

# Programski jezici

## -Laboratorijska vežba 4-

### -C#-

#### **NAPOMENA:**

***Ovom laboratorijskom vežbom je obuhvaćen materijal koji je obrađen u prva dva termina računskih vežbi. Svaki od zadataka treba rešiti korišćenjem ili interfejsa ili apstraktne klase, u zavisnosti od sadržine zadatka. Svaka od klasa koja se kreira treba da sadrži konstruktor u kome će se postavljati vrednosti svih atributa. Sve zadatke implementirati na programskom jeziku C#.***

1. Projektovati sistem za potrebe jedne fabrike. Kreirati klasu Fabrika koja se sastoji od većeg broja proizvodnih traka, gde se za svaku od traka vodi evidencija o tome koja je njena šifra, na kom spratu se nalazi, da li je aktivna, koji je maksimalni kapacitet u jednom proizvodnom ciklusu kao i ime njenog serviser. Svaka od traka ima isti način rada, prilikom startovanja treba da objavi da je krenula sa radom. Postoji traka za pravljenje lizalica, čokoladica i smokija. Za svaku od traka je zajedničko da može da krene sa proizvodnjom sa određenom količinom materijala i da može da stane sa radom. Traka koja pravi lizalice treba da prihvati koliko god šećera radnik želi da joj postavi i da na osnovu toga vrati koju količinu lizalica je napravila, ali ako broj lizalica premaši maksimalni kapacitet, traka treba da prikaže na svom ekranu „Molim Vas zovite (ime serviser)“. Traka koja pravi čokoladice, treba da objavi koliko joj vremena treba da preradi kakao u zavisnosti od unete količine. Traka koja pravi smoki treba da proveri da li je prosleđena količina kikirikija veća od maksimalnog kapaciteta, ako jeste da vrati maksimalnu količinu koje može da napravi, ako ne da vrati tu količinu kikirikija.

Definisati nad klasom Fabrika indeksers koji će da vraća traku sa zadatim indeksom iz liste traka. U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Fabirka i dodati joj nekoliko traka, pozvati metode da se traka pokrene i da se kreiraju proizvodi i prikazati koju količinu će traka da proizvede.

2. Projektovati sistem za potrebe HR odseka jedne kompanije. U klasi Kompanija treba da se čuva lista zaposlenih. Svakog zaposlenog karakteriše ime, prezime i plata. Na osnovu pozicije na kojoj rade zaposleni razlikuju se WebDeveloperi, QA inženjeri i DevOps inženjeri. Pored svih informacija koje ima Zaposleni, WebDeveloper dodatno treba da čuva listu tehnologija (tekstualni podaci) koje poznaje, QA inženjer treba da ima postavljen broj QA testova koje može da uradi na dnevnom nivou, a DevOps inženjer treba da pamti datum kada je poslednji put pregledao računare. Za svaku od ovih pozicija treba omogućiti da se prikazuju informacije ove pozicije, odnosno da se štampa ona informacija koja je njemu svojstvena. Svakom od zaposlenih se u trenutku kada je to potrebno obračunava bonus i to za WebDevelopere 20% visine plate, za QA 10%, a za DevOps 30%.

Dodatno treba preklopiti operator += nad listom radnika koji će da doda novog radnika u listu. U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Kompanija sa određenim brojem radnika, prikazati za svakog opšte informacije i odrediti koliko je ukupno novca potrebno da firma izdvoji za njihove bonuseve.

