

**Elektronski Fakultet  
NIŠ**

**Katedra za Elektroniku  
Seminarski rad  
iz predmeta  
Mikroprocesorski Sistemi  
Tema:  
Elektronska provera znanja**

**Mentor :  
prof.dr Mile Stojčev**

**Kandidati :  
Igor Marković 11705  
Velibor Mijatović 11397  
Branimir Milić 9993  
Dalibor Bogdanović 9860  
Marko Uraković 10107**

**Septembar 2005**

# SADRŽAJ

<b>-UVOD.....</b>	<b>str.3</b>
<b>-Razlozi za uvođenje elektronske provere znanja.....</b>	<b>str.3</b>
<b>-Prednosti i nedostaci testiranja znanja pomoću računara.....</b>	<b>str.3</b>
<b>-Minimalna konfiguracija računara.....</b>	<b>str.4</b>
<b>-Instaliranje softvera za proveru znanja.....</b>	<b>str.4</b>
<b>-RADNO OKRUŽENJE.....</b>	<b>str.8</b>
<b>-Najopštija blok šema aplikacije .....</b>	<b>str.8</b>
<b>-Evidentiranje studenata od strane nastavnog kadra.....</b>	<b>str.9</b>
<b>-Postupak provere znanja studenata.....</b>	<b>str.11</b>
<b>-Generisanje izveštaja.....</b>	<b>str.14</b>
<b>-Promena naslova predmeta iz kog se vrši provera znanja.....</b>	<b>str.15</b>
<b>-KOMENTARI.....</b>	<b>str.16</b>

## UVOD

### Razlozi za uvođenje elektronske provere znanja

Karakteristike društva u kome živimo, gde su znanje i načini njegovog korišćenja veoma važni u svakodnevnom životu, postavljaju nove izazove koji se tiču visokog obrazovanja. Sve veća upotreba kompjuterske tehnologije u procesu učenja primoravaju visoke obrazovne institucije da računare takodje koriste i u procesu ispitivanja studenata.

U ovom semestralnom radu biće opisan jedan pristup za elektronsko testiranje studenata na Elektroskom fakultetu u Nišu iz grupe predmeta koji se odnose na razvoj i primenu mikroračunarskih sistema. Opisani postupak elektronske provere znanja u skladu je sa tendencijama i preporukama koje se odnose na implementaciju Bolonjskog procesa u našoj zemlji.

### Prednosti i nedostaci testiranja znanja pomoću računara

Osnovni cilj elektronskog testiranja znanja studenata treba da je podređen aktivnijem učešću studenata u procesu izvodjenju nastave, koje se ogleda u:

1. redovnom praćenju gradiva
2. solidnijoj teoretskoj pripremi studenata za izvodjenje laboratorijskih vežbi
3. efikasnijem praćenju auditivnih vežbi

Da bi se ovakav način testiranja sproveo neophodno je :

- a. formirati bazu podataka sa test pitanjima za odgovarajući predmet
- b. jasno formulisati test pitanja
- c. kreirati koncept testa, uključujući podele pitanja po grupama
- d. upoznati studente sa načinom testiranja

Osnovne prednosti ovakvog pristupa provere znanja su sledeći:

- proces ispitivanja je brz jer se rezultat generiše odmah nakon njegovog završetka
- za ispitivanje nije obavezno prisustvo predmetnog nastavnika
- po mišljenju studenata izbegava se subjektivnost profesora kod ocenjivanja
- po mišljenju studenata ovim vidom ispitivanja stiže se viši nivo znanja u odnosu na tradicionalni pristup
- zahteva od studenta bržu reakciju na postavljeno pitanje
- postiže se velika ušteda u vremenu oko pripreme testa i ocenjivanja
- postiže se ušteda u pripremi ispitnog materijala, tj. nije potrebno pripremati blankete
- broj studenata koji se testira može biti veliki

Nedostaci predloženog pristupa provere znanja su sledeći:

- za implementaciju metoda neophodno je ulaganje u računarsku opremu
- ne postoji lični kontakt profesor-student što je sa aspekta socijalizacije nehumano
- pitanje je da li ovakav način testiranja u pravoj meri odražava znanje studenta

Iskustva na našem fakultetu ovakvim načinom provere znanja do sada ne postoje. Vreme će pokazati i dati odgovor na pitanje da li je na ovom polju postignut ikakav napredak, i šta treba dodatno činiti da bi se ostvarili još bolji rezultati.

Na kraju treba naglasiti da pored korišćenja u sistemu obrazovanja, provera znanja zasnovana na računaru može da ima široku primenu i u drugim vidovima testiranja (kvizovi, razgovori za posao, razna anketiranja, itd.).

### **Minimalna konfiguracija računara**

Minimalna konfiguracija hardvera i softvera računara, da bi sistem za elektronsku proveru znanja bio operativan, je:

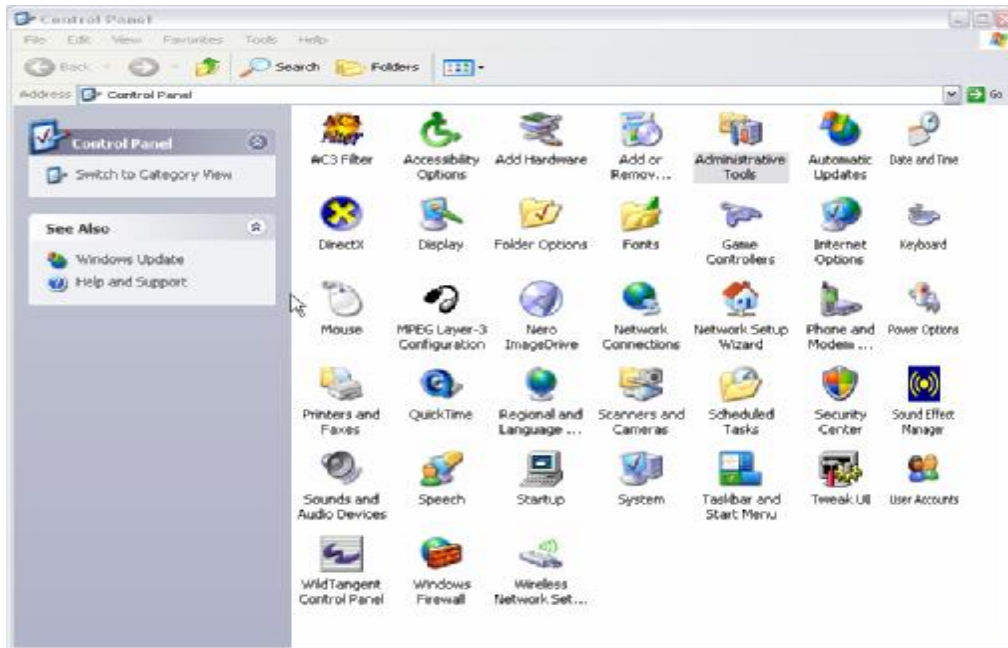
- Personalni računar Pentium IV, na 1 GHz
- RAM memorija 128 MB,
- operativni sistem Windows XP
- miš, tastatura i CD-ROM
- serijski štampač za izveštaj
- instalirani korisnički programi Visual Studio V6 i Office 2000

### **Instaliranje softvera za proveru znanja**

Da bi pokrenuli program za testiranje neophodno je da na računaru bude instaliran jedan od operativnih sistema (preporučuje se Microsoft Windows XP), kao i korisnički programi Microsoft Visual Studio 6 i Microsoft Office 2000. Za uspešno instaliranje programa za testiranje neophodno je da na hard-disku ima 25-50 MB slobodnog prostora. Program za testiranje se nalazi na CD-u, a njegov postupak instalacije čine sledeći koraci:

