

Projektni zadaci

Istorija revizija

<i>Verzija</i>	<i>Autori</i>	<i>Datum</i>	<i>Status</i>	<i>Komentar</i>
1.00w	Vesna Stanojević	8-Maj-2019	Inicijalna verzija	Početna verzija dokumenta

Reference

Sadržaj

Opšta uputstva	5
1 AvioManager (Evidencija letova).....	6
2 HotelBooking (Hotelske rezervacije)	8
3 MovieBase (Videoteka)	10
4 CourseDiary (Evidencija akademskih kurseva)	12
5 Battleship (Potapanje brodova)	14
6 Mastermind (Skočko).....	16

Opšta uputstva

Da bi projekat bio ocenjen najvišom ocenom, program treba da bude napisan u skladu sa kriterijumom za bodovanje koji se nalazi u daljem tekstu. Rad programa treba u potpunosti da odgovara tekstu zadatka. Takođe, program treba da sadrži što je moguće viši nivo zaštite od nepravilnog korišćenja (npr. ukoliko se učitava nedozvoljen ulazni podatak program bi trebalo da traži ponovni unos).

Kriterijum bodovanja:

1. Ispoštovan standard kodiranja (promenljive, funkcije, moduli su nazvani prema pravilima navedenim u prezentaciji StandardKodiranja)
2. Odgovarajuća interakcija sa korisnikom programa (pre svakog unosa nekog podatka mora da postoji poruka korisniku šta treba da unese)
3. Zaštita od nepravilnog unosa (npr. ako je u zadatku navedeno da korisnik treba da unese dužinu niza ≤ 20 , korisnik može da unese samo vrednosti između 0 i 20)
4. Zadatak je pravilno modularizovan (postoje odgovarajući moduli i celine koda su izmeštene u funkcije)
5. Upotreba odgovarajućih elemenata programskog jezika C (upotreba brojačkih kada je poznat broj iteracija, upotreba ciklusa sa izlaskom na vrhu kada se je potrebno da se naredba izvrši najmanje nijednom, upotreba ciklusa sa izlaskom na dnu kada naredba treba da se izvrši najmanje jednom)
6. Sktrukture i tipovi podataka u rešenju zadatka su u skladu sa onim što se traži u zadatku (npr. ako se u zadatku traže vremena trkača, onda treba da se koristi float ili double tip, ili ako se traži broj elemenata niza, onda se koristi unsigned, itd)
7. Program za dati ulaz daje odgovarajući izlaz
8. Algoritmi u rešenju zadatka su u skladu sa onim što se traži u zadatku

U velikom broju zadataka program treba da koristi unapred pripremljene datoteke. Jedna od obaveza učenika je da sami kreiraju ove datoteke i da ih popune adekvatnim podacima.

Uz svaki zadatak treba priložiti prateću dokumentaciju, detaljno uputstvo za korišćenje programa kao i detaljan opis strukture i tipova podataka.

Treba da formirate svoj tim. Tim se sastoji od četiri učenika. Vaš zadatak je da nam dostavite mejlom članove timova (ili sutra na testu) , a nakon toga ćemo vam dodeliti projekte .

Rok za izradu projekta je dve nedelje, odnosno 23.05. je odbrana projekta (svi zajedno).

1 AvioManager (Evidencija letova)

Implementirati program za vođenje evidencije o letovima jedne aviokompanije.

(Podatke čuvati u vidu jednostruko spregnute liste.)

Podaci o jednom letu obuhvataju:

- Broj leta
- Oznaku aviona
- Ukupan broj mesta
- Datum i vreme poletanja
- Mesto poletanja
- Datum i vreme sletanja
- Mesto sletanja
- Da li je let otkazan
- Cenu leta
- Spisak prijavljenih putnika na letu

(Može da bude struktura.)

Podaci o jednom putniku obuhvataju:

- JMBG
- Ime
- Prezime
- Datum rođenja
- Pol
- Broj pasoša

(Može da bude struktura.)

Program treba da omogućiti:

- Kreiranje novog leta i njegovo dodavanje u kolekciju letova
- Registraciju novog putnika i njegovo dodavanje na let
- Listanje:
 - Lista svih letova
 - Lista svih letova nekog putnika na osnovu njegovog JMBG-a

- Prikaz svih letova do nekog odredišta
- Prikaz svih letova iz nekog mesta
- Listanje svih letova do određene cene
- Prikaz informacija o nekom letu
- Prikaz liste putnika na nekom letu
- Izmenu podataka o letu
- Izmenu podataka o putniku na određenom letu
- Brisanje putnika sa nekog leta
- Otkazivanje leta
- Snimanje podataka u binarnu datoteku
- Učitavanje podataka iz binarne datoteke

2 HotelBooking (Hotelske rezervacije)

Implementirati program za vođenje evidencije o rezervacijama soba jednog hotela.