3. Projektovati sistem za potrebe sajma automobila. Klasa Sajam vodi računa o prostoru u kom su automobili raspoređeni, a radi se o matrici dimenzija  $n \times n$ . Svaki od automobila ima stepen prioriteta po kom se raspoređuje (1 najveći). Audi je prioriteta 1, BMW prioriteta 2, a Mercedes prioriteta 3. Kada se kreira novi automobil treba da se prosledi organizatoru koji ima informacije o stanju u sali, organizator tada treba da smesti auto u zavisnosti od prioriteta. Auto prvog prioriteta će ići u prvo slobodno mesto počevši od početka sale. Auto drugog prioriteta će biti u redovima koji su deljivi sa 3, ako ne postoji slobodnog mesta u ovim redovima treba da proba da smesti auto od poslednjeg mesta u sali pa naviše. Auto trećeg prioriteta treba da bude smešten po glavnoj dijagonali sale,

ukoliko su mesta duž cele dijagonale zauzeta, smestiti auto na prvo slobodno mesto počevši od početka sale.

Definisati nad klasom Sajam indeksir koji će da vraća automobil sa zadatim indeksima iz matrice automobila (sajam[i,j]). U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Sajam automobila, smestiti u okviru sajma veći broj automobila različitih prioriteta i prikazati indekse mesta na kom su smešteni. Izazvati situaciju gde će prva očekivana pozicija biti zauzeta da dokažete da adekvatno postavlja automobile.

4. Projektovati sistem za vođenje evidencije o prvenstvu u atletici. Kreirati klasu Prvenstvo koja će beležiti sve takmičare koji učestvuju. Za svakog od takmičara se pamti ime, prezime, država koju predstavlja, lista kategorija u kojima se takmiči zajedno sa najboljim rezultatom do sada u toj kategoriji, kao i faktor spremnosti. Pored dodavanja takmičara na prvenstvo, moguće je i diskvalifikovati takmičara s prvenstva izbacivanjem iz liste svih takmičara. Za svakog takmičara treba omogućiti izračunavanje finalnog broja. Ukoliko se radi o trkaču, za njega će se dodatno čuvati i procentualan broj sati provedenih u pripremi za ovo takmičenje dnevno (0h – 0, 24h - 1), a finalni broj je jednak broju sati pomnoženim faktorom spremnosti. Ukoliko se radi o bacaču koplja, za njega se pamti i broj pokušaja u kojima je premašio ciljanu udaljenost. Za njega se finalni broj računa tako što se broj pokušaja u kojima je premašio udaljenost pomnoži faktorom spremnosti. Moguće je i diskvalifikovati takmičara izbacivanjem iz liste svih takmičara.

Dodatno treba preklopiti operator -= nad listom takmičara koji će da diskvalifikuje takmičara brisanjem iz liste. U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main klasi kreirati instancu klase Prvenstvo sa većim brojem sportista, izračunati za svakog finalni broj i prikazati tri takmičara sa najvećim finalnim brojem, od prvog do trećeg. Nakon toga diskvalifikovati takmičara koji je osvojio jedno od prva tri mesta po finalnom broju i izračunati finalne brojeve ponovo i pokazati da će neki novi takmičar da zauzme jedno od vodećih mesta.

5. Projektovati sistem za potrebe jedne vremenske stanice. Vremenska stanica u različitim trenucima meri temperaturu i vlažnost vazduha. Klasa Vremenska stanica u svakom trenutku vodi evidenciju o terminalima koji su registrovani za primanje informacija o temperaturi. Informacije o temperaturi i vlažnosti vazduha ne moraju da stignu u isto vreme. Terminal mora da ima mehanizam da u trenutku kad vremenska stanica prođe kroz listu terminala i svima pošalje nove informacije ili o vremenu ili o vlažnosti vazduha to prikaže. Terminali na različit način prikazuju temperature pre svega zato što temperature stižu u stepenima Celzijusa, a pored Celzijusovih terminala, postoje i terminali koji temperaturu prikazuju u Farenhajtima i Kelvinima. Sto se tiče vlažnosti vazduha, Celzijusov terminal prikazuje realnu vlažnost vazduha, dok je Farenhajtov poveća za 1%, a Kelvinov smanji za jedan. Takođe, vremenska stanica ima mogućnost da dodaje i briše terminale iz svoje evidencije. Onog trenutka kada stigne informacija o temperaturi ili vlažnosti treba da obavesti svaki terminal koji prati ove informacije.

