



Univerzitet u Zenici
Politehnički fakultet
odsjek:softversko inženjerstvo

**ZAVRŠNI PROJEKAT IZ PREDMETA NAPREDNE TEHNIKE
PROGRAMIRANJA, AKADEMSKA 2019./2020. GODINA
NAZIV PROJEKTA: *PRODAVNICA KOLAČA***

03.06.2020., Zenica

Studenti koji su radili na izradi projekta:

Čančar Ajna,

Okanović Amina

Objašnjenje projekta:

Projekat iz predmeta Napradne tehnike programiranja, za akademsku 2019./2020. godinu je projekat na temu: "*Prodavnica kolača*". Prije same realizacije projekta, koju ćemo naknadno objasniti u ovoj dokumentaciji, putem Google Meet platforme smo stupile u kontakt te započele rad na ovom projektu. Na samom početku smo napravile prije svega "idejni plan", a potom i sam "realizacijski plan". Dakle, prvo smo pristupile izradi idejnog plana, odnosno šta zapravo naš program treba da sačinjava i kome je on zapravo namijenjen. Program "Prodavnica kolača" je namijenjena osoblju koje zapravo radi u slastičarnama. Ideja samog projekta se bazira na olakšavanju tzv. "administrativnog dijela posla" prilikom same izrade kolača, dostupnosti kolača, narudžbi, itd. Naša ideja je zasnovana da na samom početku, korisnik ima priliku da odabere jednu od ponuđenih opcija koja je zapravo njemu u datom trenutku potrebna, odnosno da postoji start meni koji će korisniku olakšati dalji rad. Dakle, korisnik bi u samom start meniju imao više izbora, poput unosa kolača, gdje bi korisnik imao mogućnost da pored "standardnih" kolača u sam program unese i neki novi kolač iz svoje ponude, zatim da korisnik ima mogućnost pregleda dostupnih kolača, pretrage kolača prema njihovom ID-u, gdje bi korisnik dobio sve potrebne informacije vezano za odabrani kolač, zatim pretraga prema nazivu kolača, ukoliko korisnik ne zna ID kolača, pretraga prema nazivu bi mu omogućila ispis svih potrebnih informacija vezano za odabrani kolač, zatim ispis narudžbe kolača, gdje bi korisnik imao mogućnost da zabilježi pristiglu narudžbu, odnosno da unese sve potrebne informacije koje su neophodne za tu narudžbu, te da ta narudžba bude spremljena kao .txt file, zatim korisnik bi imao mogućnost da promijeni i datum proizvodnje kolača, prilikom izrade novih kolača čiji se datum proizvodnje i datum isteka mijenja, kao i mogućnost promjene stanja samog kolača, koja se odnosi da li je traženi kolač trenutno dostupan u toj slastičarnici ili pak nije. Na samom kraju, korisnik ima mogućnost odabira opcije kraj, koja će mu ispisati odgovarajuću poruku i omogućiti izlazak iz programa. Dakle, sam program, odnosno projekat "Prodavnica kolača" je koncipiran da prvenstveno služi osoblju, kako bi im olakšao navedeni dio posla. Nakon izrade kako smo nazvale idejnog plana, napravile smo i realizacijski plan, gdje smo podijelile naše zadatke

vezano za izradu programa, kao i pisanja same dokumentacije, nakon čega smo pristupile izradi istih. U nastavku dokumentacije vezano za projekat “Prodavnica kolača” će biti komentarisani kod, odnosno specifični dijelovi koda.

Pojašnjenje koda:

U samom uvodnom dijelu pozvali smo sve biblioteke koje su nam potrebne da bi program radio.

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>
#include <sstream>
```

Na samom početku imamo deklarisanu enumeraciju:

```
enum Stanje{NEMA=1, IMA};
```

Nakon toga imamo dvije strukture, i to struct Datum i struct Kolaci, te smo definisali varijable unutar tih struktura.

