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

C++

Napisati apstraktnu klasu ***Fenjer*** koja ima polja *materijal* (tipa *Materijal*, nabrojivi tip: *STIROPOR, PLASTIKA, METAL*) i *ispravan* (tipa *bool*). U klasi implementirati:

- konstruktor bez parametara – polje *materijal* postaviti na *STIROPOR*, a polje *ispravan* na *false*
- konstruktor sa parametrima *Fenjer(Materijal, bool)*
- get metode za oba polja
- apstraktnu metodu *bool popravi()*
- metodu *void ispis() const* – vrednost oba polja obavezno ispisati rečima

Iz klase ***Fenjer*** izvesti klasu ***Lampion*** koja ima dodatno polje *cena* (tipa *double*). Pored toga, klasa sadrži još i statičko polje *instanceLampiona* (tipa *int*) koje prebrojava instance klase ***Lampion***. U klasi implementirati:

- konstruktor bez parametara – polje *cena* inicijalizovati na vrednost po želji
- konstruktor sa parametrima *Lampion(Materijal, bool, double)*
- get metodu za statičko polje
- get i set metodu za polje *cena*
- realizovati metodu za popravku lampiona koja prvo proverava da li je lampion pokvaren. Ako jeste, vrši se njegova popravka promenom vrednosti odgovarajućeg polja na *true* i metoda vraća *true*. U svim ostalim slučajevima metoda vraća *false*.
- redefinisati metodu za ispis tako da dopisuje još i vrednost polja *cena*

Napisati klasu ***NovogodisnjaRasveta*** koja ima polje *lampioni* (tipa *List<Lampion*>*) i *kapacitet* (tipa *int*). U klasi implementirati:

- konstruktor bez parametara – inicijalizuje listu na praznu, a polje *kapacitet* na 5
- metodu *bool dodaj(Lampion&)* – metoda prvo proverava da li je kapacitet popunjen. Ako to jeste slučaj ispisuje se poruka “Kapacitet je popunjen!” i metoda vraća *false*. Ukoliko ima mesta za dodavanje novog lampiona, pristupa se proveri da li je lampion pokvaren – ako jeste, vrši se njegova popravka pozivom odgovarajuće metode. Ispravan lampion izrađen od stiropora dodaje se na početak liste i tada metoda vraća informaciju o (ne)uspešnom pokušaju dodavanja. Ispravni lampioni izrađeni od plastike ili od metala dodaju se isključivo na kraj liste a metoda i tada vraća informaciju o (ne)uspešnom pokušaju dodavanja lampiona u listu. U svim ostalim slučajevima povratna vrednost metode je *false*.
- metodu *void sortiraj()* – metoda na početku od korisnika zahteva da unese vrstu sortiranja (radi jednostavnosti pretpostaviti da su moguće vrednosti 0 za opadajuće i 1 za rastuće sortiranje – bilo koji drugi unos neće uticati na redosled elemenata u listi i metoda će ispisati poruku “Nevalidna vrednost vrste sortiranja!”). Sortiranje se vrši na osnovu cene lampiona. Zadatak ove metode je i da na kraju ispiše sve informacije o lampionima iz liste (ako je lista prazna, ispisati odgovarajuću poruku).
- metodu *void sprovediAkciju(double)* – metoda smanjuje cenu svih lampiona iz liste za vrednost prosleđenog procenta (radi jednostavnosti pretpostaviti da će biti prosleđena ispravna vrednost procenta)

Napomena*:* sve get metode obavezno realizovati kao nemodifikatorske.

JAVA

Napisati interfejs ***Popravka*** koji ima metodu *boolean popravi()*.

Napisati klasu ***Lampion*** koja implementira interfejs ***Popravka*** i ima polja *sifra* (tipa *int*), *boja* (tipa *String*), *ispravan* (tipa *boolean*) i *cena* (tipa *double*). U klasi implementirati:

- konstruktor sa parametrima za sva polja

- konstruktor kopije
- get metode za sva polja
- set metodu za polje *cena*
- redefinisati metodu *boolean popravi()* – metoda proverava da li je lampion pokvaren. Ako jeste, vrši se njegova popravka promenom vrednosti odgovarajućeg polja na *true* i metoda vraća *true*. U svim ostalim slučajevima metoda vraća *false*.
- redefinisati metodu *toString()* – polje logičkog tipa obavezno ispisati kao string

Napisati klasu ***NovogodisnjaRasveta*** koja ima polja *kapacitet* (tipa *int*, čija je vrednost 5) i *lampioni* (tipa *ArrayList*<*Lampion*>). U klasi implementirati:

- konstruktor bez parametara
- metodu *boolean dodaj(Lampion)* – u slučaju da je dostignut maksimalni kapacitet ili se pokušava dodati neispravan lampion, metoda vraća *false*. Ako se ispostavi da u listi već postoji lampion čija je šifra jednaka šifri lampiona koji se pokušava dodati, dodavanje nije moguće pa metoda i tada vraća *false*. U svim ostalim slučajevima lampion se dodaje u listu i metoda vraća informaciju o (ne)uspešnom pokušaju dodavanja.
- metodu *Lampion pronadji(double, String)* – vraća prvi lampion iz liste čija je cena veća od prosleđene i čija je boja jednaka prosleđenoj. Ako takav lampion ne postoji metoda vraća *null*.
- metodu *void akcija()* – metoda za 10% snižava cenu svih lampiona iz liste čija je cena veća od 1999.99 dinara
- redefinisati metodu *toString()* – ako je lista prazna, ispisati odgovarajuću poruku

Napisati klasu ***Test*** koja sadrži statičku metodu *main* unutar koje će biti izvršeno testiranje.