Dodatno treba preklopiti operator += nad listom terminala koji će da doda novi terminal u listu. U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Vremenska stanica sa većim brojem terminala, poslati informacije o temperaturi i vlažnosti, nakon toga obrisati neki terminal i poslati ponovo, da se vidi koji terminali će sada reagovati.

6. Projektovati sistem za potrebe turističke agencije. U klasi Agencija treba da se čuva naziv agencije i lista putovanja u ponudi agencije. Svako putovanje karakteriše naziv destinacije, datum početka putovanja, trajanje u danima, cena smeštaja po danu i cena puta. Agencija organizuje sledeće vrste putovanja: Letovanje, Zimovanje i Seminar. Pored svih informacija koje ima Putovanje, Letovanje dodatno treba da čuva cenu doplate za trajekt (ako se trajekt koristi za odlazak do destinacije), Zimovanje dodatno treba da čuva cenu ski karte po danu, a Seminar treba da pamti niz izleta (izlet predstaviti samo tekstualnim nazivom) koji će bi ponuđeni učesnicima seminara. Za svako od putovanja treba omogućiti da se računa ukupna cena putovanja i da se prikazuju informacije za konkretnu vrstu putovanja, odnosno da se štampa ona informacija koja je njemu svojstvena. Klasa Agencija treba da sortira listu putovanja po ceni.

Dodatno treba preklopiti operator += nad listom putovanja koji će da doda novo putovanje u listu. U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Agencija sa određenim brojem putovanja, sortirati putovanja po ceni i prikazati informacije o Agenciji i svim putovanjima iz njene ponude.

7. Projektovati sistem za potrebe evidencije radnika neke firme. U klasi Firma treba da se čuva naziv firme i lista radnika te firme. Svakog radnika karakteriše ime, prezime, JMBG. Sistem treba da razlikuje stalno zaposlenog radnika, honorarnog radnika i šefa. Pored svih informacija o radniku za stalno zaposlenog radnika se čuva naziv radnog mesta i iznos mesečne zarade. Za honorarnog radnika se dodatno čuva broj mesečnih radnih sati i iznos zarade po radnom satu. Šef može da bude samo neko od stalno zaposlenih radnika i zbog pozicije šefa dobija zaradu uvećanu za 20% u odnosu na osnovnu. Za svakog od radnika treba omogućiti da se računa mesečna zarada i da se prikazuju informacije za konkretni tip radnika, odnosno da se štampa ona informacija koja je njemu svojstvena. Klasa Firma treba da izračuna prosečnu zaradu radnika u firmi i da otpusti radnike čija zarada je 3 ili više puta veća od prosečne.

Definisati nad klasom Firma indeksir koji će da vraća radnika sa zadatim indeksom iz liste radnika. U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Firma sa određenim brojem radnika, otpustiti radnike sa previsokom zaradom i prikazati informacije o Firmi i svim njenim radnicima.

8. Projektovati sistem za pomoć pri parkiranju vozila na javnom parkingu. Svako vozilo karakteriše registarska oznaka, proizvođač i naziv modela. Vozilo samo određuje koliko standardnih parking mesta mu je potrebno. Automobil zauzima uvek jedno parking mesto, autobus zauzima uvek 5 parking mesta, a kamion može da zauzme različit broj parking mesta (čuva taj broj kao atribut klase). Dodatni atribut autobusa je broj putnika, a dodatni atribut kamiona nosivost u tonama. Parking čuva parking mesta kao matricu referenci na vozila koja ima M vrsta i N kolona. Za vozila koja zauzimaju više od jednog parking mesta ta mesta

treba da se nalaze u istoj vrsti i jedno pored drugog. Vozilo ulazi na parking tako što poziva odgovarajuću metodu klase Parking kojoj prosleđuje broj mesta koji mu je potreban. Ako se pronađe odgovarajući broj praznih mesta reference sa tih mesta se postavljaju tako da pokazuju na to uparkirano vozilo. Ako Parking ne pronađe odgovarajući broj praznih mesta prijavljuje se greška o tome. Vozilo izlazi sa parkinga tako što poziva odgovarajuću metodu klase Parking u kojoj Parking pronalazi to vozilo i prazni sva mesta koja je to vozilo zauzimalo. Parking ima i mogućnost da prikaže trenutno stanje svojih parking mesta (da li su zauzeta ili slobodna).