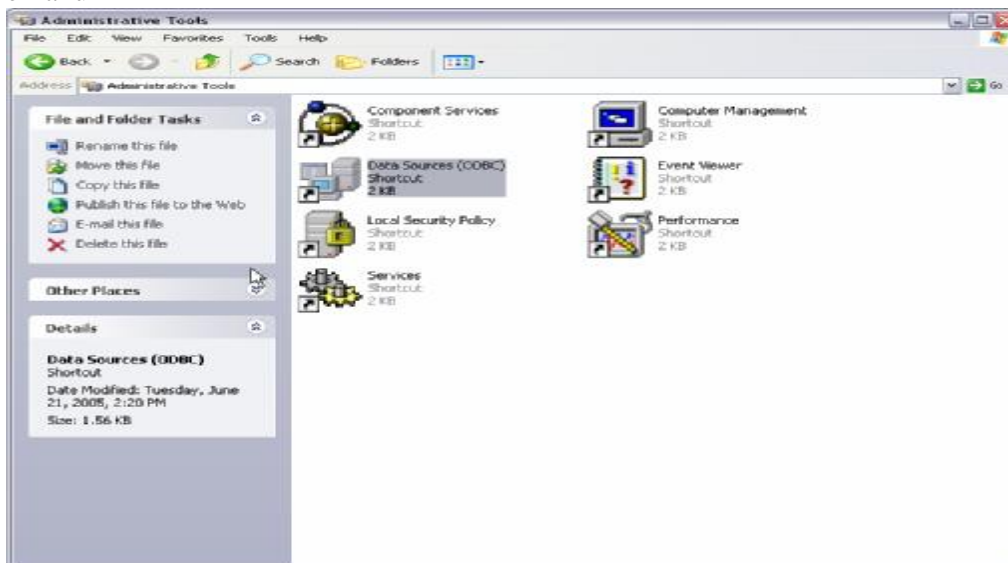
1. Ubaciti instalacioni disk u CD drajv i sačekati nekoliko trenutaka da bi proces čitanja bio moguć.
2. Kliknuti desnim tasterom miša na direktorijum **KOLOKVIJUM**, a zatim levim na komandu **copy**. Odaberati mesto na hard-disku gde treba kopirati sadržaj direktorijuma i naredbom **paste** uraditi to. Treba biti strpljiv nekoliko trenutaka dok računar ne obavi sve pripremne radnje i provere koje su potrebne za kopiranje.
3. Pored direktorijuma **KOLOKVIJUM** na CD-u postoji fajl **korak1.mdb** koji treba iskopirati u direktorijum *My Documents*. Nakon kopiranja proveriti da li je atribut **read-only** uključen, za slučaj da je uključen treba ga isključiti.
4. Ovaj korak se odnosi na instaliranje pokretačkog programa (drajvera) **Microsoft Access Driver \*.mdb** u System DSN. Postupak je sledeći: Klikom na levi taster miša selektujemo polje Start pa u okviru menija bira se polje

Control Panel. Prelaskom na polje *Control Panel* biramo ikonicu *Administrative Tools*. Izgled *Control Panel*-a prikazan je na slici 1.



Slika 1. Izgled *Control Panel*-a

5. Dvostrukim klikom na levi taster miša na ikonicu *Administrative Tools* na ekranu



Slika 2. Izgled *Administrative Tools*

6. Dvostrukim klikom na levi taster miša na ikonicu *Data Sources (ODBC)* sa slike 2 na ekranu CRT-a generiše se prikaz kao na slici 3

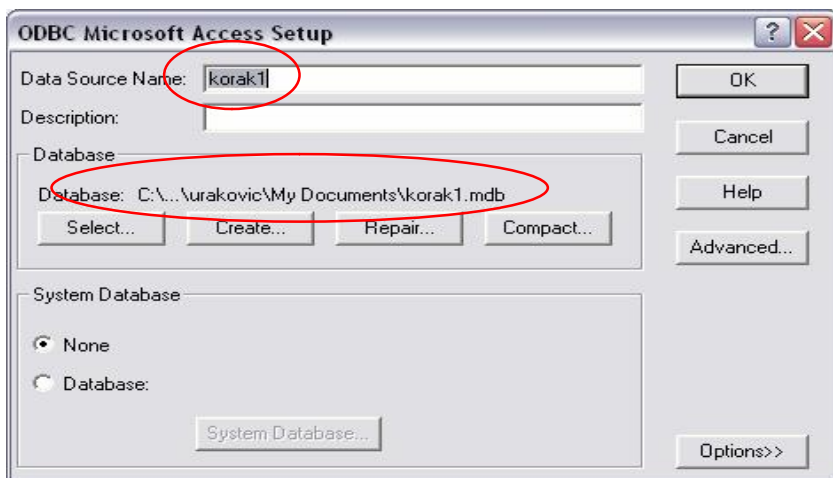


Slika 3 Izgled *ODBC Data Source Administrator*-a

7. U okviru prozora *ODBC Data Source Administrator* izabrati stavku System DSN, zatim kliknuti na polje *Add...*, nakon toga okviru polja *System Data Source...* odabrati *Microsoft Access Driver (\*.mdb)*, na kraju klikom na levi taster miša na polje *OK* potvrditi unos podataka (vidi sliku 3).

8. Na ekranu CRT-a prikazaće se prozor *ODBC Microsoft Access Setup* čiji je izgled prikazan na slici 4. U okviru polja *Data Source Name* treba uneti naziv baze, u konkretnom slučaju je to **korak 1**. Putanja se može izabrati na jedan od sledeća dva načina:

- kada korisnik zna tačnu putanju on je unosi u polje Database (slika 4)
- kada korisnik ne zna tačnu putanju on bira polje Select pa levim klikom tasterom miša u okviru direktorijuma My Documents pretražuje se lokacija gde je fajl korak 1 lociran. Nakon nalaženja putanje pritiskom na polje OK (vidu sliku 4) potvrđuje se izbor putanje.



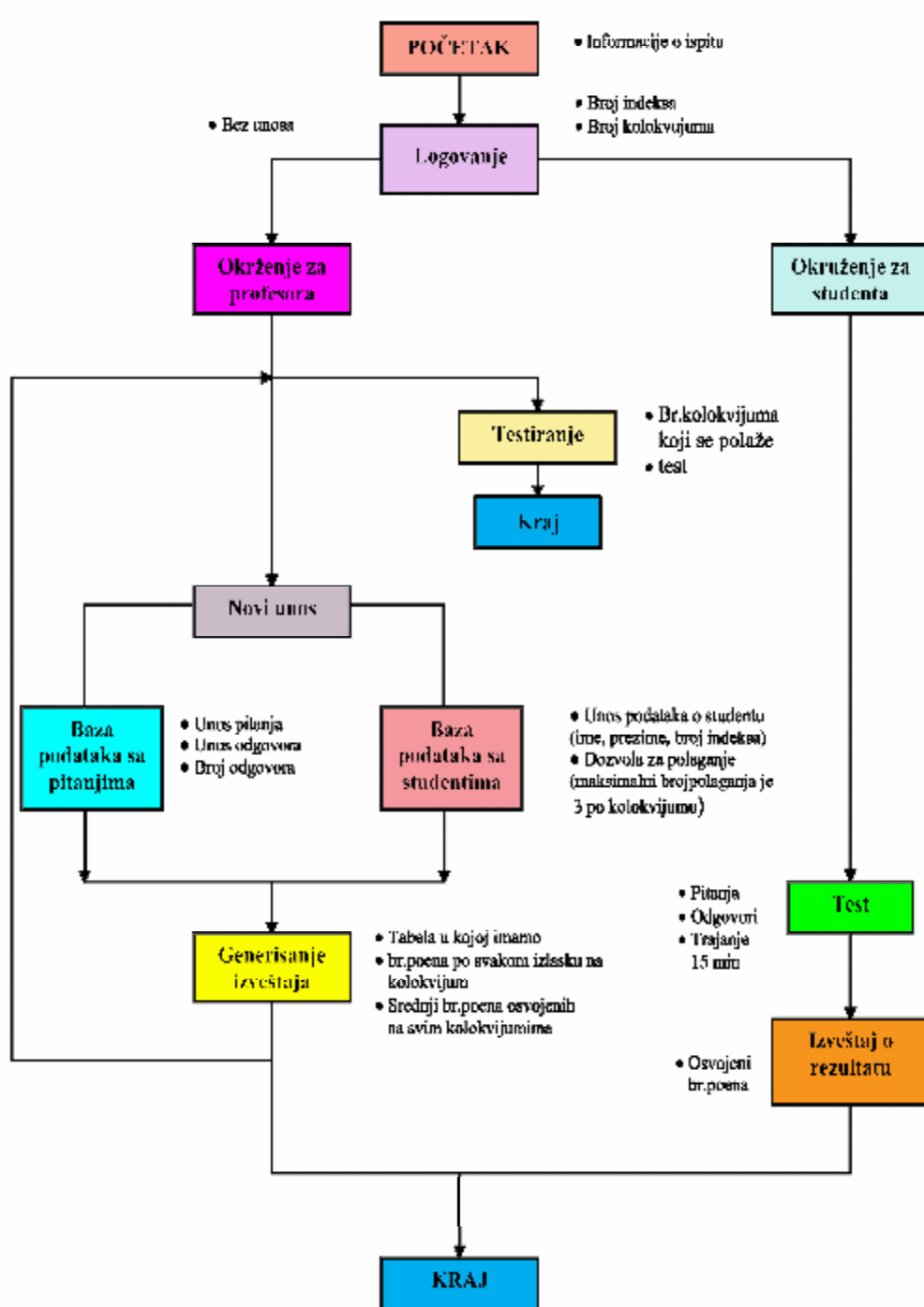
Slika 4 Izgled prozora ODBC Microsoft Access Setup

9.U ovom koraku kreira se prečica KOLOKVIJUM koja se smešta na desktopu. Prečica se kreira na sledeći način.Desnim klikom miša na izvršni fajl kolokvijum dobijamo padajući meni gde treba izabrati opciju **create shortcut**.Ovim postupkom smo dobili duplikat izvršnog fajla.Ostaje nam samo da ga smestimo na desktop jednostavnim komandama **cut i paste**.

