@st.tel.hr"><B><FONT SIZE="-1"
FACE="Arial,Helvetica">eKartelo</B></FONT></A><B><FONT SIZE="-1"
FACE="Arial,Helvetica"></B></FONT></TD>

        </TR>

        <TR VALIGN="top" ALIGN="left">

        <TD COLSPAN=2 HEIGHT =175></TD>

        </TR>

        </TABLE>

</TD>

<TD>

        <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=167>

        <TR VALIGN="top" ALIGN="left">

```



```

    <TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"><A
    HREF="mailto:skola-e-92@st.tel.hr"><IMG HEIGHT=70 WIDTH=93
    SRC="../../../sl/5.gif" BORDER=0 ALT="skola-e-92@st.tel.hr" ></A></TD>

    <TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =167></TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</BODY>

</HTML>

```

zadatak.asp

```

<%@ Language=VBScript %>

<%

if isempty(Session("UcenikID")) then

    Response.Redirect "../index.asp"

end if

if isempty(Request.QueryString("ZadatakID")) then

    Response.Redirect "../asp/ucenik_izbornik.asp"

```

```

end if

set conn = server.createobject ("adodb.connection")
conn.open "eLearningKartelo"

set rsZadatakUzet = conn.Execute("SELECT UcenikZadatakID " _
    & " from eKarteloUcenikZadaci where ZadatakID = " _
    & Request.QueryString("ZadatakID") & " and UcenikID = " _
    & Session("UcenikID"))

if rsZadatakUzet.EOF then

    conn.Execute "insert into eKarteloUcenikZadaci (ZadatakID,
UcenikID) " _
        & "values (" & Request.QueryString("ZadatakID") & ", " _
        & Session("UcenikID") & ")"

end if

set rsZadaci = conn.Execute ("SELECT
eKarteloZanimanja.ZanimanjeNaziv, " _
    & "eKarteloZadaci.ZadatakNaziv, eKarteloZadaci.Zadatak " _
    & "FROM eKarteloZadaci INNER JOIN eKarteloZanimanja ON " _
    & "eKarteloZadaci.ZanimanjeID = eKarteloZanimanja.ZanimanjeID
" _
    & "WHERE eKarteloZadaci.ZadatakID = " &
Request.QueryString("ZadatakID"))

ZadatakTekst = rsZadaci("Zadatak")

%>

<HTML>

<HEAD>

<TITLE>Zadatak</TITLE>

```

```

</HEAD>

<BODY BGCOLOR="#FFFFFF" LINK="#008080" VLINK="#004080"
TEXT="#000000">

<TABLE CELLSPACING=0 CELLPADDING=0 BORDER=0 WIDTH=640>

<TR VALIGN="top" ALIGN="left">

<TD>

    <TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0 WIDTH=205>

        <TR VALIGN="top" ALIGN="left">

            <TD HEIGHT =98></TD>

            <TD WIDTH=180 ALIGN="left" VALIGN="top"><IMG HEIGHT=98
WIDTH=180 SRC="../../../sl/12.gif" BORDER=0 ALT="Slika" ></TD>

        </TR>

        <TR VALIGN="top" ALIGN="left">

            <TD COLSPAN=2 HEIGHT =5></TD>

        </TR>

    </TABLE>

</TD>

<TD>

    <TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0 WIDTH=435>

        <TR VALIGN="top" ALIGN="left">

            <TD WIDTH=392><P ALIGN="CENTER"><B><FONT COLOR="#0000A0"
SIZE="+3" FACE="Arial,Helvetica">Zanimanje <% Response.Write
rsZadaci("ZanimanjeNaziv") %><br> Zadatak:</br> <% Response.Write
rsZadaci("ZadatakNaziv") %></B></FONT></TD>

        </TR>

    </TABLE>

</TD>

```

```
</TR>
```

```
</TABLE>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=640>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD></TD>
```

```
<TD WIDTH=589><P><% Response.Write ZadatakTekst %></TD>
```

```
<TD></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 WIDTH=640>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=176>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD HEIGHT =96></TD>
```

```
<TD WIDTH=150 ALIGN="left" VALIGN="top"><A  
HREF=" ../asp/ucenik_izbornik.asp"><IMG HEIGHT=96 WIDTH=150  
SRC=" ../s1/4.gif" BORDER=0 ALT="Ulaz" ></A></TD>
```

```
</TR>
```

```
<TR VALIGN="top" ALIGN="left">
```

```
<TD COLSPAN=2 HEIGHT =102></TD>
```

```
</TR>
```

```
</TABLE>
```

```
</TD>
```

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=342>

<TR VALIGN="top" ALIGN="left">

<TD></TD>

<TD WIDTH=146 COLSPAN=2><P ALIGN="RIGHT">Ima još eKartelo</TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =22></TD>

</TR>

</TABLE>

</TD>

<TD>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=122>

<TR VALIGN="top" ALIGN="left">

<TD WIDTH=93 HEIGHT =70 ALIGN="left" VALIGN="top"></TD>

<TD></TD>

</TR>

<TR VALIGN="top" ALIGN="left">

<TD COLSPAN=2 HEIGHT =14></TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</BODY>

</HTML>

Literatura

1. Professional Active Server Pages 2.0, Francis, Fedorov, Harrison, Homer, Murphy, Sussman, Smith and Wood, Wrox Press Ltd., 1998.
2. Beginning E-Commerce with Visual Basic, ASP, SQL Server 7.0 and MTS, Matthew Reynolds, Wrox Press Ltd., 2000.
3. Instant ASP Scripts, Second Edition, Greg Buczek, McGraw-Hill, 2000.
4. Designing Active Server Pages, Scott Mitchell, O'Reilly, 2000.
5. MCSD Visual Basic 6 Exms 70-175 and 70-176, Howard Hawhee, New Riders, 1999.