

I parcijalni ispit (varijanta A)

Ukupno bodova: 20 (Bodovi će se dodijeliti proporcionalno broju uspješnih testova.)

Na repozitoriju se nalazi gotov projekat koji sadrži samo praznu Main klasu i testove. Vaš zadatak je da napravite kompletan Java program koji zadovoljava postavku zadatka i prolazi testove.

Svojim potpisom student izjavljuje da je saglasan sa ovim sistemom bodovanja i da se rješenje postavljenog zadatka nalazi na serveru kako je objašnjeno u uputama za izradu ispita.

Zadatak 1:

Potrebno je omogućiti korisniku da koristi *kalendar* kako bi vodio evidenciju o svim aktivnostima koje ima u danu, upisivao događaje, dešavanja, rođendane i sl. Za potrebu toga implementirajte klasu **Događaj** koji je opisan **nazivom**, **datumom** i **vremenom početka** te **datumom** i **vremenom kraja** (tipa `LocalDateTime`).

- Klasa treba slijediti JavaBean konvenciju.
- Ukoliko se pokušaju zadati ili naknadno postaviti početak i kraj događaja takvi da je početak nakon kraja potrebno je baciti izuzetak tipa **NeispravanFormatDogađaja** sa porukom "Neispravan format početka i kraja događaja".
- Događaj može biti događaj **visokog**, **srednjeg** i **niskog** prioriteta.

Zadatak 2:

Zatim implementirajte klasu **Kalendar** koja je obična kontejnerska klasa u kojoj se vodi evidencija o svim događajima koje je korisnik unio. Klasa **Kalendar** se može ispisati na ekran u formatu:

```
mojDogađaj1 (nizak prioritet) - početak: dd/mm/yyyy (hh:mm), kraj: dd/mm/yyyy (hh:mm)
mojDogađaj2 (visok prioritet) - početak: dd/mm/yyyy (hh:mm), kraj: dd/mm/yyyy (hh:mm)
...
```

Pored toga, klasa treba imati sljedeće metode:

- **zakaziDogađaj** - zakazuje događaj proslijeđen kao parametar u korisnikov kalendar
- **otkaziDogađaj** - otkazuje događaj proslijeđen kao parametar iz korisnikovog kalendar
- **zakaziDogađaje** - zakazuje kolekciju događaja proslijeđenu kao parametar
- **otkaziDogađaje** - otkazuje sve događaje iz kolekcije proslijeđene kao parametar
- **otkaziDogađaje** - verzija prethodne metode koja otkazuje sve događaje koji ispunjavaju kriterij poslan u metodu
- **dajKalendar** - vraća listu zakazanih događaja
- **dajKalendarPoDanima** - vraća mapu elemenata čiji je ključ datum (`LocalDate`) a pridružena vrijednost lista svih događaja koji počinju na taj dan

- **dajSljedeciDogadjaj** - metoda koja vraća prvi događaj koji će nastupiti nakon datuma poslanog u funkciju (LocalDateTime). Ako ne postoji takav događaj metoda treba baciti izuzetak IllegalArgumentException sa porukom “Nemate događaja nakon navedenog datuma”.

Zadatak 3:

Klasa Kalendar treba da implementira interfejs **Pretrazivanje** koji omogućuje korisniku naprednu pretragu kalendara kako bi se poboljšala efikasnost. Metode koje interfejs propisuje su:

- **filtrirajPoKriteriju** - metoda koja vraća listu događaja iz kalendara koji ispunjavaju kriterij proslijeđen kao parametar metode. Ovu metodu obavezno napišite koristeći streamove. Ovo će biti ručno pregledano i drugačije implementacije se neće priznavati.
- **dajDogadjajeZaDan** - metoda koja vraća listu događaja koji počinju na dan specificiran parametrom metode
- **dajSortiraneDogadjaje** - metoda koja vraća listu događaja sortiranih koristeći komparatorsku funkciju koja joj se šalje kao parametar (tj. funkciju koja prima dva događaja i vraća cijeli broj - 1, 0 ili 1 u zavisnosti od njihovog poretka)
- **dajsortiranePoPrioritetu** - metoda koja vraća skup događaja sortiranih po prioritetu od visokog do niskog prioriteta. Događaji istog prioriteta se međusobno sortiraju alfabetski u rastućem poretku.
- **daLiSamSlobodan** - metoda koja govori korisniku da li je slobodan na dan koji se proslijedi kao parametar. Korisnik je slobodan ukoliko nema ni jedan događaj koji traje u toku tog dana.
- **daLiSamSlobodan** - metoda koja govori korisniku da li je slobodan u vremenskom intervalu izmedju dana koji se proslijede kao parametar. Ukoliko proslijeđeni parametri nisu takvi da je prvi datum prije drugog datuma baciti izuzetak tipa IllegalArgumentException sa porukom “Neispravni podaci o početku i kraju”

Sarajevo, 7. 12. 2019

Vedran Zuborić

I parcijalni ispit (varijanta B)

Ukupno bodova: 20 (Bodovi će se dodijeliti proporcionalno broju uspješnih testova.)

Na repozitoriju se nalazi gotov projekat koji sadrži samo praznu Main klasu i testove. Vaš zadatak je da napravite kompletan Java program koji zadovoljava postavku zadatka i prolazi testove.

Svojim potpisom student izjavljuje da je saglasan sa ovim sistemom bodovanja i da se rješenje postavljenog zadatka nalazi na serveru kako je objašnjeno u uputama za izradu ispita.

Zadatak 1:

Potrebno je razviti sistem za vođenje evidencije o rezervacijama sala na fakultetu. Za potrebe ovog sistema kreirajte klasu **Rezervacija** koja je opisana atributima **nazivSale**, **predavac**, **pocetak** i **kraj**. Početak i kraj predstavljaju vrijeme u danu kada je sala zauzeta (npr. od 09:00 do 12:00 tipa `LocalTime`).