*Napomenimo da fajl koji pokreće aplikativni program za elektronsko testiranje studenata se nalazi u poddirektorijumu **Release** pod imenom **KOLOKVIJUM** (korak2 MFC Application), čime je proces instalacije završen.*

# RADNO OKRUŽENJE

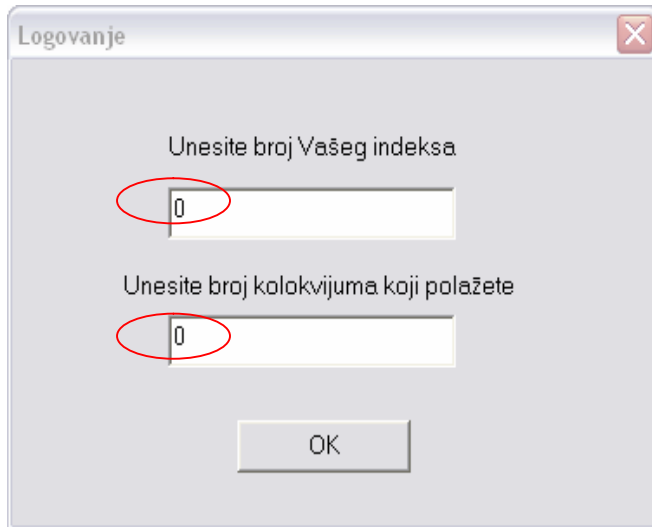
## -najopštija blok šema aplikacije-





## Evidentiranje studenata od strane nastavnog kadra

Ako se kod prozora Logovanje u polja indeks studenta i broj kolokvijuma ne unese ništa tj.ostanu obe nule kao na slici 5 i ako se potvrdi tasterom OK otvara se prozor koji predviđen kao radno okruženje za profesora koji izgleda kao na slici 6.



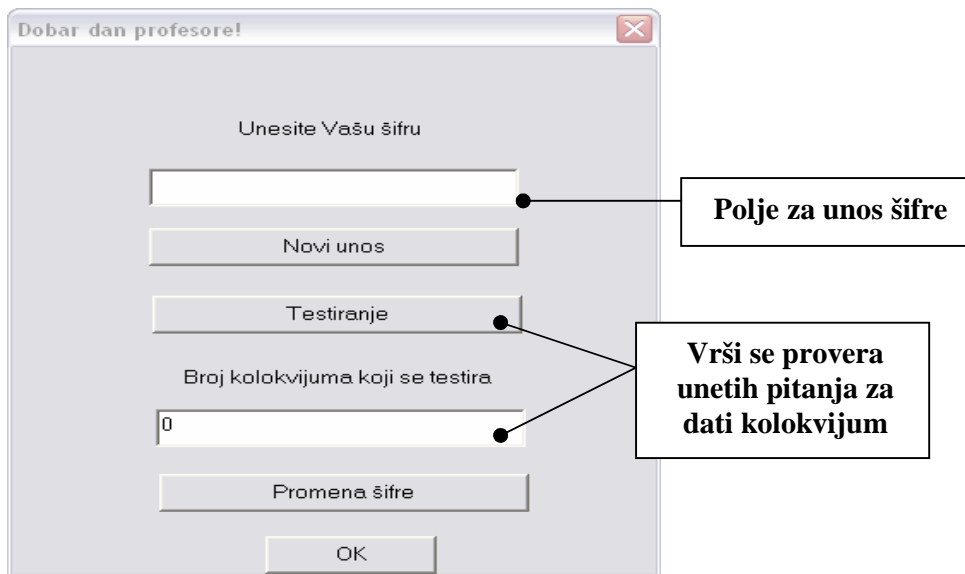
Logovanje

Unesite broj Vašeg indeksa

Unesite broj kolokvijuma koji polažete

OK

Slika 5. Izgled prozora za logovanje profesora



Dobar dan profesore!

Unesite Vašu šifru

Novi unos

Testiranje

Broj kolokvijuma koji se testira

Promena šifre

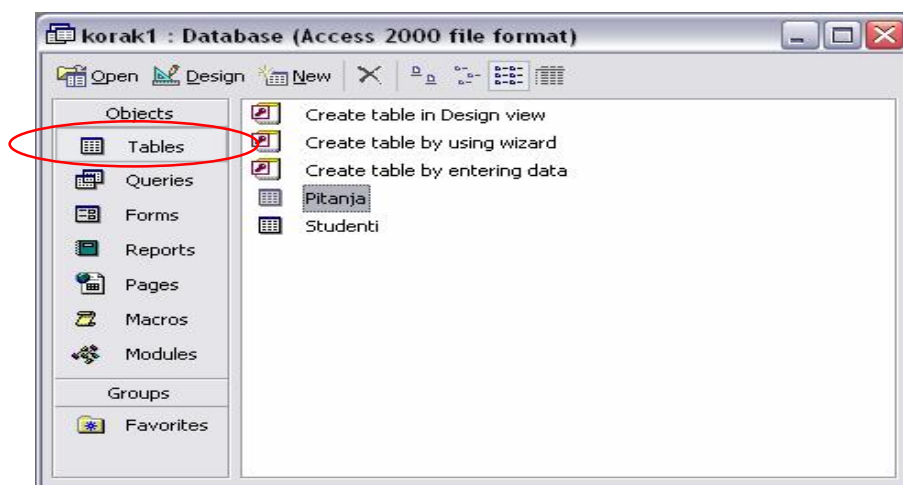
OK

Polje za unos šifre

Vrši se provera unetih pitanja za dati kolokvijum

Slika 6. Izgled radnog okruženja profesora

Pritiskom na taster **novi unos** otvara se sledeći prozor:



Slika 7 Izgled Database (Access 2000 file format) sa slektovanim objektom Tables

U okviru ovog prozora postoje objekti **Tables** ( preko kog ulazimo u bazu pitanja i bazu studenata ) i **Queries** ( gde se nalazi baza za štampanje ). Pritiskom na ikonicu **pitanja** otvara se baza sa pitanjima i odgovorima za kolokvijume:

The screenshot shows a table view with the following columns: 'Redni broj kolokvijuma', 'Redni broj pitanja', 'Tekst pitanja', and 'Ukupan broj odgovora'. The table contains multiple rows of data with various values in the columns.