```
struct Datum{
    int dan_proizvodnje, mjesec_proizvodnje, godina_proizvodnje;
    int dan_isteka, mjesec_isteka, godina_isteka;
};

struct Kolaci{
    string id;
    string ime;
    double cijena;
    Datum datum;
    Stanje stanje;
```

Nakon toga definisali smo funkciju unosKolaca bez povratnog tipa. Funkcija void unosKolaca () kao što i sam naziv govori vrši unos novog kolača u datoteku s tim da se ne može unijeti kolač s istim ID-om koji već postoji u datoteci i za to nam služi for (;;) petlja koja je objašnjena u nastavku:

```
void unosKolaca(){
```

```

int br;
for(;;){ // Beskonačna for petlja
    string provjera;
    br=0;
    cout << "ID: ";
    cin >> this->id; // Unos id u objekat od strukture
    ifstream datotekakolaca("kolaci.txt"); //Otvaranje datoteke
    while(!datotekakolaca.eof()){ // Pretrazivanje datoteke i pretraga da
        li postoji isti ID
            datotekakolaca >> provjera;
            if(provjera == id){
                br++; // Ako postoji isti ID brojac se uvecava
            }
        }
    if(br!=0){ // Ukoliko je brojač različit od nule, ispisuje se poruka da
        taj ID već postoji i traži se ponovni unos
            cout << "Kolac sa tim ID-om vec postoji!\n";
            cout << "Molimo unesite drugi id!\n";
        }else if(br==0){ // Ukoliko je brojač jedan nuli
            break; // Izlazi se iz for petlje
        }
    }

    ofstream datoteka("kolaci.txt", ios::app); // Unos u datoteku
    if(datoteka.is_open()){
        datoteka << "ID: " << id << endl;
        cin.ignore();
        cout << "Naziv: ";
        char *kolac = new char[50]; //Dinamičko alociranje memorije
        cin.getline(kolac, 50);
        ime = kolac;
        datoteka << "Naziv: " << ime << endl;
        cout << "Datum proizvodnje: ";
        cin >> this->datum.dan_proizvodnje >>
        this->datum.mjesec_proizvodnje >> this->datum.godina_proizvodnje;
    }
}

```

```

        datoteka << "Proizvedeno: " << setfill('0') << setw(2) <<
this->datum.dan_proizvodnje << "." << setfill('0') << setw(2) <<
this->datum.mjesec_proizvodnje << "." << this->datum.godina_proizvodnje << "."
<< endl;

        cout << "Datum isteka: ";
        cin >> this->datum.dan_isteka >> this->datum.mjesec_isteka >>
this->datum.godina_isteka;
        datoteka << "Istice: " << setfill('0') << setw(2) <<
this->datum.dan_isteka << "." << setfill('0') << setw(2) <<
this->datum.mjesec_isteka << "." << this->datum.godina_isteka << "." << endl;
        /* setfill('0') << setw(2) omogućava unos nule ispred jednocifrenog broja,
odnosno da ukoliko se unese datum npr. 5 5 2020, u datoteku se upisuje 05.05.2020.

        cin >> this->cijena;
        datoteka << "Cijena: " << setprecision(2) << fixed << this->cijena <<
endl;
        // setprecision(2)<< fixed se odnosi na to da je cijena zaokružena na dvije
decimale

```

Dalje, u programu imamo enumeraciju, koja predstavlja da li ima određenih kolača na stanju. Ukoliko unesemo vrijednost 1 kolača nema na stanju, ukoliko je ta vrijednost 2 kolača ima na stanju. Dakle, uz pomoć *casting* operatora se pretvara u enumeraciju i unosi u samu datoteku.