- Klasa treba slijediti JavaBean konvenciju.
- Ukoliko se pokušaju zadati ili naknadno postaviti početak i kraj termina rezervacije takvi da je početak nakon kraja potrebno je baciti izuzetak tipa **NeispravanFormatRezervacije** sa porukom "Neispravan format početka i kraja rezervacije".
- Rezervacija može biti **periodična** i **vanredna**.
 - Periodična rezervacija čuva i informaciju o tome za koji dan u sedmici se rezerviše sala (broj od 0 do 6 koji označava dane od ponedjeljka do nedjelje). Ukoliko se kroz konstruktor ili setter ove klase pokuša postaviti dan koji je van tog opsega potrebno je baciti izuzetak `IllegalArgumentException` sa porukom ("Neispravan dan u sedmici").
 - Vanredna rezervacija čuva informaciju o tome za koji datum se rezerviše sala (`LocalDate`)

Zadatak 2:

Pored toga razvijte i kontejnersku klasu **Evidencija** koja čuva informacije o svim rezervacijama za taj fakultet. Objekat ove klase se može ispisati na ekran u formatu:

```
sala1 - predavac1 (periodična) - pocetak: hh:mm, kraj: hh:mm, svakog ponedjeljka  
sala2 - predavac2 (vanredna) - pocetak: hh:mm, kraj: hh:mm, na dan dd/mm/yyyy  
...
```

Pored toga, klasa treba imati sljedeće metode:

- **rezervisi** - evidentira rezervaciju koja se prosljeđuje kao parametar ukoliko je sala slobodna za taj termin (tj. za vanredne rezervacije ne postoji ni jedna rezervacija za taj dan, a za periodične ni jedna rezervacija koja pada na taj dan)
- **rezervisi** - evidentira listu rezervacija koja se prosljeđuje kao parametar
- **otkaziRezervaciju** - otkazuje rezervaciju koja se prosljeđuje kao parametar

- **otkaziRezervacije** - otkazuje listu rezervacija koja se prosljeđuje kao parametar
- **otkaziRezervacije** - verzija prethodne metode koja otkazuje sve rezervacije koji ispunjavaju kriterij poslan u metodu
- **dajEvidenciju** - vraća evidenciju o svim rezervacijama sala u obliku mape elemenata čiji je ključ naziv sale a pridružena vrijednost lista svih rezervacija za tu salu
- **dajSveRezervacije** - metoda koja vraća listu svih rezervacija o kojima se vodi evidencija
- **dajEvidencijuZaSalu** - vraća listu svih rezervacija za salu čiji naziv se prosljeđuje kao parametar. Ukoliko za tu salu ne postoje zauzeća treba baciti izuzetak tipa `IllegalArgumentException` sa porukom "Za salu 123 ne postoje evidentirane rezervacije" gdje je 123 naziv sale.

Zadatak 3:

Klasa Evidencija treba da implementira interfejs **Pretrazivanje** koji omogućuje sistematičan način za pretragu evidencije kako bi se lakše pronašle slobodne sale. Ovaj interfejs propisuje sljedeće metode (*sve metode koje slijede i primaju naziv sale trebaju da bacaju isti izuzetak kao i dajEvidencijuZaSalu*):

- **filtrirajPoKriteriju** - metoda koja vraća listu rezervacija koje ispunjavaju kriterij prosljeđen kao parametar metode. Ovu metodu obavezno napišite koristeći streamove. Ovo će biti ručno pregledano i drugačije implementacije se neće priznavati.
- **dajRezervacijeZaDan** - metoda koja prima dva parametra, naziv sale i datum, a vraća listu svih rezervacija te sale za taj datum. Ovo uključuje sve vanredne rezervacije sale za taj datum i sve periodične rezervacije koje padaju na taj datum.
- **dajSortiraneRezervacije** - metoda koja prima dva parametra, naziv sale i kriterij sortiranja (tj. funkciju koja prima dvije rezervacije i vraća cijeli broj -1, 0 ili 1 u zavisnosti od njihovog poretka), a vraća tako sortiranu listu rezervacija za tu salu.
- **dajSortiranePoTipu** - metoda koja vraća skup svih rezervacija za salu čije ime se prosljeđuje kao parametar sortiran po tipu (prvo periodične pa vanredne), dok će se periodične rezervacije sortirati između sebe po danu u sedmici (od 0 do 6), a vanredne po datumu (prvo najranije)
- **dajRezervacijeNakon** - metoda koja vraća listu svih vanrednih rezervacija za salu čije se ime prosljeđuje kao parametar koje padaju na datum nakon onog poslanog kao drugi parametar metode (ne uključujući njega)
- **daLiJeSlobodna** - metoda koja prima četiri parametra: naziv sale, datum, vrijeme početka i vrijeme kraja rezervacije. Ova metoda treba dati odgovor na pitanje da li je sala slobodna za rezervaciju u tom terminu. (*Obratite pažnju da je sala slobodna tada samo ako ne postoji preklapanje sa bilo kojim terminom koji je već rezervisan. Dozvoljeno je da novi termin počne odmah po završetku drugog termina tj. ukoliko je sala rezervisana od 09:00 do 12:00 ona je slobodna za rezervaciju u terminu 12:00 do 14:00*). Metoda baca izuzetak `IllegalArgumentException` ako se prosljede početak i kraj koji nisu validni (tj početak nije prije kraja) sa porukom "Neispravni početak i kraj rezervacije".

Sarajevo, 7. 12. 2019

Vedran Zuborić