Redni broj kolokvijuma	Redni broj pitanja	Tekst pitanja	Ukupan broj odgovora
14	11	11. a) 1000	10
15	11	11. a) 1000	10
16	11	11. a) 1000	10
17	11	11. a) 1000	10
18	7	11. a) 1000	10
19	7	11. a) 1000	10
20	7	11. a) 1000	10
21	7	11. a) 1000	10
22	7	11. a) 1000	10
23	7	11. a) 1000	10
24	7	11. a) 1000	10
25	7	11. a) 1000	10
26	7	11. a) 1000	10
27	7	11. a) 1000	10
28	7	11. a) 1000	10
29	7	11. a) 1000	10
30	7	11. a) 1000	10
31	7	11. a) 1000	10
32	7	11. a) 1000	10
33	7	11. a) 1000	10
34	7	11. a) 1000	10
35	7	11. a) 1000	10
36	7	11. a) 1000	10
37	7	11. a) 1000	10
38	7	11. a) 1000	10
39	7	11. a) 1000	10
40	7	11. a) 1000	10
41	7	11. a) 1000	10
42	7	11. a) 1000	10
43	7	11. a) 1000	10
44	7	11. a) 1000	10
45	7	11. a) 1000	10
46	7	11. a) 1000	10
47	7	11. a) 1000	10
48	7	11. a) 1000	10
49	7	11. a) 1000	10
50	7	11. a) 1000	10
51	7	11. a) 1000	10
52	7	11. a) 1000	10
53	7	11. a) 1000	10
54	7	11. a) 1000	10
55	7	11. a) 1000	10
56	7	11. a) 1000	10
57	7	11. a) 1000	10
58	7	11. a) 1000	10
59	7	11. a) 1000	10
60	7	11. a) 1000	10
61	7	11. a) 1000	10
62	7	11. a) 1000	10
63	7	11. a) 1000	10
64	7	11. a) 1000	10
65	7	11. a) 1000	10
66	7	11. a) 1000	10
67	7	11. a) 1000	10
68	7	11. a) 1000	10
69	7	11. a) 1000	10
70	7	11. a) 1000	10
71	7	11. a) 1000	10
72	7	11. a) 1000	10
73	7	11. a) 1000	10
74	7	11. a) 1000	10
75	7	11. a) 1000	10
76	7	11. a) 1000	10
77	7	11. a) 1000	10
78	7	11. a) 1000	10
79	7	11. a) 1000	10
80	7	11. a) 1000	10
81	7	11. a) 1000	10
82	7	11. a) 1000	10
83	7	11. a) 1000	10
84	7	11. a) 1000	10
85	7	11. a) 1000	10
86	7	11. a) 1000	10
87	7	11. a) 1000	10
88	7	11. a) 1000	10
89	7	11. a) 1000	10
90	7	11. a) 1000	10
91	7	11. a) 1000	10
92	7	11. a) 1000	10
93	7	11. a) 1000	10
94	7	11. a) 1000	10
95	7	11. a) 1000	10
96	7	11. a) 1000	10
97	7	11. a) 1000	10
98	7	11. a) 1000	10
99	7	11. a) 1000	10
100	7	11. a) 1000	10

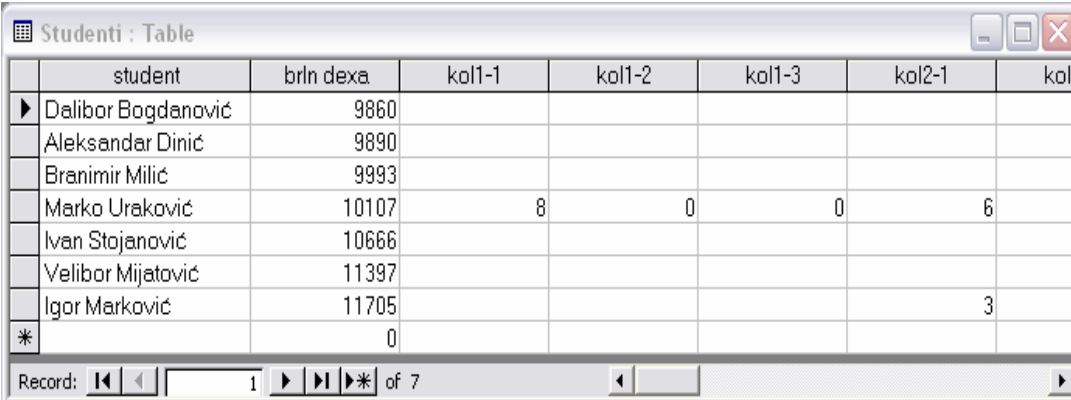
Slika 8 Izgled tabele sa pitanjima i odgovorima

Tekst pitanja može sadržati maksimalno **255 karaktera**. Isto važi i za svaki odgovor ponaosob.

U koloni **odgovor 0** uvek smeštamo tačan odgovor, s tim što u slučaju više tačnih odgovora u ovu kolonu upisujemo redne brojeve kolona u kojima se nalaze tačni odgovori (a za odgovor 1, b za odgovor 2, itd.). Ovde treba voditi računa da se slova koja označavaju tačne odgovore pri upisu odvajaju zarezom pa blanko znakom i između poslednjeg i preposlednjeg slova uvek upisujemo veznik i .Maksimalan broj odgovora na jedno pitanje je 8.

Sadržaj baze sa pitanjima i odgovorima se može obrisati jedino markiranjem svih kolona prethodne tabele sem kolone ID.

Klikom na ikonicu **studenti** otvara se baza sa spiskom studenata koji mogu da polažu kolokvijume koja je prikazana na slici 9.



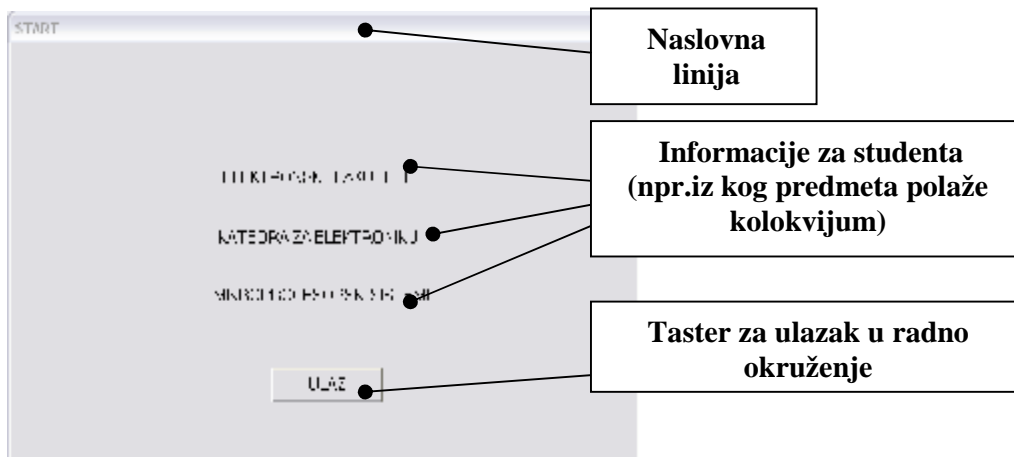
	student	brln dexa	kol1-1	kol1-2	kol1-3	kol2-1	kol:
▶	Dalibor Bogdanović	9860					
	Aleksandar Dinić	9890					
	Branimir Milić	9993					
	Marko Uraković	10107	8	0	0	6	
	Ivan Stojanović	10666					
	Velibor Mijatović	11397					
	Igor Marković	11705				3	
*		0					

Slika 9 Izgled tabele sa spiskom studenata

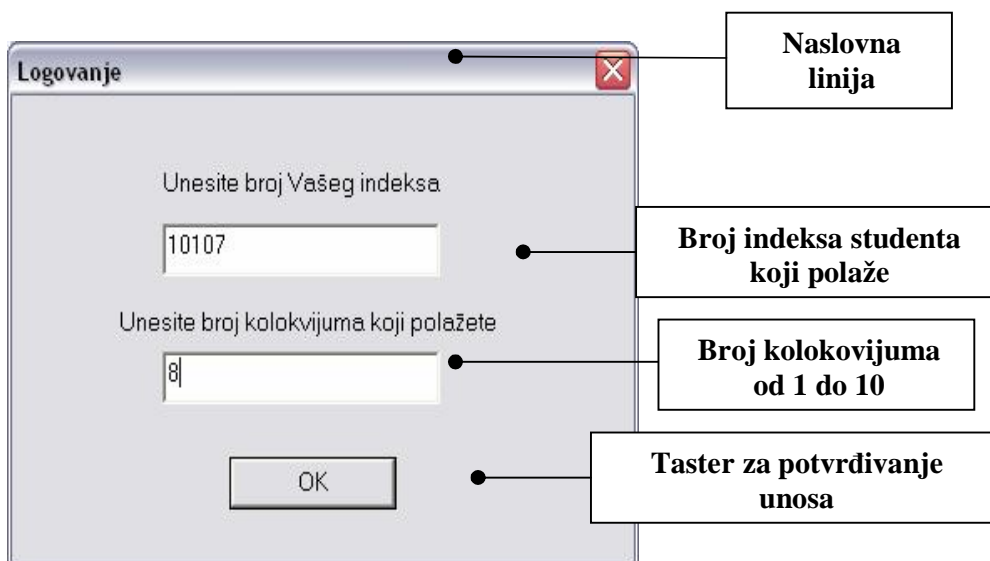
Pomoću tabele koja se ovom prilikom otvara može se dobiti informacijama o broju poena koje je student osvojio pri svakom izlasku na kolokvijum. Maksimalan broj kolokvijuma je 10, a maksimalan broj izlazaka na pojedini kolokvijum je 3.