```

int x;

cout << "Da li kolaca ima trenutno na stanju?\n";
cout << "\t1. Nema na stanju\n";
cout << "\t2. Ima na stanju\n";
do{
    cout << "Odabir: ";
    cin >> x;

}while(x<1 || x>2);
this->stanje = Stanje(x);
datoteka << "Stanje: ";

```

```

    if(x==1){
        datoteka << "Nema" << endl;
    }else if(x==2){
        datoteka << "Ima" << endl;
    }
}
}else{
    cout << "[GRESKA] Datoteka se ne moze otvoriti!\n";
}

    datoteka.close();
}

```

Sve navedene funkcije se nalaze, odnosno smještene su u strukturi.

Nakon toga imamo funkciju **void** pregledKolaca, gdje se iz datoteke “kolaci.txt” čita sadržaj, te se ispisuje na ekran. Također, unutar same funkcije stavile smo i `left << setw (br)` radi ljepšeg ispisa. Dakle, ukoliko je datoteka otvorena, učitat će se podaci i ispisati.

Sljedeća funkcija jeste **void** pretragapremaID. Dakle u ovoj funkciji, prvo unosimo ID, zatim ga unosimo u char, a onda taj char pretvaramo u string, te otvaramo samu datoteku. Nakon što otvorimo datoteku vršimo provjeru jednog po jednog ID-a. Kada program nađe traženi ID, postavljamo da je `a=true`. Ukoliko je `a=true`, program će ispisati podatke vezano za traženi kolač (naziv, stanje, datum proizvodnje, datum isteka,..). Ukoliko je `a=false`, program će ispisati odgovarajuću poruku, odnosno ispisat će da ne postoji kolač pod traženim ID unosom.

Nakon toga slijedi funkcija **void** Promijeni, koja se odnosi na promjenu datuma proizvodnje i datuma isteka. Dakle, funkcija promijeni mijenja datum određenog kolača na osnovu unesenog ID-a. Prvo se provjerava da li uneseni ID postoji u datoteci, te ukoliko ne postoji ispisuje se odgovarajuća poruka, nakon koje se zahtjeva ponovni unos. S druge strane, ukoliko uneseni ID postoji u datoteci, vršit će se promjena datuma proizvodnje i isteka. *Zamjena* je opisana u nastavku:

```

    cout << "Unesite novi datum proizvodnje: ";

```

```
cin >> datum.dan_proizvodnje >> datum.mjesec_proizvodnje >> datum.godina_proizvodnje;
```

```
ostreamstream s;
```

/* ostreamstream s; je deklaracija string s u koji unosimo datum ali na taj način da ga unesemo tako što automatski namjestimo izgled za datoteku, odnosno omogućava dodavanje nula tokom unosa jednocifrenih brojeva, pr. 5 5 2020, bit će upisano 05.05.2020. */

```
s << setw(2) << setfill('0') << datum.dan_proizvodnje << "." << setw(2) << setfill('0') << datum.mjesec_proizvodnje << "." << datum.godina_proizvodnje << ".";
```

```
string string_niz[1000], provjera; // smješta se linija po linija iz datoteke
```

int brojac=0; // Deklarisan brojač koji će brojati koliko ima linija do datuma koji želimo promijeniti, odnosno broji linije do kolača s traženim ID-om čiji datum proizvodnje želimo promijeniti

```
bool nadjen = false;
```

```
fstream file;
```

```
file.open("kolaci.txt", ios::in);
```

/* U narednoj while petlji se traži ID unesenog kolača u datoteci i broji se koliko ima linija do tog ID-a i do tog datuma proizvodnje, jer se datum proizvodnje unosi prvo*/

```
while(!file.eof()){
```

```
file >> provjera;
```

```
if(provjera == "ID:" && nadjen==false){
```

```
file >> provjera;
```

```
if(provjera == id){
```

```
brojac+=2; // Kada se nađe ID dodaje se +2 jer u datoteci
```

imamo još dvije linije od datuma proizvodnje

```
nadjen = true;
```

```
}else{
```

```
while(getline(file,provjera)){
```

```
if(provjera == "Stanje: Ima" || provjera == "Stanje:
```

```
Nema"){
```

```

        break;
    }else{
        brojac++;
    }
}
}
}
    brojac++;
}
if(nadjen == true){
    // Nakon što je ID pronađen izlazi se iz petlje
    break;
}
}
file.close();

```

