

Univerzitet Džemal Bijedić Mostar

Fakultet informacionih tehnologija Mostar

Mostar, 19.04.2010

## **UPR – Uvod u programiranje**

***Seminarski rad***

Student: **Husremović Ernad, DL 2792**

## **1. Pozadina**

Student je položio ispit “Programiranje 1” na ETF-u Sarajevo. Na tom predmetu je korišten programski jezik Pascal.

Svrha ovog seminarskog je demonstracija vještina u korištenju programskog jezika C/C++.

Seminarski treba da sadrži praktični rad po izboru studenta, koji će obuhvatiti sve programske strukture i elemente koji se obrađuju na ovom predmetu.

## **2. Uvod**

Iskreno rečeno, najteži dio posla za mene je odabir programskog zadatka koji će zadovoljiti zahtjeve nastavnika, a ujedno predstavljati neki smislen programski kod :(.

Kako se moja firma bavi izradom poslovnih programskih rješenja, odlučio sam da odaberem neki segment iz naših aplikacija i napravim ga u C/C++.

Odabrao sam izradu – štampu porezne fakture, naravno u pojednostavljenoj formi :).

### **3. Programski zadatak**

Napraviti program koji će na osnovu predefinisano seta (statički šifrnici) ulaznih podataka generisati poreznu fakturu.

#### **3.1 Ulazni podaci**

Korisnik će imati na izbor odabir sljedećih predefinisanih podataka:

##### **3.1.1 Partneri (Kupci):**

1. P1 – Rama-glas Sarajevo
2. P2 – Finkon Sarajevo
3. P3 – Agencija GM Konjic

##### **3.1.2 Dobra (roba/usluge) koja se fakturišu kupcima:**

1. D1 – Programerska aktivnost – 90 KM/h (cijena bez PDV)
2. D2 – System admin aktivnost - 60 KM/h
3. D3 – Servisna aktivnost – 50 KM/h

##### **3.1.3 Porezna stopa PDV = 17%**

#### **3.2 User story**

Definišimo uobičajeni tok aplikacije korištenja (application workflow);

##### **3.2.1) Odabir partnera:**

Korisnik putem menija odabire jednog od kupaca iz šifrnika 3.1.1

##### **3.2. 2) Unos stavki fakture:**

Nakon odabira kupca unosi stavke fakture i to: odabir dobra putem menija (šifrnika 3.1.2), te količinu.

Ako korisnik unese 10 stavki, program prekida (max stavki fakture).

Ako korisnik unese 0 prilikom odabira dobra, program takođe prekida rad.

##### **3.2.3) Prikaz fakture.**

Nakon toga program vrši obračun stavki:

- Ukupno bez PDV
- PDV
- SVEUKUPNO Sa PDV

### 3.2.4 Prikaz fakture - primjer

Kao rezultat dobijamo artifakt – tekstualnu fakturu kao na slici:

```
=====
Naziv firme: bring.out. doo Sarajevo
Adresa:      Juraja Najtharta 3, 71000 Sarajevo
web:         http://www.bring.out.ba
-----
Datum: 2010-04-19
=====
Kupac: Finkon Sarajevo
=====
R.br      Dobro      Kolicina  Cijena  Vrijednost
-----
1. Programerske aktivnosti      2.30    90.00    207.00
2. Servisne aktivnosti          4.80    50.00    240.00
-----
Ukupno BEZ PDV (KM):            447.00
PDV (KM):                        75.99
SVEUKUPNO SA PDV (KM):          522.99
=====
```

## **4. Testiranje**

Sa ruby interaktivnim alatom ručno provjerena je provjerena matematička ispravnost programa:

```
irb(main):001:0> (2.3*90+4.80*50)*1.17
=> 522.99
```

## **5. Reference**

1. *Beginning ANSI C++, Ivor Horton, Apress 2004*
2. *Google pretraga/internet sadržaji*
  - *content.hccfl.edu/pollock/CPlus/IOStream.doc*
  - <http://www.cplusplus.com/reference/cstring/strcmp/>

## **6. Radno okruženje**

- Radna stanica OS: Ubuntu desktop 10.04
- Aplikacije:
  - Openoffice 3.2
  - Eclipse 3.5.2
  - vi editor
- compiler: gcc 4.4.3
- ruby 1.8.7 sa irb utility-jem