### Postupak provere znanja studenata

Da bi se pokrenuo test za proveru znanja studenata treba kliknuti dvaput na prečicu **KOLOKVIJUM**. Posle nekoliko trenutaka pred vama će se otvoriti prozor kao na slici. Prvi prozor daje informaciju studentu o ispitu koji polaže i šta ga čeka. Tasterom **ULAZ** idemo na radno okruženje za studente. Iako deluje prilično neugledno, imajte na umu da je ovo moćan program čija je namena da radi u "pozadini" a da je pred vama samo ogoljeni okvir. U njemu možete da uočite nekoliko celina od kojih svaka ima važnu ulogu.

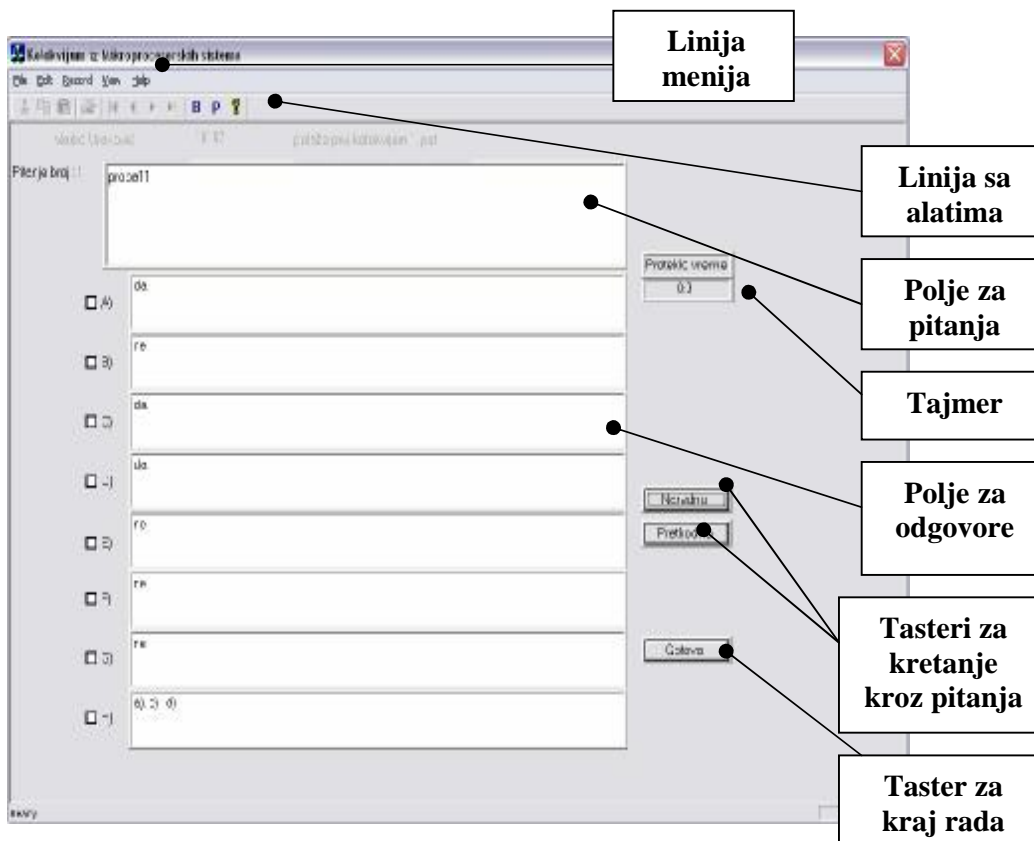


Slika 10 Izgled prozora start



Slika 11 Izgled prozora logovanje za studente

Kada se unese što je potrebno od strane studenta otvara se prozor u kom student može krenuti sa odgovaranjem na pitanja kao na slici 12. U suprotnom student neće dobiti dozvolu za rad.



Slika 12 Izgled radnog okruženja za studente

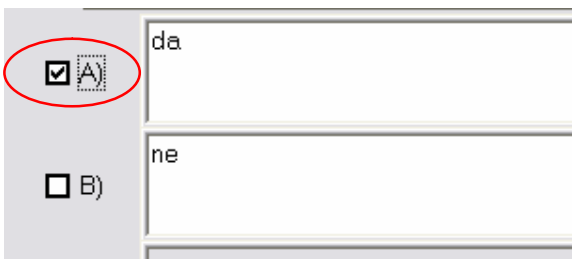
Detaljno upoznavanje sa svakom od njih će vam umnogome olakšati rad i zato krenimo redom.

- **Naslovna linija** nam govori u kom delu programa se trenutno nalazimo.
- **Linija sa alatima.** U ovoj liniji su ostavljena dva tastera **B** i **P** koja se mogu dodatno isprogramirati za određene potrebe. Tu je i taster **?** u kom se nalaze informacije o programu

Marko Uraković 10107 polaže prvi kolokvijum 1. put

Polje koje je istaknuto pokazuje ime i prezime studenta, broj indeksa i koji kolikvijum koji put polaže. Maksimalan broj izlazaka na jedan kolokvijum je tri puta.

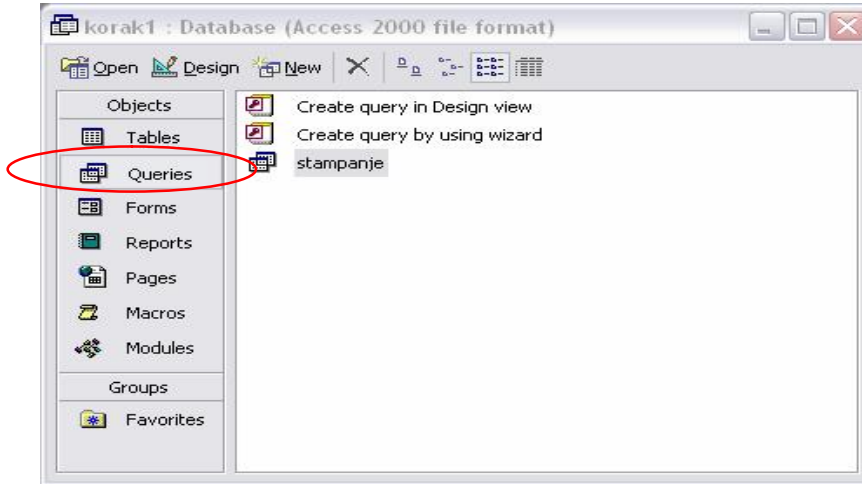
- **Tajmer** – pokazuje koliko je vremena prošlo od početka polaganja. Polaganje traje **15 minuta** posle čega više nije moguće odgovarati na pitanja.
- **Naredno i Prethodno** su tasteri koji omogućavaju kretanje kroz pitanja.



-Na slici se vidi deo polja sa odgovorima. Može se izabrati samo jedan odgovor tj. samo jedno slovo može biti obeleženo.


## Generisanje izveštaja

U objektu **Queries** se nalazi ikonica **štampanje** koja nam daje sređeni pregled rezultata koje je student postigao na svakom od kolokvijuma.



Slika 13 Izgled Database (Access 2000 file format) sa slektovanim objektom *Queries*

Primer tabele sa prikazom rezultata koje su studenti postigli na kolokvijumima imamo na slici 14 .



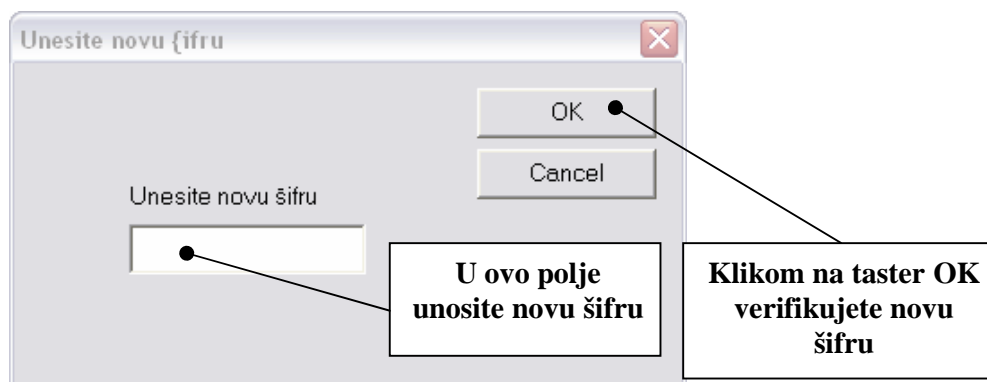
The screenshot shows the Microsoft Access 2000 interface with a table named 'štampanje' displayed in 'Select Query' view. A callout box labeled 'Taster za štampanje' points to the printer icon in the toolbar. The table contains the following data:

student	bitn dexa	kol1-1	kol2-1	kol3-1	kol4-1	kol5-1	kol6-1	kol7-1	kol8-1	kol9-1	kol10-1	ukupno
Dalibor Bogdanović	9860											
Aleksandar Dinić	9890											
Branimir Milić	9993											
Marko Uraković	10107	8	6									7.00
Ivan Stojanović	10666											
Velibor Mijatović	11397											
Igor Marković	11705		3									3.00
*	0											0.00

Slika 14 Izgled konačne tabele za štampanje

Na kraju treba naglasiti da se pri štampanju u okviru kolona za pojedine kolokvijume pojavljuje maksimalan broj poena koje je student osvojio na tom kolokvijumu, dok je poslednja kolona namenjena za njihovu aritmetičku sredinu .U slučaju da je student na nekom od kolokvijuma osvojio najviše 0 ili neki negativan broj poena program za maksimum uzima 0.

