

Prvi regularni kolokvijum iz objektno-orijentisanog programiranja

Datum: 1.12.2024.

Paket izuzeci

Tip **NedozvoljenaVrednostException** koji je izuzetak.

- Konstruktor prima referencu na hotel i celobrojnu promenljivu koji predstavlja kapacitet. Konstruktor postavlja poruku izuzetka u sledećem formatu: Kapacitet {trazeni kapacitet} prevazilazi dozvoljenu vrednost u hotelu {string reprezentacija hotela}.

Tip **NemaDovoljnoNovcaException** koji je izuzetak.

- Konstruktor prima referencu na korisnika i smestaj. Konstruktor postavlja poruku izuzetka u sledećem formatu: Korisnik {ime} nema dovoljno novca da rezervise smestaj - {string reprezentacija smestaja}.

Tip **NemaSlobodnihException** koji je izuzetak.

- Konstruktor prima referencu na korisnika. Konstruktor postavlja poruku izuzetka u sledećem formatu: Korisnik ne moze vise da vrsi rezervacije - {string reprezentacija korisnika}.

Tip **RezervacijaNijeMogućaException** koji je izuzetak.

Paket smeštaj

Apstraktni tip **IUporediv** koji ima sledeća ponašanja:

- Javnu metodu **uporedi** koja prihvata Object i vraća true/false povratnu vrednost.

Apstraktni tip **Smeštaj** koji je **uporediv** i ima sledeća stanja i ponašanja:

- Privatnu string promenljivu **naziv**.
- Privatnu double promenljivu **cenaPoDanu**.
- Privatnu boolean promenljivu **dostupan**.
- Konstruktor koji prihvata naziv i cenuPoDanu (dostupnost je podrazumevano setovana na true).
- Getter i setter za promenljivu **dostupan**.
- Javnu metodu **dajCenu** koja prihvata celobrojnu promenljivu broj dana i vraća double vrednost. Ukupna cena smeštaja računa se kao proizvod broja dana i cene po danu.
- Javnu metodu **rezerviši** koja omogućava rezervaciju smeštaja za korisnika. Metoda prihvata korisnika i celobrojnu promenljivu broj dana. Ako je smeštaj zauzet, metoda vraća false. U suprotnom, smeštaj se dodaje u listu rezervacija korisnika, dostupnost smeštaja se menja na zauzet, i metoda vraća true. Ukoliko rezervacija ne uspe, jer nije moguće dodati smeštaj, ispisuje se poruka o grešci i metoda vraća false.
- Prepisanu metodu **uporedi** koja poredi da li su dva smeštaja jednaka. Smeštaji su jednaki ukoliko imaju iste nazive, cenePoDanu i dostupnost. Smeštaj se može porediti samo sa smeštajem, u suprotnom se vraća false vrednost.
- Prepisanu metodu **toString** koja vraća string oblika: naziv - (cenaPoDanu).

Tip **Apartman** koji je **smeštaj** i ima sledeća stanja i ponašanja:

- Privatnu boolean promenljivu **kuhinja** (prisustvo/odsustvo).
- Konstruktor koji prihvata naziv, cenu i kuhinju.
- Prepisanu metodu **dajCenu** koja računa cenu Apartmana za prosleđeni broj dana. Ukoliko se kuhinja nalazi u apartmanu, tada je cena 10 puta veća od osnovne cene smeštaja, a u suprotnom 5 puta veća.
- Prepisanu metodu **toString** koja vraća string oblika: [Apartman: naziv - (cena)].

Tip **HotelskaSoba** koja je **smeštaj** i ima sledeća stanja i ponašanja:

- Privatnu celobrojnu promenljivu **brojKreveta**.
- Konstruktor koji prihvata naziv, cenu i brojKreveta.
- Konstruktor koji prihvata naziv i cenu, a podrazumevano važi da je soba dvokrevetna.
- Prepisanu metodu **dajCenu** koja računa cenu Sobe za prosleđeni broj dana. Ukoliko je soba jednokrevetna ili dvokrevetna, tada je cena 2 puta veća od osnovne cene smeštaja, a u slučaju da soba sadrži više od dva kreveta, tada je cena 4 puta veća.
- Prepisanu metodu **toString** koja vraća string reprezentaciju hotelske sobe u vidu: [HotelskaSoba: (naziv - cena)].