(Podatke čuvati u vidu jednostruko spregnute liste.)

Podaci o jednoj rezervaciji obuhvataju:

- Broj sobe (ukupan broj soba u hotelu je 25)
- Sprat
- Broj ležaja u sobi
- Datum i vreme dolaska
- Datum i vreme odlaska
- Da li je rezervacija otkazana
- Cenu smeštaja po ležaju
- Osobu na koju se rezervacija vodi
- Spisak gostiju u sobi

(Može da bude struktura.)

Podaci o jednom gostu obuhvataju:

- JMBG
- Ime
- Prezime
- Datum rođenja
- Pol
- Broj lične karte

(Može da bude struktura.)

Program treba da omogućiti:

- Kreiranje nove rezervacije sobe i njeno dodavanje u kolekciju svih rezervacija
- Registraciju novog gosta i njegovo dodavanje u određenu rezervaciju sobe
- Listanje:
 - Lista svih rezervacija (broj sobe, datum i vreme dolaska/odlaska, da li je rezervacija otkazana, osoba na koju se rezervacija vodi)
 - Prikaz gostovanja jednog gosta na osnovu njegovog JMBG-a
 - Lista gostiju u nekoj sobi

- Lista svih gostiju u hotelu
- Lista svih soba koje nisu zauzete na određenom spratu
- Lista svih soba sa određenim brojem ležaja koje su slobodne
- Istorija jedne sobe (informacije o svim rezervacijama koje su se odnosile na ovu sobu – npr. datum i vreme dolaska/odlaska, da li je rezervacija otkazana, osoba na koju se rezervacija vodi)
- Prikaz svih informacija o nekoj rezervaciji
- Izmenu podataka o rezervaciji
- Izmenu podataka o određenom gostu
- Otkazivanje rezervacije
- Snimanje podataka u binarnu datoteku
- Učitavanje podataka iz binarne datoteke

3 MovieBase (Videoteka)

Implementirati program za vođenje evidencije o iznajmljivanjima filmova u jednom video klubu.

(Podatke čuvati u vidu jednostruko spregnute liste.)

Svako iznajmljivanje treba da obuhvati sledeće podatke:

- Broj članske kartice osobe koja iznajmljuje filmove
- Datum i vreme iznajmljivanja
- Datum i vreme vraćanja filmova
- Da li su filmovi vraćeni
- Spisak filmova koje iznajmljivanje obuhvata

(Može da bude struktura.)

Podaci o filmu obuhvataju:

- Identifikator filma
- Naziv
- Žanr
- Jezik
- Godinu izdavanja
- Cenu iznajmljivanja
- Da li je film iznajmljen

(Može da bude struktura.)

Podaci o jednoj osobi obuhvataju:

- JMBG
- Ime
- Prezime
- Broj članske kartice

(Može da bude struktura.)

Program treba da omogući:

- Evidentiranje novog iznajmljivanja i njegovo dodavanje u listu svih iznajmljivanja
- Registraciju novog filma i njegovo dodavanje u listu iznajmljivanja

- Vraćanje filmova u video klub (svi filmovi se vraćaju istovremeno)
- Listanje:
 - Svih filmova u videoklubu
 - Svih iznajmljenih filmova
 - Svih neiznajmljenih filmova
 - Prikaz svih filmova nekog žanra/jezika/godine/imena
 - Svih iznajmljivanja jedne osobe
 - Svih iznajmljivanja jednog filma
 - Svih iznajmljivanja koja su u kašnjenju
- Izmene podataka o nekom iznajmljivanju (uklanjanje nekog filma sa liste iznajmljivanja, trajanje iznajmljivanja, osoba koja iznajmljuje)
- Izmenu podataka o nekom filmu
- Snimanje podataka u binarnu datoteku
- Učitavanje podataka iz binarne datoteke

4 CourseDiary (Evidencija akademskih kurseva)

Implementirati program za vođenje evidencije o polaznicima kurseva u jednoj školi.

(Podatke čuvati u vidu jednostruko spregnute liste.)

Podaci o jednom kursu obuhvataju:

- Šifru kursa
- Naziv
- Oblast
- Cenu kursa
- Predavača (Ime i prezime)

(Može da bude struktura.)

Podaci o polazniku obuhvataju:

- JMBG
- Ime
- Prezime
- Datum rođenja
- Zanimanje
- Pol

(Može da bude struktura.)

Svaki termin održavanja kursa treba da obuhvata podatke:

- Kurs koji se u tom terminu održava
- Spisak polaznika koji su prijavljeni u tom terminu
- Mesto održavanja
- Datum i vreme početka održavanja
- Datum i vreme završetka održavanja

(Može da bude struktura.)