Ukoliko želite promenu šifre za pristup predhodno pomenutim bazama otvoriće Vam se sledeći prozor:



Slika 15 Izgled prozora za promenu šifre

Šifra može sadržati najviše 6, a najmanje 1 karakter. Karakteri mogu biti i slovnog i brojanog tipa.

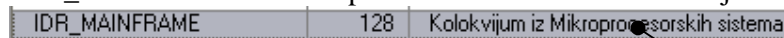
### Promena naslova predmeta iz koga se vrši provera znanja

Kako je kolokvijum rađen za ispit iz mikroprocesorskih sistema, za primenu na druge ispite bilo bi poželjno promeniti nazive u start prozoru i u radnom okruženju kako se student ne bi zbunio. Promena se radi na sledeći način.

U direktorijumu kolokvijum treba pronaći fajl **korak2.dsw** (Project Workspace).

Da biste njega otvorili morate imati instaliran **Visual Studio 6**. Posle otvaranja ovog fajla na levoj strani pri dnu prozora treba izabrati tab **ResourceView**, zatim izaberite Dialog a onda dvoklik na **IDD\_DIALOG1** i pojaviće se prvi prozor gde se piše ime predmeta.

Kliknite desnim klikom na ime predmeta i izaberite **properties** i u polju **caption** obrišite ime koje piše i upišite ono koje želite. Dok za **IDD\_KORAK2\_FORM** pored navedenog postupka treba još ući na **string table**. Sada na desnoj strani pod ID-om **IDR\_MAINFRAME** treba promeniti naziv kao na sledećoj slici.



Polje gde treba promeniti naziv

Sada je sve lako i poznato pa se promena naziva vrši na način koji je već opisan. Posle ovoga je potrebno program ponovo iskompajlirati da bi sve nove izmene proradile.

## KOMENTARI

Sve navedene klase imaju interfejsu u odgovarajućim .h fajlovima.

### Dodatak 1

▼ Ckorak2Set je klasa za povezivanje tabele sa pitanjima i programa.

```
// korak2Set.h : interface of the CKorak2Set class
//
///////////////////////////////////////////////////////////////////

#ifndef AFX_KORAK2SET_H__8537A376_CDA6_492D_B6B3_644C5C0EE578__INCLUDED_
#define AFX_KORAK2SET_H__857A376_CDA6_492D_B6B3_644C5C0EE578__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CKorak2Set : public CRecordset
{
public:
    CKorak2Set(CDatabase* pDatabase = NULL);
    DECLARE_DYNAMIC(CKorak2Set)

// Field/Param Data
    //{ AFX_FIELD(CKorak2Set, CRecordset)
    long m_ID;
    long m_broj;
    long m_kolokvijum;
    long m_broj_odgovora;
    CString m_pitanje;
    CString m_odgovor0;
    CString m_odgovor1;
    CString m_odgovor2;
    CString m_odgovor3;
    CString m_odgovor4;
    CString m_odgovor5;
    CString m_odgovor6;
    CString m_odgovor7;
    //}AFX_FIELD

// Overrides
    // ClassWizard generated virtual function overrides
    //{ AFX_VIRTUAL(CKorak2Set)
    public:
    virtual CString GetDefaultConnect(); // Default connection string
    virtual CString GetDefaultSQL(); // default SQL for Recordset
    virtual void DoFieldExchange(CFieldExchange* pFX); // RFX support
    //}AFX_VIRTUAL

// Implementation
#ifdef _DEBUG
```



```

        virtual void AssertValid() const;
        virtual void Dump(CDumpContext& dc) const;
    #endif
};
//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif//!defined(AFX_KORAK2SET_H_8537A376_CDA6_492D_B6B3_644C5C0EE578__INCLUD
ED_)

```

## Dodatak 2

### ▼ CAboutDlg je klasa dijaloga About

```

// korak2.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "korak2.h"

#include "MainFrm.h"
#include "korak2Set.h"
#include "korak2Doc.h"
#include "korak2View.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////
// CKorak2App

BEGIN_MESSAGE_MAP(CKorak2App, CWinApp)
//{{AFX_MSG_MAP(CKorak2App)
    ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
        // NOTE - the ClassWizard will add and remove mapping macros here.
        // DO NOT EDIT what you see in these blocks of generated code!
    }}AFX_MSG_MAP
    // Standard print setup command
    ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////
// CKorak2App construction

CKorak2App::CKorak2App()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

/////////////////////////////////////////////////////////////////
// The one and only CKorak2App object

CKorak2App theApp;

/////////////////////////////////////////////////////////////////
// CKorak2App initialization

```

```

BOOL CKorak2App::InitInstance()
{
    AfxEnableControlContainer();

    // Standard initialization
    // If you are not using these features and wish to reduce the size
    // of your final executable, you should remove from the following
    // the specific initialization routines you do not need.

#ifdef _AFXDLL
    Enable3dControls(); // Call this when using MFC in a shared DLL
#else
    Enable3dControlsStatic(); // Call this when linking to MFC statically
#endif

    // Change the registry key under which our settings are stored.
    // TODO: You should modify this string to be something appropriate
    // such as the name of your company or organization.
    SetRegistryKey(_T("Local AppWizard-Generated Applications"));

    LoadStdProfileSettings(); // Load standard INI file options (including MRU)

    // Register the application's document templates. Document templates
    // serve as the connection between documents, frame windows and views.

    CSingleDocTemplate* pDocTemplate;
    pDocTemplate = new CSingleDocTemplate(
        IDR_MAINFRAME,
        RUNTIME_CLASS(CKorak2Doc),
        RUNTIME_CLASS(CMainFrame), // main SDI frame window
        RUNTIME_CLASS(CKorak2View));
    AddDocTemplate(pDocTemplate);

    // Parse command line for standard shell commands, DDE, file open
    CCommandLineInfo cmdInfo;
    ParseCommandLine(cmdInfo);

    // Dispatch commands specified on the command line
    if (!ProcessShellCommand(cmdInfo))
        return FALSE;

    // The one and only window has been initialized, so show and update it.
    m_pMainWnd->ShowWindow(SW_SHOW);
    m_pMainWnd->UpdateWindow();

    return TRUE;
}

////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

    // Dialog Data
   //{{AFX_DATA(CAboutDlg)

```

```

enum { IDD = IDD_ABOUTBOX };
//{{AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
// No message handlers
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
//{{AFX_DATA_INIT(CAboutDlg)
//}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
CDialog::DoDataExchange(pDX);
//{{AFX_DATA_MAP(CAboutDlg)
//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
//{{AFX_MSG_MAP(CAboutDlg)
// No message handlers
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

// App command to run the dialog
void CKorak2App::OnAppAbout()
{
CAboutDlg aboutDlg;
aboutDlg.DoModal();
}

/////////////////////////////////////////////////////////////////
// CKorak2App message handlers

```

## Dodatak 3

### ▼ CDialogLogg je klasa dijaloga za logovanje

```
#if
!defined(AFX_DIALOGLOGG_H__07AB3863_F990_430A_8BE3_6F67036F9F76__INCLUDE
D_)
#define
AFX_DIALOGLOGG_H__07AB3863_F990_430A_8BE3_6F67036F9F76__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// DialogLogg.h : header file
//

/////////////////////////////////////////////////////////////////
// CDialogLogg dialog

class CDialogLogg : public CDialog
{
// Construction
public:
    CDialogLogg(CWnd* pParent = NULL); // standard constructor

// Dialog Data
   //{{AFX_DATA(CDialogLogg)
    enum { IDD = IDD_DIALOG_LOG };
    int         m_logBrojIndeksa;
    int         m_brkolokvijuma;
    //}}AFX_DATA

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CDialogLogg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:

    // Generated message map functions
    //{{AFX_MSG(CDialogLogg)
        // NOTE: the ClassWizard will add member functions here
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif //
!defined(AFX_DIALOGLOGG_H__07AB3863_F990_430A_8BE3_6F67036F9F76__INCLUDE
D_)
```

