Liste

1. Kreirati klasu MojaLista koja implementira jednostruko-olančanu (probati implementaciju sa dvostruko ulančanom) listu u kojoj se čuvaju cijeli brojevi. Jedini član te klase je promjenljiva glava tipa Cvor. Klasa Cvor ima sljedeće članove: vrijednost – tipa int i sljedeci – tipa Cvor (referenca na sljedeći čvor liste). Napisati konstruktor za klasu Cvor. U klasi MojaLista napisati sljedeće metode:
   1. stampajListu() koji štampa elemente liste,
   2. dodaj1() koja između svaka dva elementa u listi dodaje element koji predstavlja njihovu razliku,
   3. dodaj2(int k) koja iza svakog elementa djeljivog sa k dodaje -1,
   4. dodaj3(int a, int b) koja dodaje nakon svakog čvora čija je vrijednost veća od a, a manja od b dodaje nulu,
   5. dodaj4(int k, int v) koja nakon svakog k-tog čvora liste (k>0) dodaje čvor sa vrijednošću v. Na primjer, ukoliko je data lista: 1 2 3 4 5, k=2, v=0, rezultujuća lista treba da izgleda ovako: 1 2 0 3 4 0 5,
   6. obrisi(int d, int k) koji iz liste briše sve čvorove koji se čija je apsolutna razlika sa brojem d manja od k,
   7. pronadjiSredinu() koja vraća vrijednost čvora koji se nalazi na sredini liste. Pokušati implementaciju bez izračunavanja dužine liste. (Hint: koristiti dvije pomoćne reference),
   8. preuredi1(int k) koja okreće listu u grupama od po k elemenata k≥1. Na primjer, ako je data lista čiji su elementi: 2 4 6 8 10 nakon poziva funkcije za k=3, lista će izgledati ovako: 6 4 2 8 10,
   9. preuredi2(int x) koja mijenja datu listu tako da svi brojevi manji od x budu ispred brojeva većih od x. Na primjer, ako je data lista: 25 24 10 12 36 i x=17, nakon poziva funkcije lista treba da izgleda ovako: 10 12 25 24 36. Kako postoji više mogućih rješenja, dovoljno je kreirati barem jedno od njih.

Napomena: Maksimalan broj poena se dobija ukoliko funkcija konačan broj puta prođe datom listom i ne koristi pomoćnu memoriju (drugu listu, niz i sl.).

* 1. sortirajListu(boolean reverse) koja sortira listu. Ako je parametar reverse=true, listu treba sortirati u opadajućem poretku, inače, listu sortirati u rastućem poretku. Za maksimalan broj poena u ovom zadatku nije dozvoljeno mijenjati polje value u čvorovima liste, niti koristiti pomoćne strukture (liste, nizove i sl.).
  2. spojiListe(MojaLista t) koja sortira listu i listu t, u rastućem poretku, a zatim u listu dodaje sve elemente iz liste t tako da ona ostane sortirana. Za maksimalan broj poena u ovom zadatku nije dozvoljeno koristiti pomoćne strukture (liste, nizove i sl.).

1. Kreirati klasu Rijeci koja implementira jednostruko-olančanu uređenu listu u kojoj se čuvaju stringovi uređeni u leksikografskom poretku. Jedini član te klase je promjenljiva glava tipa Cvor. Klasa Cvor ima sljedeće članove: rijec – tipa String i sljedeci – tipa Cvor (referenca na sljedeći čvor liste). Napisati konstruktor za klasu Cvor. U klasi Rijeci, napisati sljedeće metode:
   1. stampajListu() koji štampa elemente liste
   2. dodaj(String d) – koji u listu unosi string d, tako da lista ostane uređena, ako on već ne postoji u listi (ako postoji ne unositi ga)
   3. obrisi(String d, int k) koji iz liste briše sve stringove koji razlikuju od stringa d na više od k karaktera.
2. Kreirati klasu Paralelogram koja ima dva četvočlana niza realnih brojeva x i y koji predstavljaju redom koordinate tjemena paralelograma u smjeru kretanja kazaljke sata, tj. (x[i],y[i]) je koordinata i-tog tjemena. Napisati konstruktor sa dva argumenta. Klasa Paralelogram treba da implementira interfejs Comparable tako da manjim paralelogramom smatramo onaj koji je manje površine, a ako su površine jednake, tada je manji onaj čiji je obim manji. Redefinisati metod toString tako u jednom redu ispisuje koordinate sva 4 tjemena. Na primjer: (0,0)-(0,1)-(1,1)-(1,0).