```

int brojac_linija=0; //Deklarisan brojač koji će brojati ukupan broj linija
fstream file2;
file2.open("kolaci.txt", ios::in);

```

/* U sljedećoj while petlji broji se ukupan broj linija u datoteci nakon čega se sadržaj briše iz datoteke*/

```

while(getline(file2, string_niz[brojac_linija])){
    brojac_linija++; // Kroz brojanje linija koda, linije se spremaju u
    prethodno deklarisan string niz
}

```

```

file2.clear(); //Brisanje sadržaja datoteke
file2.close();
fstream file3;
file3.open("kolaci.txt", ios::out);

```

// U sljedećoj for petlji se ponovo unosi sadržaj u datoteku ali na specifičan način

```

for(int j=0; j<brojac_linija; j++){
    if(j != brojac-1){ // Sve dok je j različito od pozicije na kojoj se

```

nalazi datum proizvodnje kolača


```

        file3 << string_niz[j] << endl; // Linije datoteke se pišu u nju iz
string niza
    }else{ // Međutim, kada program dođe do linije u kojoj se nalazi datum
    proizvodnje kolača, tada se na mjesto unosi novi datum

        file3 << "Proizvedeno: " << s.str() << endl; // Mjesto je određeno
    prvom while petljom, s.str se odnosi na ispis datuma jer samo s bi ispisalo adresu
    }
    }
    file3.close();

```

Što se tiče promjene **datuma isteka kolača**, on se vrši na identičan način kao i promjena datuma proizvodnje kolača koja je već opisana.

Nakon toga, imamo funkciju **void** PromjenaStanja, koja je identična funkciji **void** Promijeni. U funkciji Promijeni imali smo promjenu datuma proizvodnje i datuma isteka kolača, dok u funkciji *PromjenaStanja* mijenjamo stanje kolača, odnosno da li je neki kolač trenutno dostupan ili ne.

Sljedeća funkcija je **void** pretragapremaNazivu koja omogućava pretragu nekog kolača prema njegovom imenu. Dakle, ovo je funkcija u kojoj radimo s karakterima. U samoj funkciji imamo deklarisan char trazen[50] koji omogućava rad s karakterima. Dakle, pomoću ove funkcije korisnik ima mogućnost pretražiti kolač prema nazivu, nakon čega će se otvoriti datoteka i vršiti provjera. Ukoliko se željeni kolač nalazi u datoteci, korisniku će se ispisati podaci o tom kolaču, ukoliko kolač ne postoji u datoteci (e= =0), program će ispisati odgovarajuću poruku, odnosno da ne postoji kolač pod tim nazivom.

Nakon toga, imamo funkciju povratnog tipa **double** CijenaKolaca. Radi se o rekurzivnoj funkciji za izračunavanje cijene kolača, gdje će svaki treći kupljeni kolač imati popust od 5%.

Sljedeća funkcija je **void** narudzbaKolaca. U ovoj funkciji se od korisnika traži da unese ID kolača za narudžbu, nakon provjere ukoliko kolač pod tim ID-om ne postoji ispisuje se odgovarajuća poruka. S druge strane ukoliko je

željeni kolač pronađen, od korisnika se traži da unese ime, prezime i količinu. Nakon toga ova funkcija poziva rekurziju koja je prethodno definisana. Dakle, ova funkcija vrši upis narudžbe u sistem, ali s tim da se narudžba sprema u posebnu datoteku u obliku računa ili priznanice. Kako bi se osigurala uspješnost, nakon što korisnik uspješno naruči program će mu ispisati odgovarajuću poruku, odnosno da je njegova narudžba uspješno snimljena.

Zatim, slijedi funkcija **void** Meni, koja služi za ispis te korisniku daje jasnu sliku da izabere onu opciju koja mu najviše odgovara.

Na kraju, u glavnoj funkciji (**int main**) je prije svega svega omogućen izbor željene opcije iz menija, te se na osnovu odabira pozivaju se i odgovarajuće funkcije.