Program treba da omogućiti:

- Evidentiranje novog kursa i njegovo dodavanje u evidenciju svih kurseva
- Evidentiranje novog termina za kurs i njegovo dodavanje u evidenciju svih termina

- Evidentiranje podataka o polazniku kursa i njegova prijava za kurs u nekom terminu
- Odustajanje polaznika od kursa u nekom terminu
- Listanje:
 - Svih kurseva
 - Svih kurseva nekog predavača
 - Svih kurseva iz željene oblasti
 - Svih termina
 - Svih termina za koje je određeni kandidat prijavljen
 - Svih kandidata koji su prijavljeni na kurs u određenom terminu
 - Svih termina koji se održavaju u nekom mestu
 - Svih termina čije održavanje počinje željenog datuma
- Izmenu podataka o terminu održavanja nekog kursa
- Izmenu podataka o nekom polazniku
- Snimanje podataka u binarnu datoteku
- Učitavanje podataka iz binarne datoteke

5 Battleship (Potapanje brodova)

Implementirati društvenu igru *Battleship* (*Potapanje brodova*).

Pravila igre Battleship su sledeća:

- Igru igraju 2 igrača.
- Svaki igrač ima svoje bojno polje (*bojište*), tajnu mapu (*mapa*) i brodove.

➤ Bojno polje i tajna mapa su predstavljeni matricama 10x10, pri čemu su vrste označene slovima engleske abecede A-J, a kolone brojevima 1-10, čime je svako polje matrice jednoznačno određeno uređenim parom, npr. A8.

➤ Brodovi (10 komada) su predstavljeni poliomina (pločicama) datim u sledećoj tabeli:

<i>Količina</i>	<i>Naziv</i>	<i>Dimenzije</i>
1	Nosač aviona	1x4
2	Krstarica	1x3
3	Razarač	1x2
4	Podmornica	1x1

- Pre početka igre igrači postavljaju brodove na mapu prema sledećim pravilima:

- brodovi se postavljaju u horizontalni ili vertikalni položaj tako da pokrivaju ceo broj polja matrice;
- polja koja zauzimaju dva različita broda ne smeju imati zajedničkih tačaka;
- svi brodovi se moraju postaviti.

- Igra se igra u naizmeničnim potezima.

➤ U svakom potezu igrač bira (*napada*) jedno protivničko polje. Protivnik je dužan da ga obavesti o tome da li na izabranom polju njegove mape postoji brod. U slučaju da postoji, igrač ima pravo da nastavi sa napadom, sve dok ne *promaši* (izabere polje protivničke mape na kojem ne postoji brod), čime se završava potez.

➤ Nakon svakog napada protivnik je dužan da, ukoliko su sva polja koja zauzima neki brod pogodoena, ukloni taj brod sa mape i to prijavi igraču.

➤ Prilikom svakog poteza, igraču je na raspolaganju bojno polje na kojem su označena polja koja je do tog trenutka napao. Postoje dve vrste oznaka, jedna za pogodak i jedna za promašaj.

- Igra se završava kada jedan od igrača (*pobednik*) potopi sve protivničke brodove (jasno je da uvek postoji tačno jedan pobednik).

Omogućiti korisniku igranje Battleship igre *protiv računara* primenom sledećih osnovnih funkcionalnosti:

- Postavljanje brodova na mapu:
 - stalan uvid u trenutno stanje mape i brodove koji nisu postavljeni;
 - postavljanje i skidanje brodova sa mape;
 - završetak postavke brodova na mapu;
 - nasumično popunjavanje protivničke mape.

- Početak igre (nasumično se bira igrač koji počinje).
- Povlačenje poteza, pri čemu protivnik (računar) povlači poteze nasumično.
- Završetak igre.

6 Mastermind (Skočko)

Implementirati igru *Mastermind* (skočko).

Pravila igre Mastermind su sledeća:

- Na raspolaganju je 6 različitih znakova: tref, karo, herc, pik, zvezda, skočko.
- Zadaje se nepoznata varijacija od 4 znaka, pri čemu može doći do ponavljanja.
- Cilj igre je pronaći nepoznatu varijaciju.
- Korisnik ima maksimalno 6 pokušaja da pronađe pravu varijaciju, pri čemu, nakon svakog pokušaja dobija informacije o broju znakova:
 - koji se nalaze na dobrom mestu;
 - koji se nalaze u varijaciji, ali nisu na dobrom mestu.

Omogućiti korisniku igranje Mastermind igre primenom sledećih osnovnih funkcionalnosti:

- Nasumično zadavanje nepoznate varijacije.
- Realizacija pokušaja sa prikazom prethodnih pokušaja i njihovih ishoda.
- Završetak igre.