Kreirati klasu ListaParalelograma koja implementira dvostruko ulančanu uređenu listu objekata klase Paralelogram. Lista se zadaje referencom na prvi čvor. Napisati metod dodaj(Paralelogram p) za dodavanje paralelograma u listu tako da ona ostane u rastućem poretku. Napisati metod stampaj() koji štampa sadržaj liste i metod obrisi(double a, double b) koji iz liste uklanja sve paralelograme čija je površina veća od a i manja od b.

1. Napisati klase Vozac, Cvor i ListaVozaca. Klasa Vozac opisuje vozače Formule 1 i ima polja ime i tim – tipa String i pobjede i trening (nizovi cijelih brojeva, broj bodova ostvarenih na 20 trka i pozicije ostvarene na 20 treninga). Klasa Vozac implementira interfejs Comparable, na sljedeci nacin: vozač a je „veći“ od vozača b ako ima više bodova na trkama. Ako imaju jednak broj bodova, tada se posmatra broj prvih pozicija na treninzima, a ako je i to jednako, posmatra se abecedni poredak. Jednostruko olančana uređena lista vozača opisuje se pomoću klase ListaVozaca i ima polja pocetak i kraj koji su tipa Cvor i polje brojElemenata, koje je tipa int. Polje pocetak je referenca na prvi čvor liste, polje kraj je referenca na posljednji element liste, dok polje brojElemenata sadrži broj elemenata u listi. Klasa Cvor predstavlja jedan čvor liste i ima dva polja: info - tipa Vozac i sljedeci – tipa Cvor. U klasi ListaVozaca napisati sljedeće metode:
   1. dodaj(Vozac v) koja u listu dodaje vozača opisanog zapisom v tako da lista ostane uređena.
   2. brisi(String ime) koja iz liste briše čvor koji sadrži vozača čije je ime dato argumentom ime.
   3. stampaj() koja na standardni izlaz štampa imena vozača i broj pobjeda svakog od njih.
2. Kreirati klase Datum, Posao i Radnik. Klasa Datum ima tri cjelobrojna privatna polja dan, mjesec i godina. Kreirati konstruktor za ovu klasu. U konstruktoru provjeravati da li je datum ispravan. Obratiti pažnju na prestupne godine (one koje su djeljive sa 4 a nisu sa 100, a jesu sa 400. Npr. 2000, 1996 i 1892 su prestupne a 1900, 1800 i 2007 nisu prestupne). Klasa Posao ima tri privatna polja: nazivPoslodavca - tipa String, datumZaposlenja - tipa Datum i datumPrestankaZaposlenja - tipa Datum. Klasa Radnik ima polja maticniBroj - tipa String, ime – tipa String, staz – tipa int (broj godina staža), plata – tipa float i zaposlenja (niz objekata klase Posao). Sva ova polja su privatna. Kreirati konstruktore za ove dvije klase. Klasa Datum implementira interfejs Comaparable. Kreirati klasu DblLista koja predstavlja dvostruko olančanu listu objekata klase Radnik. Lista se zadaje referencom na prvi čvor. Za predstavljanje čvora liste koristiti klasu Cvor. Napisati metod za dodavanje radnika u listu, na početak liste. Napisati metod koji štampa sadržaj liste. Napisati metod koji iz liste uklanja radnika sa datim matičnim brojem. Napisati metod koji za dati datum d štampa imena i matične brojeve svih radnika koji su datuma d bili u radnom odnosu.
3. U programskom jeziku Java kreirati klasu Fajlovi koja implementira uređenu jednostruko olačanu listu u kojoj se čuvaju fajlovi. Jedini član te klase je promjenljiva glava tipa Cvor. Klasa Cvor ima članove ime\_fajla i sadrzaj tipa String i polje sljedeci tipa Cvor. Fajlovi su uređeni u leksikografskom poretku po imenima. U klasi Fajlovi napisati sljedeće metode:
   1. stampajListu() – koja štampa sve fajlove u sljedećem obliku: ime\_fajla (sadrzaj). Na primjer: 2.txt (efgh).
   2. dodajFajl(String ime\_fajla, String sadrzaj) – koja dodaje novi fajl u listu tako da ona ostane uređena.
   3. pronadjiFajlove(String tekst) – koja u listi pronalazi i štampa sve fajlove koji u svom imenu sadrže string tekst.
   4. pronadjiDuplikate() – koja u listi pronalazi i štampa fajlove koji imaju duplikat. Dva fajla se smatraju duplikatima ukoliko im je isti sadržaj.
4. U programskom jeziku Java kreirati klasu Racuni koja implementira uređenu jednostruko olačanu listu u kojoj se čuvaju računi u bankama. Jedini član te klase je promjenljiva glava tipa Cvor. Klasa Cvor ima članove vlasnik i banka tipa String, polje iznos tipa float i polje sljedeci tipa Cvor. Računi su uređeni u opadajućem poretku po iznosu. U klasi Racuni napisati sljedeće metode:
   1. stampajListu() – koja štampa sve racune u sljedećem obliku: vlasnik – banka (iznos). Na primjer: Marko - CKB (1250.00).
   2. dodajRacun(String vlasnik, String banka, float iznos) – koja dodaje novi račun u listu tako da ona ostane uređena.
   3. pronadjiRacune(String tekst1, String tekst2) – koja u listi pronalazi i štampa sve račune koji u imenu vlasnika imaju string tekst1, a u imenu banke string tekst2.
   4. pronadjiVlasnike() – koja u listi pronalazi i štampa osobe koje imaju račun u više od jedne banke.
5. U programskom jeziku Java kreirati pojednostavljenu verziju Tvitera. Napraviti klasu Korisnik koja sadrži sljedeća polja: ime\_korisnika – String, pratioci – lista objekata klase Korisnik i objave – lista stringova. U klasi Korisnik napisati sljedeće metode:
   1. prikažiPratioce() – koja štampa sve pratioce trenutnog korisnika.
   2. prikažiObjave() – koja štampa sve objave trenutnog korisnika.
   3. dodajObjavu(String tekst) – koja na kraj liste objava dodaje novu objavu sa navedenim tekstom.
   4. zaprati(Korisnik k) – koja dodaje trenutnog korisnika u listu pratilaca korisnika k.
   5. otprati(Korisnik k) – koja briše trenutnog korisnika iz liste pratilaca korisnika k.
   6. pronadjiFanove() – koja u listi pratilaca trenutnog korisnika pronalazi i štampa njegove fanove. Korisnik A je fan korisnika B ukoliko A prati B, a B ne prati A.
6. U programskom jeziku Java kreirati elektronsku verziju albuma za sličice za Svetsko prvenstvo u fudbalu. Napraviti klasu Slicica koja sadrži sljedeća polja: ime\_fudbalera – string, reprezentacija – string i broj – redni broj sličice tog fudbalera u reprezentaciji (od 1 do 20). U klasi Slicica redefinisati metodu toString tako da prikazuje sve tri informacije o jednoj sličici. Napraviti klasu Album koja sadrži dvije liste objekata klase Slicice, album i duplikati. U klasi Album napisati sljedeće metode:
   1. prikaziAlbum() – koja štampa sve sličice u albumu.
   2. prikaziDuplikate() – koja štampa sve duplikate.
   3. dodajSlicice(Slicica[] paket) – koja u album dodaje sličice iz paketa. Sličice u albumu treba da budu poređane redom po reprezentacijama (leksikografski), a unutar reprezentacije po rednim brojevima. Sličice iz paketa koje se već nalaze u albumu, dodaju se u listu duplikata.
   4. pronadjiNedostajuce() – koja pronalazi i štampa sve sličice koje nedostaju u albumu.
   5. razmijeniSlicice(Album a) – koja vrši razmjenu sličica sa albumom a. Razmjenjuju se isključivo duplikati tako što se u suprotni album dodaju duplikati onih sličica koje nedostaju tom albumu.
7. U programskom jeziku Java kreirati pojednostavljenu verziju email servera. Napraviti klasu Korisnik koja sadrži sljedeća polja: ime\_korisnika – String, email\_adresa – String, prijemno\_sanduce – lista stringova i poslate\_poruke – lista stringova. U klasi Korisnik napisati sljedeće metode:
   1. prikaziPrimljene(int pocetak, int kraj) – koja štampa primljene poruke počevši od poruke sa rednim brojem pocetak, do poruke sa rednim brojem kraj.
   2. prikaziPoslate(int pocetak, int kraj) – koja štampa poslate poruke počevši od poruke sa rednim brojem pocetak, do poruke sa rednim brojem kraj.

Napraviti klasu Server koja sadrži jednu listu objekata klase Korisnik nazvanu korisnici (0.5). U klasi Server napisati sljedeće metode:

* 1. prikaziKorisnike() – koja štampa imena i email adrese svih korisnika.
  2. sortirajKorisnike() – koja sortira listu korisnika u rastućem poretku po email adresi.
  3. otvoriNalog(String ime\_korisnika, String email\_addresa) – koja kreira novog korisnika i dodaje ga u listu korisnici. Dva korisnika ne smiju imati istu email adresu. U slučaju da je email adresa već zauzeta, odštampati da je došlo do greške.
  4. obrisiNalog(String email\_addresa) koja briše korisnika sa datom email adresom iz liste. Ukoliko takvog korisnika nema, štampati da je došlo do greške.
  5. posaljiPoruku(Korisnik posiljalac, Korisnik primalac, String tekst) koja šalje email poruku od korisnika posiljalac korisniku primalac sa sadržajem tekst, tako što tu poruku doda u poslate poruke korisnika posiljalac, a u prijemno sanduce korisnika primalac.