

Programski jezici

-Laboratorijska vežba 5-

-C#-

NAPOMENA:

Ovom laboratorijskom vežbom je obuhvaćen materijal koji je obrađen u prva tri 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. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Kolekcija koja će da pamti niz elemenata tipa Traka ili tipa izvedenog iz klase Traka. U okviru kolekcije pamtit i niz elemenata generičkog tipa. Omogućiti čitanje i upis svih informacija o Fabrici korišćenjem klase BinaryReader, odnosno BinaryWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Fabrika, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Fabrika. Kada je sadržaj fajla pročitao štampati sve podatke o Fabrici na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. Omogućiti da se ukoliko premaši kapacitet niza traka u klasi Kolekcija baca izuzetak NemaMesta.

2. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Kolekcija koja će da pamti listu (List<T>) elemenata tipa Zaposleni ili tipa izvedenog iz klase Zaposleni. Omogućiti čitanje i upis svih informacija o Kompaniji korišćenjem klase StreamReader, odnosno StreamWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Kompanija u fajl, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Kompanija. Kada je sadržaj fajla pročitao štampati sve podatke o Kompaniji na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. U klasi Kompanija definisati konstantu pod imenom

maxBonus i omogućiti da ukoliko se proračunati bonus prekorači se baci izuzetak PrevelikiBonus.

3. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Matrica koja će da pamti matricu elemenata tipa Automobil ili tipa izvedenog iz klase Automobil. Omogućiti čitanje i upis svih informacija o Sajmu korišćenjem klase StreamReader, odnosno StreamWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Sajam u fajl, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Sajam. Kada je sadržaj fajla pročitao štampati sve podake o Sajmu na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. Omogućiti da ukoliko više nema mesta na sajmu se baca izuzetak NemaMesta.

4. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Kolekcija koja će da pamti niz elemenata tipa Takmičar ili tipa izvedenog iz klase Takmičar. U okviru kolekcije pamtit i niz elemenata generičkog tipa. Omogućiti čitanje i upis svih informacija o Prvenstvu korišćenjem klase BinaryReader, odnosno BinaryWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Prvenstvo, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Prvenstvo. Kada je sadržaj fajla pročitao štampati sve podake o Prvenstvu na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. Omogućiti da se ukoliko premaši kapacitet niza takmičara u klasi Kolekcija baca izuzetak NemaMesta.

5. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Kolekcija koja će da pamti listu (List<T>) elemenata tipa Terminal ili tipa izvedenog iz klase Terminal. Omogućiti čitanje i upis svih informacija o Vremenskoj stanici korišćenjem klase StreamReader, odnosno StreamWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Vremenske stanice u fajl, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Vremenska stanica. Kada je sadržaj fajla pročitao štampati sve podake o Kompaniji na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. Ukoliko je vlažnost vazduha veća od 70% baciti izuzetak PadaćeKiša.

6. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Kolekcija koja će da pamti listu (List<T>) elemenata tipa Putovanje ili tipa izvedenog iz klase Putovanje. Omogućiti čitanje i upis svih informacija o Agenciji korišćenjem klase StreamReader, odnosno StreamWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Agencija u fajl, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Agencija. Kada je sadržaj fajla pročitan štampati sve podake o Agenciji na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. U klasi Agencija definisati konstantu koja će predstavljati maksimalnu cenu putovanja. Ukoliko je ukupna cena putovanja veća od maksimalne baciti izuzetak PutovanjePreskupo.

7. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Kolekcija koja će da pamti niz elemenata tipa Radnik ili tipa izvedenog iz klase Radnik. U okviru kolekcije pamtit i niz elemenata generičkog tipa. Omogućiti čitanje i upis svih informacija o Firmi korišćenjem klase BinaryReader, odnosno BinaryWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Firma, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Firma. Kada je sadržaj fajla pročitan štampati sve podake o Firmi na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. Omogućiti da ukoliko je plata 2 puta veća od prosečne plate baci izuzetak PlataPotencijalnoPrevelika.

8. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Matrica koja će da pamti matricu elemenata tipa Vozilo ili tipa izvedenog iz klase Vozilo. Omogućiti čitanje i upis svih informacija o Parkingu korišćenjem klase BinaryReader, odnosno BinaryWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Parking u fajl, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Parking. Kada je sadržaj fajla pročitan štampati sve podake o Parkingu na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. Omogućiti da ukoliko više nema mesta na parking u se baca izuzetak NemaMesta.

9. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Kolekcija koja će da pamti listu (List<T>) elemenata tipa Stavka ili tipa izvedenog iz klase Stavka. Omogućiti čitanje i upis svih informacija o Literaturi korišćenjem klase StreamReader, odnosno StreamWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Literatura u fajl, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Literatura. Kada je sadržaj fajla pročitao štampati sve podatke o Literaturi na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. U klasi Literatura definisati konstantu maxStavke. Baciti NemaMesta izuzetak ukoliko se proba dodavanje nove Stavke u Literaturu, a Literatura je već popunjena.

10. Nadograditi zadatak sa prethodne laboratorijske vežbe kreiranjem generičke klase Kolekcija koja će da pamti listu (List<T>) elemenata tipa Poruka ili tipa izvedenog iz klase Poruka. Omogućiti čitanje i upis svih informacija o Telefonu korišćenjem klase BinaryReader, odnosno BinaryWriter. Obavezno je testiranje ovih metoda u Main metodi, tako što se prvo upiše instanca klase Telefon u fajl, a nakon toga se pročita sadržaj fajla i smesti u novi objekat klase Telefon. Kada je sadržaj fajla pročitao štampati sve podatke o Telefonu na standardni izlaz da biste demonstrirali da sve metode rade. Čitanje i upisivanje ne sme biti obavljeno u Main-u. Prilikom rada sa fajlovima voditi računa o zatvaranju tokova. Ukoliko poruka ne prođe validaciju, baciti izuzetak NevalidnaPoruka.