## **7. artifakti na gitorious.org**

Na lokaciji nalaze se svi artifakti ovog rada: <http://gitorious.org/fit-mostar/upr/trees/master/seminarski>

## **8. Source kod**

Na lokaciji <http://gitorious.org/fit-mostar/upr/blobs/master/seminarski/fakturisi.cpp> nalazi se puno pregledniji prikaz source koda:

```
===== početak =====
#include <iostream>

// setw ...
#include <iomanip>

#include <stdio.h>

// system cmd je definisan u stdlib-u
#include <stdlib.h>

// strcmp
#include <string.h>

#include <math.h>

using namespace std;

// ----- begin konstante -----

// sirina terminala
const int COL_COUNT = 80;
const char LINIJA_DEBELA = '=';
const char LINIJA_TANKA = '-';

const char *FIRMA = "bring.out. doo Sarajevo";
const char *FIRMA_WEB = "http://www.bring.out.ba";
const char *FIRMA_ADRESA = "Juraja Najtharta 3, 71000 Sarajevo";

const int KUPCI_LENGTH = 3;

// sifrarnik kupaca
const char *A_KUPCI_NAZ[] = {"Rama-glas Sarajevo", "Finkon Sarajevo", "Agencija GM Konjic"};

const int DOBRA_LENGTH = 3;

// Napomena:
// sifrarnik dobara je predstavljen sa dvije matrice
// bilo bi puno bolje koristiti napraviti struct Dobro, u kome bi se u jednoj matrici
// sve to stavilo.
// Međutim, u UPR-u do te lekcije nismo dosli :).

const char *A_DOBRA_NAZ[] = {
    "Programerske aktivnosti",
    "Sysadmin aktivnosti",
    "Servisne aktivnosti"
};

const double A_DOBRA_CIJ[] = {
    90.00,
    70.00,
    50.00
};

// dole je definisana a_faktura - 3 - dim matrica.
// red predstavlja stavku fakture
```

```

// kolona 1: kolicina dobara
// kolona 2: id dobra
// kolona 3: vrijednost stavke fakture

// maksimalno stavki na jednoj fakturi
const int FAKTURA_MAX_STAVKI = 10;

const int FAKT_COL_KOL = 0;
const int FAKT_COL_ID_DOBRA = 1;
const int FAKT_COL_ID_VRIJEDNOST = 2;

const double PDV_STOPA = 17.00;

// ----- end konstante -----

// ----- begin prototype -----
void podvuci(const char linija);
void show_danasnji_datum();
void show_firma_podaci();
// ----- end prototype -----

double aFaktura[FAKTURA_MAX_STAVKI][3];

void show_danasnji_datum() {
/* 18 znakova ovaj string uzima
Datum: 2010-15-12
*/

// hocu dad postignem desno poravnanje
for (int i=1; i < (COL_COUNT-18); i++ )
    cout << " ";

cout << "Datum: ";
// flushiraj cout buffer da bi se Datum: prikazalo prije samog datuma
flush(cout);
system("date +%F");

}

void show_firma_podaci()
{
    cout << "Naziv firme: " << FIRMA << endl;
    cout << "Adresa:      " << FIRMA_ADRESA << endl;
    cout << "web:          " << FIRMA_WEB << endl;

}

void show_zaglavljje()
{
    podvuci(LINIJA_DEBELA);
    show_firma_podaci();

    podvuci(LINIJA_TANKA);
    show_danasnji_datum();
    podvuci(LINIJA_DEBELA);

}

const char *get_kupac_naziv(int id_kupac)