Tip **Hotel** koji je **uporediv** i ima sledeća stanja i ponašanja:

- Privatnu string promenljivu **naziv**.
- Privatnu celobrojnu promenljivu **brojZvezdica**.
- Privatni niz smeštaja.
- Konstruktor koji prihvata naziv, kapacitet i brojZvezdica. Hotel može da ima maksimalno 5 smeštajnih objekata. Ukoliko se prosledi vrednost koja je veća od 5, baciti izuzetak **NedozvoljenaVrednostException**.
- Getter i setter za promenljivu brojZvezdica.
- Setter za niz smestaja.
- Prepisanu metodu **uporedi** koja poredi da li su dva hotela jednaka. Hoteli su jednaki ukoliko imaju iste nazive i brojZvezdica. Hotel se može porediti samo sa hotelom, u suprotnom se vraća false vrednost.
- Javnu metodu **dajNajjeftinijiSmestaj** koja prihvata celobrojnu promenljivu broj dana i vraća najjeftiniji smestaj u hotelu za prosledjeni broj dana.
- Javnu metodu **zauzmiSmeštaj** koja prihvata korisnika i celobrojnu promenljivu brojDana, a povratna vrednost je Smestaj. Potrebno je izvršiti rezervaciju slobodnog smeštaja u hotelu. Ukoliko je rezervacija moguća, vratiti rezervisani smestaj. U slučaju da nije moguće rezervisati nijedan od smeštaja u hotelu, baciti izuzetak **RezervacijaNijeMogućaException**.
- Prepisanu metodu **toString** koji vraća string reprezentaciju naziva hotela i svih smeštaja koje hotel nudi.

Paket korisnik

Tip **Korisnik** koji ima sledeća stanja i ponašanja:

- Privatnu string promenljivu **ime**
- Privatnu double promenljivu **budžet**.
- Privatni niz rezervisanih smeštaja.
- Konstruktor koji prihvata ime, budžet i kapacitet niza.
- Gettere za promenljive ime i rezervisane smestaje.
- Javnu metodu **dodajRezervaciju** koja prihvata smeštaj i celobrojnu promenljivu broj dana, a povratni tip je void. Metoda prvo proverava da li korisnik ima dovoljno novca da izvrši rezervaciju smeštaja. Ukoliko nema dovoljno novca, metoda baca izuzetak `NemaDovoljnoNovcaException`. U suprotnom, smanjuje budžet korisnika i dodaje smeštaj u niz na prvo slobodno mesto. Ukoliko nema mesta u nizu, baciti grešku `NemaSlobodnihException`.
- Javnu metodu **izaberiHotel** koja prihvata niz hotela i vraća hotel sa najvećim brojem zvezdica. U slučaju da postoji više hotela koji dele isti najveći broj zvezdica, metod može vratiti bilo koji od njih.
- Prepisanu metodu **toString** koja vraća string reprezentaciju korisnika i svih njegovih rezervacija u obliku: `[Korisnik (ime – budžet)] [smestaj 1][smestaj 2]`.

Default-ni paket

Klasa **Test** koja ima sledeća stanja i ponašanja:

- Napraviti četiri hotela (jedan hotel treba da ima kapacitet smestajnih objekata veci od 5).
- Napraviti jednog korisnika.
- Dodati po tri smestaja u svaki napravljeni hotel.
- Korisnik treba da izabere hotel sa najvećim brojem zvezdica i da izabrani hotel ispiše.
- Korisnik treba da rezervise smestaj u prethodno izabranom hotelu u trajanju od dva dana.
- Nakon rezervacije proveriti da li je rezervisani smestaj ujedno i najjeftiniji i ispisati komentar o tome.
- Ispisati string reprezentacije svih smestaja koje je korisnik rezervisao.
- Ispisati broj hotelskih soba koje je korisnik rezervisao.

```
public class Test {  
  
    public static void main(String[] args) {  
  
        Hotel[] hoteli;  
  
        // nizu hotela dodaj  
  
        // new Hotel("Hilton", 2, 3);  
  
        // new Hotel("Sumarice", 1, 1);  
    }  
}
```

```

// new Hotel("Kragujevac", 3, 5);

// new Hotel("Sloboda", 6, 5);

Korisnik korisnik = new Korisnik("Marko Nikolic", 870.0, 3);

Smestaj[] hilton_smestaji;

Smestaj[] sumarice_smestaji;

Smestaj[] kragujevac_smestaji;

// Hotelu Hilton dodeli sledece smestaje

// new Apartman("A-1", 200.0, false);

// new Apartman("A-2", 50.0, false);

// new HotelskaSoba("S-1", 50.0, 1);

// Hotelu Sumarice dodaj sledece smestaje

// new HotelskaSoba("S-1", 20.0, 2);

// new Apartman("A-1", 30.0, true);

// new HotelskaSoba("S-3", 40.0, 1);

// Hotelu Kragujevac dodaj sledece smestaje

// new HotelskaSoba("S-1", 60.0, 1);

// new HotelskaSoba("S-2", 80.0, 3);

// new Apartman("A-1", 30.0, true);

// koriscenjem foreach petlje ispisi string reprezentacije svih hotela

// Korisnik treba da izabere hotel sa najvećim brojem zvezdica i da ispise njegovu
string reprezentaciju.

// Korisnik rezervise smestaj u prethodno izabranom hotelu u trajanju od 2 dana

// Nakon rezervacije proveriti da li je rezervisani smestaj ujedno i najjeftiniji i ispisi
komentar o tome

// ispisi string reprezentacije svih smestaja koje je korisnik rezervisao

// ispisi broj hotelskih soba koje je korisnik rezervisao

}

}

```

Napomena: Pri hvatanju izuzetaka, neophodno je ispisati poruku o grešci.