Definisati nad klasom Parking indeksirani koji će da vraća automobil sa zadatim indeksima iz matrice automobila (`parking[i,j]`). U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Parking sa određenim brojem vrsta i kolona i nekoliko vozila, uparkirati ta vozila na Parking, prikazati stanje na parking, a zatim ih isparkirati sa Parkinga. U klasi Parking jednostavnosti radi ne voditi računa o ukupnjavanju praznih mesta, već se vozila smeštaju na prvo parking mesto koje zadovoljava njihove zahteve.

9. Projektovati sistem za sređivanje spiska literature na kraju neke knjige ili članka u časopisu. Klasa Literatura treba da sadrži listu stavki iz literature. Jedna stavka treba da vraća svoju godinu objavljivanja i da vraća tekst koji sadrži sve bitne podatke o njoj. Stavka u literaturi može da predstavlja knjigu i za nju se čuva naziv, lista autora, mesto i godina izdanja. Stavka u literaturi može da predstavlja članak iz časopisa i za njega se čuva naziv časopisa, naziv članka, lista autora, mesto i godina izdanja. Stavka u literaturi može da bude i web stranica i za nju se čuva samo adresa (link) i godina objavljivanja. Klasa Literatura treba da omogući sortiranje stavki po godini izdanja u rastućem ili opadajućem redosledu, sortiranje stavki po njihovoj tekstualnoj reprezentaciji u rastućem ili opadajućem redosledu (po abecedi) i prikaz svih stavki na standardnom izlazu.

Dodatno treba preklopiti operator `+=` nad listom stavki koji će da doda novu stavku u listu. U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Literatura sa određenim brojem stavki, sortirati stavke po tekstu u rastućem redosledu i prikazati sve stavke iz literature.

10. Projektovati sistem za validaciju i slanje različitih poruka sa mobilnog telefona. Klasa Telefon treba da sadrži listu poruka. Jedna poruka treba da omogući svoju validaciju i svoje slanje. Poruka može da predstavlja SMS poruku i za nju se čuva telefonski broj pošiljaoca, telefonski broj primaoca i tekst poruke. Za SMS poruku se validacijom proverava da tekst poruke ima ne više od 160 karaktera i da su brojevi telefona pošiljaoca i primaoca dužine između 9 i 13 cifara. Poruka može da predstavlja mejl poruku i za nju se čuva adresa pošiljaoca, adresa primaoca i tekst poruke. Za mejl poruku se validacijom proverava da mejl adrese pošiljaoca i primaoca sadrže obavezno karakter '@'. Poruka može da predstavlja čet poruku i za nju se čuva korisničko ime pošiljaoca, korisničko ime primaoca i tekst poruke. Za čet poruku se validacijom proverava da tekst poruke nije prazan. Za sve poruke se slanje simulira prikazom pošiljaoca, primaoca i teksta poruke na standardnom izlazu. Klasa Telefon treba da omogući validaciju (u toku validacije nevalidne poruke izbaciti iz liste poruka) i prikaz svih poruka na standardnom izlazu.

Dodatno treba preklopiti operator -= nad listom poruka koji će da obriše poruke koje ne prolaze validaciju iz liste poruka. U svim klasama iskoristiti svojstva (property) za pristup atributima klase.

U Main metodi kreirati instancu klase Telefon sa određenim brojem poruka, izbaciti nevalidne poruke i prikazati sve preostale poruke.