## Dodatak 4

### ✓ CKorak2App je klasa CWinApp za čitavu aplikaciju

```
// korak2.h : main header file for the KORAK2 application
//

#if
!defined(AFX_KORAK2_H_BC46CED6_B512_4868_8BD6_3D834A26576C__INCLUDED_)
#define AFX_KORAK2_H_BC46CED6_B512_4868_8BD6_3D834A26576C__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#ifndef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h"    // main symbols

/////////////////////////////////////////////////////////////////
// CKorak2App:
// See korak2.cpp for the implementation of this class
//

class CKorak2App : public CWinApp
{
public:
    CKorak2App();

// Overrides
    // ClassWizard generated virtual function overrides
   //{{AFX_VIRTUAL(CKorak2App)
public:
    virtual BOOL InitInstance();
   //}}AFX_VIRTUAL

// Implementation
   //{{AFX_MSG(CKorak2App)
afx_msg void OnAppAbout();
        // NOTE - the ClassWizard will add and remove member functions here.
        // DO NOT EDIT what you see in these blocks of generated code !
   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif //
!defined(AFX_KORAK2_H_BC46CED6_B512_4868_8BD6_3D834A26576C__INCLUDED_)
```

## Dodatak 4.1

### ▼ CKorak2Set2 je klasa koja povezuje tabelu studenti i program

```
#ifndef AFX_KORAK2SET2_H_5DD7704C_B969_4C44_8B91_CC74B2957A7A_INCLUDED_
)
#define AFX_KORAK2SET2_H_5DD7704C_B969_4C44_8B91_CC74B2957A7A_INCLUDED_

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// Korak2Set2.h : header file
//

/////////////////////////////////////////////////////////////////
// CKorak2Set2 recordset

class CKorak2Set2 : public CRecordset
{
public:
    CKorak2Set2(CDatabase* pDatabase = NULL);
    DECLARE_DYNAMIC(CKorak2Set2)

// Field/Param Data
//{{AFX_FIELD(CKorak2Set2, CRecordset)
    CString m_student;
    long m_brIn_dexa;
    long m_kol1_1;
    long m_kol1_2;
    long m_kol1_3;
    long m_kol2_1;
    long m_kol2_2;
    long m_kol2_3;
    long m_kol3_1;
    long m_kol3_2;
    long m_kol3_3;
    long m_kol4_1;
    long m_kol4_2;
    long m_kol4_3;
    long m_kol5_1;
    long m_kol5_2;
    long m_kol5_3;
    long m_kol6_1;
    long m_kol6_2;
    long m_kol6_3;
    long m_kol7_1;
    long m_kol7_2;
    long m_kol7_3;
    long m_kol8_1;
    long m_kol8_2;
    long m_kol8_3;
    long m_kol9_1;
    long m_kol9_2;
    long m_kol9_3;
    long m_kol10_1;
    long m_kol10_2;
    long m_kol10_3;
    double ukupno;
//}}AFX_FIELD
```

```

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CKorak2Set2)
public:
virtual CString GetDefaultConnect(); // Default connection string
virtual CString GetDefaultSQL(); // Default SQL for Recordset
virtual void DoFieldExchange(CFieldExchange* pFX); // RFX support
//}}AFX_VIRTUAL

// Implementation
#ifdef _DEBUG
virtual void AssertValid() const;
virtual void Dump(CDumpContext& dc) const;
#endif
};
//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX_KORAK2SET2_H_5DD7704C_B969_4C44_8B91_CC74B2957A7A_INCLU
DED_)

```

## Dodatak 5

- ▼ **CKorak2View je osnovna klasa za izvršenje programa i ona je odgovorna za inicijalizaciju, rad i prikaz programa**

```

// korak2View.h : interface of the CKorak2View class
//
////////////////////////////////////////////////////////////////////

#ifndef AFX_KORAK2VIEW_H_47ACE489_C3D9_496C_94B1_B40C8307AA18_INCLUDED_
#define AFX_KORAK2VIEW_H_47ACE489_C3D9_496C_94B1_B40C8307AA18_INCLUDED_

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CKorak2Set;
class CKorak2Set2;
class ILista;

class CKorak2View : public CRecordView
{
protected: // create from serialization only
    CKorak2View();
    DECLARE_DYNCREATE(CKorak2View)

public:
    {{{AFX_DATA(CKorak2View)
    enum { IDD = IDD_KORAK2_FORM };
    CKorak2Set* m_pSet;
    CKorak2Set2* m_pSet2;
    CString m_edit1;
    CString m_edit2;
    CString m_edit3;
    CString m_edit4;
    CString m_edit5;
    }}}

```

```

CString m_edit6;
CString m_edit7;
CString m_edit8;
CString m_editPitanje;
BOOL m_checkA;
BOOL m_checkB;
BOOL m_checkC;
BOOL m_checkD;
BOOL m_checkE;
BOOL m_checkF;
BOOL m_checkG;
BOOL m_checkH;
CString m_sTime;
CString m_text;
CString m_student;
int m_index;
int m_pit;
//}}AFX_DATA

// Attributes
public:
    CKorak2Doc* GetDocument();

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
   //{{AFX_VIRTUAL(CKorak2View)
public:
    virtual CRecordset* OnGetRecordset();
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    virtual void OnInitialUpdate(); // called first time after construct
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
    //}}AFX_VIRTUAL

// Implementation
public:
    void Exit();
    int PrebaciNazad3(ILista& t);
    int prompass;
    int test;
    int unos;
    int prof;
    void PrebaciNazad2(ILista &t,int zad);
    int brojKolokvijuma;
    int brojIndexa;
    int zatammiOd;
    void Zasivi(int i);
    int PrebaciNazad(ILista& t,int zad);
    int brojBodova;
    void OnCheck();
    CString izmjeniOdgovor(CString pom,ILista& t);
    virtual ~CKorak2View();
    CDatabase db;
    int prviPut;
    int brPitanjaKol;
    int brPitanjaKol1;

```



```

        int brPitanjaKol2;
        int brPitanjaKol3;
        int brPitanjaKol4;
        int brPitanjaKol5;
        int brPitanjaKol6;
        int brPitanjaKol7;
        int brPitanjaKol8;
        int brPitanjaKol9;
        int brPitanjaKol10;
        int brPitanjaSviKol;
#ifdef _DEBUG
        virtual void AssertValid() const;
        virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
    //{ AFX_MSG(CKorak2View)
    afx_msg void OnButtonNaredno();
    afx_msg void OnButtonPrethodno();
    afx_msg void OnCheckA();
    afx_msg void OnCheckB();
    afx_msg void OnCheckC();
    afx_msg void OnCheckD();
    afx_msg void OnCheckE();
    afx_msg void OnCheckF();
    afx_msg void OnCheckG();
    afx_msg void OnCheckH();
    afx_msg void OnButtonGotovo();
    afx_msg void OnTimer(UINT nIDEvent);
    afx_msg void OnButton1();
    afx_msg void OnButton3prof();
    afx_msg void OnButtonpass();
    afx_msg void OnButton2();
    afx_msg void OnUpdateAppExit(CCmdUI* pCmdUI);
    //}AFX_MSG
    DECLARE_MESSAGE_MAP()
    CString vratiOdgovor(int n);
    CString odgovori[8];
    int brojOdgovora;
    int brojPitanja;
    int rasporedPitanja[10];
    int datiOdgovori[10];
    void Promjesaj(ILista& t);
    inline bool PostojiLi(int n,ILista& t);
    inline bool PostojiLi(int n);
    ILista* glava;
    ILista* tekuce;
    int minuti;
    int sekunde;
    bool dozvolaZaRad;
};

#ifdef _DEBUG // debug version in korak2View.cpp
inline CKorak2Doc* CKorak2View::GetDocument()
    { return (CKorak2Doc*)m_pDocument; }
#endif

```

```

////////////////////////////////////
//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif!defined(AFX_KORAK2VIEW_H_47ACE489_C3D9_496C_94B1_B40C8307AA18_INCLU
DED_)

```

## Dodatak 6

### ▼ ILista je klasa koja se koristi za formiranje lančanih lista

```

class ILista
{
public:
    CString pitanje;
    CString odgovori[8];
    int rasporedOdgovora[8];
    int brojOdgovora;
    int brojPitanja;
    ILista* ILink;
    ILista* dLink;
};