```

```

{
    // id_dobro = 1 => daj prvi elemenat matrice !
    return A_KUPCI_NAZ[id_kupac-1];
}

const char *get_dobro_naziv(int id_dobro)
{
    // id_dobro = 1 => daj prvi elemenat matrice !
    return A_DOBRA_NAZ[id_dobro-1];
}

const double get_dobro_cijena(int id_dobro)
{
    return A_DOBRA_CIJ[id_dobro-1];
}

int meni_odaberi_kupca()
{
    int rez;
    podvuci(LINIJA_DEBELA);

    podvuci(LINIJA_TANKA);

    // http://www.cs.duke.edu/courses/cps149s/fall99/resources/n2.html
    // setw( width ) : gives a minimum width for the next output operation
    // ovdje sam citav sahat ostao dok nisam nekako dodo ovaj resetiosflags
    // koji mi je pomogao da red formatiram kako zelim ?!
    // je li ovo gcc/libc++ bug ili nesto drugo ne znam

    for (int i=1; i<=KUPCI_LENGTH; i++)
        cout << setiosflags(ios::right) << setw(5) << i
            << resetiosflags(ios::right) << setw(0) << " "
            << setiosflags(ios::left) << setw(2) << "."
            << setiosflags(ios::left) << setw(50) << get_kupac_naziv(i)
            << endl ;

    podvuci(LINIJA_DEBELA);

    rez = 0;

    do {
        cout << "Odaberite zeljenog kupca (unijeti broj): " ;
        cin >> rez;
    } while( (rez<1) || (rez>KUPCI_LENGTH));

    return rez;
}

int meni_odaberi_dobro()
{
    int rez;
    podvuci(LINIJA_DEBELA);

    podvuci(LINIJA_TANKA);

    // http://www.cs.duke.edu/courses/cps149s/fall99/resources/n2.html
    // setw( width ) : gives a minimum width for the next output operation
    // ovdje sam citav sahat ostao dok nisam nekako dodo ovaj resetiosflags
    // koji mi je pomogao da red formatiram kako zelim ?!
    // je li ovo gcc/libc++ bug ili nesto drugo ne znam

```



```

for (int i=1; i<=DOBRA_LENGTH; i++)
    cout << setiosflags(ios::right) << setw(5) << i
    << resetiosflags(ios::right) << setw(0) << ""
    << setiosflags(ios::left) << setw(2) << "."
    << setiosflags(ios::left) << setw(38) << get_dobro_naziv(i)
    << setiosflags(ios::right) << setw(8) << get_dobro_cijena(i) << " KM/sahat (cijena bez PDV)"
    << endl ;

podvuci(LINIJA_DEBELA);

rez = -1;

do {
    cout << "Odaberite zeljeno dobro (0-izlaz): " ;
    cin >> rez;

    if (rez == 0)
        return 0;

} while( (rez<1) || (rez>DOBRA_LENGTH));

return rez;
}

void get_stavke_fakture()
{
    int dobro_id, stavki_fakture;

    double kolicina;

    stavki_fakture = 0;

    // vrsti dok ne uneses max stavki fakture ili korisnik ne odabere 0 na odabiru dobra
    do {
        podvuci(LINIJA_DEBELA);
        cout << "UNOSIM STAVKU FAKTURE No: " << stavki_fakture+1 << endl;
        podvuci(LINIJA_DEBELA);

        dobro_id = meni_odaberi_dobro();

        if (dobro_id == 0) {
            // korisnik ne zeli dalji unos stavki
            // upravo ce nam kolona id dobra = 0 signalizirati da je predhodna stavka
            // posljednja unesena za fakturu
            aFaktura[stavki_fakture][FAKT_COL_ID_DOBRA] = 0;
            cout << "korisnik je prekinuo unos na stavci " << stavki_fakture+1 << endl;
            break;
        }

        cout << "Unesite kolicinu: ";
        cin >> kolicina;

        if (abs(kolicina) > 0.0) {
            // punim matricu aFaktura
            cout << "dodajem stavku" << stavki_fakture+1 << " u fakturu ." << endl;
            aFaktura[stavki_fakture][FAKT_COL_KOL] = kolicina;
            aFaktura[stavki_fakture][FAKT_COL_ID_DOBRA] = dobro_id;
            aFaktura[stavki_fakture][FAKT_COL_ID_VRIJEDNOST] = get_dobro_cijena(dobro_id) *
                kolicina;

            stavki_fakture++;
        }
    }
}

