

[OOP] PRIPREMA ZA KOLOKVIJUM: C++ / JAVA

ŠKOLSKI PRIBOR

C++

Napisati apstraktnu klasu **OLovka** koja ima polje *cena* (tipa *double*). U klasi implementirati:

- konstruktor bez parametara – polje inicijalizovati na -1
- konstruktor sa parametrima *OLovka(double)*
- apstraktnu metodu *bool postaviCenu()*

Iz klase **OLovka** izvesti klasu **Bojica** koja ima dodatna polja *tip* (tipa *Tip*: nabrojivi tip, moguće vrednosti su *DRVENA* i *VOSTANA*) i *boja* (tipa *DinString*). U klasi implementirati:

- konstruktor bez parametara – nasleđena polja inicijalizovati na vrednosti po želji, polje *tip* treba da ima vrednost *DRVENA*, a polje *boja* *CRVENA*
 - konstruktor sa parametrima *Bojica(double, Tip, const DinString&)*
 - get metode za polje *tip* i *boja*
 - realizovati metodu za postavljanje cene – korisnik preko tastature unosi vrednost nove cene. Metoda zatim proverava da li je uneta vrednost pozitivan broj, i ako to jeste slučaj, cena se postavlja na unetu vrednost pri čemu metoda vraća *true*. U svim ostalim slučajevima povratna vrednost metode je *false*.
 - preklopiti operator ispisa – ispisuje sve informacije o bojici. Polje nabrojivog tipa obavezno ispisati rečima.
- Napisati klasu **Pernica** koja ima polja *bojice* (tipa *List<Bojica>*) i *bojiceNaAkciji* (tipa *List<Bojica>*). Pored toga, klasa sadrži još i statičko polje *drvenihBojicaNaAkciji* (tipa *int*) koje prebrojava koliko je trenutno drvenih bojica na akciji. U klasi implementirati:
 - konstruktor bez parametara – inicijalizuje liste na prazne
 - metodu *bool dodajUPonudu(const Bojica b)* – drvene bojice dodaje na kraj liste *bojice*. Voštane bojice dodaju se na početak liste *bojice*. U oba slučaja povratna vrednost metode predstavlja informaciju o (ne)uspešnom pokušaju dodavanja bojice u listu *bojice*.
 - metodu *bool akcija(const DinString&)* – metoda proverava da li u listi *bojice* postoji bojica čija je boja jednaka prosleđenoj, da li je postavljanje cene izvršeno uspešno (pozvati odgovarajuću metodu) i da li je bojica uspešno izbrisana iz liste *bojice*. Ako je to uspešno izvršeno ispisuje se poruka "Uspešna realizacija akcije.", da bi se naposljetku ista ta bojica dodala na početak liste *bojiceNaAkciji*. Povratna vrednost metode tada predstavlja informaciju o (ne)uspešnom pokušaju dodavanja bojice u listu *bojiceNaAkciji*. U svim ostalim slučajevima metoda ispisuje poruku "Akcija nije primenjena!" i vraća *false*.
napomena: potrebno je voditi računa o tome da li se povećao broj drvenih bojica na akciji (ažurirati vrednost statičkog polja ako je bojica koja se dodaje drvena).
 - preklopiti operator ispisa - ispisuje informacije o bojicama, o bojicama na akciji, kao i vrednost statičkog polja. Ako su liste prazne, ispisati odgovarajuću poruku.

Napomena: sve get metode obavezno realizovati kao nemodifikatorske.

JAVA

Napisati interfejs **IzracunavanjeCene** koji ima metodu *boolean akcijskaCena(double)*.

Napisati klasu **Bojica** koja implementira interfejs **IzracunavanjeCene** i ima polja *serijskiBroj* (tipa *String*), *cena* (tipa *double*), *boja* (tipa *String*) i *tip* (tipa *String*). U klasi implementirati:

- konstruktor sa parametrima za sva polja
- konstruktor kopije
- get metode za polja *serijskiBroj* i *cena*

- redefinisati metodu *boolean akcijskaCena(double)* – ako je prosleđen parametar sa negativnom vrednošću ili sa vrednošću 0, ispisuje se poruka “Neuspesan pokušaj smanjivanja cene!” i metoda vraća *false*. Ukoliko se ispostavi da prosleđeni parametar ima pozitivnu vrednost (radi jednostavnosti pretpostaviti da njegova vrednost neće biti veća od 99), vrši se smanjivanje cene za navedeni procenat. Nakon realizovanog smanjenja ispisuje se poruka “Cena je snizena za <<*vrednost procenta*>>%.” i metoda vraća *true*.
- redefinisati metodu *toString()*

Napisati klasu ***Pernica*** koja ima polja *bojice* (tipa *HashMap*, ključ je serijski broj bojice) i *raspolozivoMesta* (tipa *int*, vrednost je 5). U klasi implementirati:

- konstruktor bez parametara – inicijalizuje mapu na praznu
- metodu *boolean dodaj(Bojica)* – ukoliko u mapi već postoji bojica sa serijskim brojem čija je vrednost jednaka vrednosti serijskog broja bojice koja se pokušava dodati, dodavanje nije moguće i metoda vraća *false*. Ako broj raspoloživih mesta nije premašen, vrši se dodavanje prosleđene bojice u mapu i metoda vraća *true*, dok u svim ostalim slučajevima metoda vraća *false*.
- metodu *int prebroj(double)* – metoda vrši prebrojavanje bojica iz mape čija je cena veća od prosleđene. Povratna vrednost metode odgovara broju bojica koje zadovoljavaju traženi kriterijum.
- redefinisati metodu *toString()* – ako je mapa prazna, ispisati odgovarajuću poruku
- statičku metodu *main* unutar koje će biti izvršeno testiranje