```

## Dodatak 7

### ▼ CDialogLogg je klasa dijaloga za logovanje

```

#ifndef AFX_DIALOGLOGG_H_07AB3863_F990_430A_8BE3_6F67036F9F76_INCLUDED_
#define AFX_DIALOGLOGG_H_07AB3863_F990_430A_8BE3_6F67036F9F76_INCLUDED_

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// DialogLogg.h : header file
//

////////////////////////////////////
// CDialogLogg dialog

class CDialogLogg : public CDialog
{
// Construction
public:
    CDialogLogg(CWnd* pParent = NULL); // standard constructor

// Dialog Data
//{{AFX_DATA(CDialogLogg)
enum { IDD = IDD_DIALOG_LOG };

```

```

        int            m_logBrojIndeksa;
        int            m_brkolokvijuma;
    //}}AFX_DATA

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CDialogLogg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:

    // Generated message map functions
    //{AFX_MSG(CDialogLogg)
        // NOTE: the ClassWizard will add member functions here
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif//!defined(AFX_DIALOGLOGG_H_07AB3863_F990_430A_8BE3_6F67036F9F76_INCLUD
ED_)

```

## Dodatak 8

### ▼ CPrviProzor je klasa prvog prozora

```

#ifndef AFX_PRVIPROZOR_H_81A61BE4_403D_4F3E_BB8D_3A07F64CDC86_INCLUDED
_
#define AFX_PRVIPROZOR_H_81A61BE4_403D_4F3E_BB8D_3A07F64CDC86_INCLUDED_

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// PrviProzor.h : header file
//

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CPrviProzor dialog

class CPrviProzor : public CDialog
{
// Construction
public:
    CPrviProzor(CWnd* pParent = NULL); // standard constructor

// Dialog Data
    //{AFX_DATA(CPrviProzor)
        enum { IDD = IDD_DIALOG1 };
        // NOTE: the ClassWizard will add data members here
    //}}AFX_DATA

// Overrides

```

```

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CPrviProzor)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:

// Generated message map functions
//{{AFX_MSG(CPrviProzor)
// NOTE: the ClassWizard will add member functions here
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif!defined(AFX_PRVIPROZOR_H_81A61BE4_403D_4F3E_BB8D_3A07F64CDC86_INCLUD
ED_)

```

## Dodatak 9

### ▼ Cprofesor je klasa dijaloga koji koristi profesor

```

#ifndef AFX_PROFESOR_H_BD197E60_6523_43EA_82FF_6E1BF4244E4D_INCLUDED_
#define AFX_PROFESOR_H_BD197E60_6523_43EA_82FF_6E1BF4244E4D_INCLUDED_

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// Profesor.h : header file
//

////////////////////////////////////

// CProfesor dialog

class CProfesor : public CDialog
{
// Construction
public:
    CProfesor(CWnd* pParent = NULL); // standard constructor

// Dialog Data
//{{AFX_DATA(CProfesor)
enum { IDD = IDD_DIALOG2 };
CString m_prof_pass;
int m_brKol;
//}}AFX_DATA

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CProfesor)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

```

```

        //}}AFX_VIRTUAL

// Implementation
protected:

        // Generated message map functions
        //{{AFX_MSG(CProfesor)
        afx_msg void OnButton1();
        afx_msg void OnButton2();
        afx_msg void OnButton3();
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()

public:
        int test;
        int unos;
        int prof;
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif!defined(AFX_PROFESOR_H_BD197E60_6523_43EA_82FF_6E1BF4244E4D_INCLUDED_)

```

## Dodatak 10

### ▼ Cputanja ja klasa dijaloga za unos putanje do Access-a

```

#ifndef AFX_PUTANJA_H__5A218690_837F_4EDD_97E9_9397EB5DF126__INCLUDED_
#define AFX_PUTANJA_H__5A218690_837F_4EDD_97E9_9397EB5DF126__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// Putanja.h : header file
//

////////////////////////////////////
// CPutanja dialog

class CPutanja : public CDialog
{
// Construction
public:
        CPutanja(CWnd* pParent = NULL); // standard constructor

// Dialog Data
        //{{AFX_DATA(CPutanja)
        enum { IDD = IDD_DIALOG_PUTAJNA };
        CString m_putanja;
        //}}AFX_DATA

// Overrides
        // ClassWizard generated virtual function overrides
        //{{AFX_VIRTUAL(CPutanja)
        protected:
        virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
        //}}AFX_VIRTUAL

```

```

// Implementation
protected:

    // Generated message map functions
    //{ {AFX_MSG(CPutanja)
        // NOTE: the ClassWizard will add member functions here
    //} }AFX_MSG
    DECLARE_MESSAGE_MAP()
};

//{ {AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif //
!defined(AFX_PUTANJA_H__5A218690_837F_4EDD_97E9_9397EB5DF126__INCLUDED_)

```

## Dodatak 11

### ▼ Cprofesor je klasa dijaloga koji koristi professor

```

#if !defined(AFX_PROFESOR_H__BD197E60_6523_43EA_82FF_6E1BF4244E4D__INCLUDED_)
#define AFX_PROFESOR_H__BD197E60_6523_43EA_82FF_6E1BF4244E4D__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// Profesor.h : header file
//

//////////////////////////////////////
// CProfesor dialog

class CProfesor : public CDialog
{
// Construction
public:
    CProfesor(CWnd* pParent = NULL); // standard constructor

// Dialog Data
    //{ {AFX_DATA(CProfesor)
        enum { IDD = IDD_DIALOG2 };
        CString m_prof_pass;
        int m_brKol;
    //} }AFX_DATA

// Overrides
    // ClassWizard generated virtual function overrides
    //{ {AFX_VIRTUAL(CProfesor)
        protected:
        virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    //} }AFX_VIRTUAL

// Implementation
protected:

    // Generated message map functions
    //{ {AFX_MSG(CProfesor)

```

```

        afx_msg void OnButton1();
        afx_msg void OnButton2();
        afx_msg void OnButton3();
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
public:
        int test;
        int unos;
        int prof;
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif //
!defined(AFX_PROFESOR_H__BD197E60_6523_43EA_82FF_6E1BF4244E4D__INCLUDED_)

```

## Dodatak 12

### ▼ CnovaSifra je klasa dijaloga za promenu sifre

```

#if
!defined(AFX_NOVASIFRA_H__0F967533_F1D6_4CD4_B47B_1F6205A70ABC__INCLUDED_)
#define AFX_NOVASIFRA_H__0F967533_F1D6_4CD4_B47B_1F6205A70ABC__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// NovaSifra.h : header file
//

//////////////////////////////////////
// CNovaSifra dialog

class CNovaSifra : public CDialog
{
// Construction
public:
        CNovaSifra(CWnd* pParent = NULL); // standard constructor

// Dialog Data
        //{AFX_DATA(CNovaSifra)
        enum { IDD = IDD_DIALOG3 };
        CString m_nSifra;
        //}}AFX_DATA

// Overrides
        // ClassWizard generated virtual function overrides
        //{AFX_VIRTUAL(CNovaSifra)
        protected:
        virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
        //}}AFX_VIRTUAL

// Implementation
protected:

        // Generated message map functions
        //{AFX_MSG(CNovaSifra)

```

```

        // NOTE: the ClassWizard will add member functions here
    //{AFX_MSG
    DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif //
!defined(AFX_NOVASIFRA_H__0F967533_F1D6_4CD4_B47B_1F6205A70ABC__INCLUDED_)

```

## Dodatak 13

### ▼ CMainFrame je klasa okvira

```

// MainFrm.h : interface of the CMainFrame class
//
////////////////////////////////////////////////////////////////////

#if !defined(AFX_MAINFRM_H__FD42BFE8_ED93_4239_B3B7_AA902075656C__INCLUDED_)
#define AFX_MAINFRM_H__FD42BFE8_ED93_4239_B3B7_AA902075656C__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CMainFrame : public CFrameWnd
{

protected: // create from serialization only
    CMainFrame();
    DECLARE_DYNCREATE(CMainFrame)

// Attributes
public:

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CMainFrame)
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    //}AFX_VIRTUAL

// Implementation
public:
    virtual ~CMainFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected: // control bar embedded members

```



```

        CStatusBar m_wndStatusBar;
        CToolBar m_wndToolBar;

// Generated message map functions
protected:
   //{{AFX_MSG(CMainFrame)
   afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);

   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif //
!defined(AFX_MAINFRM_H__FD42BFE8_ED93_4239_B3B7_AA902075656C__INCLUDED_)

```