```

```

        if (stavki_fakture == FAKTURA_MAX_STAVKI) {
            cout << "max broj stavki u jednoj fakturi je " << FAKTURA_MAX_STAVKI << endl;
            break;
        }

    } while (1);

}

void show_kupac(int kupac_id)
{
    cout << "Kupac: " << get_kupac_naziv(kupac_id) << endl;
}

void show_faktura_header()
{
    cout << setiosflags(ios::right) << setw(6) << "R.br"
        << resetiosflags(ios::right) << setw(0) << ""
        << setiosflags(ios::left) << setw(40) << "        Dobro"
        << setiosflags(ios::right) << setw(10) << "Kolicina"
        << setw(10) << "Cijena"
        << setw(12) << "Vrijednost"
        << endl;
}

int show_stavka_fakture(int stavka_no)
{
    int dobro_id = (int) aFaktura[stavka_no][FAKT_COL_ID_DOBRA];

    if (dobro_id > 0) {
        cout << setiosflags(ios::right) << setw(5) << stavka_no+1
            << resetiosflags(ios::right) << setw(0) << ""
            << setiosflags(ios::left) << setw(2) << "."
            << setiosflags(ios::left) << setw(40) << get_dobro_naziv(dobro_id)
            << setiosflags(ios::right) << setw(10) << setprecision(2)
            << setiosflags(ios::fixed|ios::showpoint)
            << aFaktura[stavka_no][FAKT_COL_KOL]
            << setw(10) << get_dobro_cijena(dobro_id)
            << setw(12) << aFaktura[stavka_no][FAKT_COL_ID_VRIJEDNOST]
            << endl;

    }

    return dobro_id;
}

void show_faktura(int kupac_id)
{
    show_zaglavlje();

    show_kupac(kupac_id);
    podvuci(LINIJA_DEBELA);

    show_faktura_header();
    podvuci(LINIJA_TANKA);

    int dobro_id;

    double ukupno_bez_pdv = 0.0;

```

```

for (int i=0; i<FAKTURA_MAX_STAVKI; i++) {

    dobro_id = show_stavka_fakture(i);
    if (dobro_id == 0) break;

    ukupno_bez_pdv += aFaktura[i][FAKT_COL_ID_VRIJEDNOST];

}

podvuci(LINIJA_TANKA);

cout << setw(COL_COUNT-13) << "Ukupno BEZ PDV (KM):" << setw(12) << setprecision(2)
        << setiosflags(ios::fixed|ios::showpoint)
        << ukupno_bez_pdv << endl;

double v_pdv = ukupno_bez_pdv * PDV_STOPA / 100.00;

cout << setw(COL_COUNT-13) << "PDV (KM):" << setw(12)
        << v_pdv << endl;
cout << setw(COL_COUNT-13) << "SVEUKUPNO SA PDV (KM):" << setw(12)
        << ukupno_bez_pdv + v_pdv << endl;

podvuci(LINIJA_DEBELA);

}

// ispisi ===== ...
void podvuci(const char linija)
{
    for (int i=1; i<COL_COUNT; i++)
        cout << linija;
    cout << endl;
}

void test_sif_kupci()
{
    podvuci(LINIJA_DEBELA);
    cout << "test prikaz sifrarnika kupaca:" << endl;
    podvuci(LINIJA_TANKA);
    for(int i=1; i<=KUPCI_LENGTH; i++)
        cout << get_kupac_naziv(i) << endl;
    podvuci(LINIJA_DEBELA);
}

void test_sif_dobra()
{
    podvuci(LINIJA_DEBELA);
    cout << "test prikaz sifrarnika dobara:" << endl;
    podvuci(LINIJA_TANKA);
    for(int i=1; i<=DOBRA_LENGTH; i++)
        cout << get_dobro_naziv(i) << "\t" << get_dobro_cijena(i) << endl;
    podvuci(LINIJA_DEBELA);
}

void test_zaglavljje()
{
    show_zaglavljje();
}

void test()
{
    test_sif_dobra();
    test_sif_kupci();
    test_zaglavljje();
}

```

```

// http://publications.gbdirect.co.uk/c_book/chapter10/arguments_to_main.html
void usage(int argc, char **argv) {

    // prosetaj kroz argumente koji su proslijeđeni programu
    while (argc--) {

        char *t_arg = *argv++;

        // http://www.cplusplus.com/reference/cstring/strcmp/
        if (strcmp(t_arg, "--help") == 0) {
            cout << "UPR seminarski, hernad@bring.out.ba, ver 0.01" << endl << endl;
            cout << "Ovaj program omogućava unos pravljenje jednostavne porezne fakture" << endl;
            podvuci(LINIJA_DEBELA);
        }

        if (strcmp(t_arg, "--test") == 0) {
            test();
        }

    };

}

int main(int argc, char **argv) {

    usage(argc, argv);

    //cout << meni_odaberi_kupca();

    int kupac_id = meni_odaberi_kupca();

    get_stavke_fakture();

    // pomakni se par redova radi preglednosti
    for (int i=0; i<5; i++)
        cout << endl;

    show_faktura(kupac_id);

    return 0;
}

==== kraj ===

```