

Objektno programiranje, Kolokvijum, Grupa 3

Matematički fakultet

Školska godina 2017/2018

Napomena: Napraviti direktorijum /home/ispit/IdeaProjects. Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat pod imenom `oop_Ime_Prezime_Indeks_Asistent` (npr. `oop_Pera_Peric_mi12082_NM`). U napravljenom projektu, paket takođe nazvati isto tako.

Kod **ne sme** imati sintaksnih grešaka niti izbacivanje `NullPointerException`-a.

Vreme za rad: **1.5 sat**

Minimalan potreban broj poena da bi se položio kolokvijum: **5 poena**

Potrebno je napisati barem jednu klasu i testirati njen rad u test klasi.

Inicijalni asistenta: Biljana - BS, Anja - AB, Božidar - BA, Nemanja - NM

1. (5 poena) Napraviti klasu `Zvucnik` koju karakterišu atributi `proizvodjac` (`String`), `model` (`String`), `snaga` (ceo broj koji označava snagu zvučnika u vatima) i `uključen` (`boolean` indikator da li je zvučnik uključen).

Implementirati:

- Konstruktor koji prihvata vrednosti za prva tri polja, a vrednost polja `uključen` postavlja na `false`.
- Konstruktor kopije.
- Get metode za polja `proizvodjac`, `model`, `snaga` i `uključen`.
- Metod `void switchOnOff()` koji gasi odnosno pali zvučnik postavljanjem odgovarajuće vrednosti polja `uključen`.
- Metod `toString()` koji vraća nisku kao u primeru:

```
Zvucnik Yamaha HS8, snage 75W je iskljucen.  
Zvucnik Yamaha HS7, snage 60W je iskljucen.  
Zvucnik Logitech Z313, snage 25W je iskljucen.
```

Napraviti klasu `TestZvucnik` koja instancira tri navedena zvučnika i ispisuje ih na standardni izlaz.

2. (4 poena) Napraviti baznu klasu `Prostorija` koja sadrži polje `naziv` (`String`), `duzina` i `sirina` (`double` vrednosti izražene u metrima).

Implementirati:

- Konstruktor koji prihvata `naziv` prostorije, njenu `duzinu` i `sirinu`.
- Get metode za polja `naziv`, `duzina`, `sirina`.
- Metod `toString()` koji vraća nisku kao u primeru:

```
Prostorija Lotus povr sine 50.0m^2.
```

3. (6 poena) Napraviti klasu `Skladiste` koja nasleđuje klasu `Prostorija` i dodatno se karakteriše poljem `preostaloMesta` (`double` koje označava površinu skladišta koja nije zauzeta paletama) i poljem `dimenzijaPalete` (`double` dužina stranice kvadratne palete u metrima).

Implementirati:

- Konstruktor koji prihvata `naziv` skladišta, njegovu `duzinu` i `sirinu` kao i `stranicuPalete`, a polje `preostaloMesta` postavlja na površinu skladišta (`duzina*sirina`).
- Get i set metode za polja `preostaloMesta` i `dimenzijaPalete`.
- Metod `void dodajPalete(int brojPaleta)` - pokušava da doda `brojPaleta` paleta u skladište. Ukoliko su uspešno smeštene sve palete ispisuje poruku "Uspešno su smeštene sve palete!", ako je smešteno n paleta (od zadatih `brojPaleta`, `n < brojPaleta` i `n!=0`) ispisati "Uspešno je smešteno n od brojPaleta paleta.". U slučaju da nije moguće smestiti ni jednu paletu ispisati poruku "Neuspešno smeštanje, skladište je puno!".
- Metod `toString()` (pogledati primere ispod):

```
Skladiste Hangar1 povr sine 400.0m^2 (slobodno 140.0m^2).  
Skladiste Tvrđjava povr sine 1000.0m^2 (slobodno 0.0m^2).
```

4. (5 poena) Napraviti klasu `MuzickaSala` koja takođe nasleđuje klasu `Prostorija`. Osim atributa bazne klase, muzičku salu karakterišu polja `brMesta` (`int` broj numerisanih mesta za sedenje, numeracija počinje od 0), `mesta` (`boolean[]` niz koji govori da li su mesta sa datim indeksom zauzeta) i polje `ozvucenje` (`Zvucnik[]` niz zvučnika u datoj sali)

Obezbediti:

- Metod `static int brojInstanci()` koji vraća broj instanci (implementirati brojanje instanci klase).
- Konstruktor koji prima podatke o *nazivu*, *duzini* i *sirini* muzičke sale, *broju mesta* i *ozvučenju* koje se u njoj nalazi. Sva mesta su u trenutku pravljenja muzičke sale prazna (svi članovi niza *mesta* su `false`).
- Get metode za polja *brMesta*, *mesta*, *ozvucenje*.
- Metod `void zauzmiMesto(int i)` - zauzima mesto pod rednim brojem *i*. Ukoliko je mesto već zauzeto ispisati poruku "Mesto pod rednim brojem *i* je već zauzeto!".
- Metod `boolean proveriOzvucenje()` - proverava da li je snaga svih uključenih zvučnika dovoljna da pokrije površinu muzičke sale. Površina sale je pokrivena ako za svakih Pm^2 ima barem $2 \cdot P$ vati snage ozvučenja. U slučaju da su sva sedišta zauzeta, količina potrebne snage ozvučenja je za 20% veća.
- Metod `toString()` koji vraća nisku kao u primeru:

Muzicka sala Bah površine $300.0m^2$ sa 30 mesta za sedenje.

5. (5 poena) Napisati klasu `Gradjevina` koja testira rad prethodno implementiranih klasa. Potrebno je prvo napraviti niz zvučnika (ručno ih instancirati):

Proizvođač: Yamaha; Model: HS8; Snaga: 75
 Proizvođač: Yamaha; Model: HS7; Snaga: 60
 Proizvođač: Logitech; Model: Z313; Snaga: 25

Zatim treba napraviti jedno skladište. Nakon unosa ispisati podatke o skladištu.

Unesite naziv skladista:
 Hangar
 Unesite duzinu skladista:
 11
 Unesite sirinu skladista:
 5
 Unesite stranicu palete:
 2

Skladiste Hangar površine $55.0m^2$ (slobodno $55.0m^2$).

U ovo skladište dodati 8 paleta, zatim 4 i posle toga 2 palete. Program bi trebalo da ispiše redom:

Uspešno su smestene sve palete!
 Uspešno je smesteno 2 od 4 paleta.
 Neuspešno smestanje, skladište je puno!

Napraviti muzičku salu. Kao ozvučenje konstruktoru proslediti prethodno napravljen niz zvučnika. Nakon unosa ispisati podatke o muzičkoj sali.

Unesite naziv muzičke sale:
 Schumann
 Unesite duzinu muzičke sale:
 8
 Unesite sirinu muzičke sale:
 10
 Unesite broj mesta:
 10

Muzicka sala Schumann površine $80.0m^2$ sa 10 mesta za sedenje.

Uključiti sve zvučnike u nizu *ozvucenje* i zatim pozvati metod `proveriOzvucenje()` i u slučaju da vrati `true` ispisati "Dobro ozvuceno!" u suprotnom ispisati poruku "Lose ozvuceno!".

Dobro ozvuceno!

Zatim popuniti svih 10 mesta za sedenje i pozvati metod `proveriOzvucenje()` ponovo.

Lose ozvuceno!