

1. kolokvij iz kolegija Java, 30.04.2021. - praktični dio (90 minuta)

Rješenja pošaljite na mail adresu java@math.hr sa svog **studentskog maila** uz navedeno ime, prezime i JMBAG. Dozvoljeno je **samo** korištenje okruženja Apache NetBeans, a ukoliko automatsko dopunjavanje u NetBeans-u **ne radi**, i materijala na poveznici <https://docs.oracle.com/en/java/javase/15/docs/api/java.base/module-summary.html>.

Ime i prezime:

JMBAG:

1. (3+5+6 bodova) Implementirajte generičku klasu **Par** koja ima dva parametra tipa **T** (proizvoljni tip) i **S** (klasa koja mora biti izvedenica broja). Implementirajte konstruktor bez parametara, konstruktor koji prima vrijednosti elemenata **x** i **y** para (**x,y**), metode za dohvat/postavljanje vrijednosti komponenti. Napravite da rade naredbe **Par<Character, Integer> p = new Par<>('C', 10); System.out.println(p);** // treba ispisati (C , 10).

Implementirajte generičku klasu **Red** koja ima jedan parametar tipa **T** (klasa koja je izvedenica ranije implementiranog generičkog para). Klasa simulira rad reda koristeći polje za spremanje elemenata. **Red<T> r;** //definira prazno polje duljine 100

Omogućite da se red može definirati i neovisno o broju parova, kao: **Par<Character, Integer> p = new Par<>('A',1), p1 = new Par<>('B',2), p2 = new Par<>('C',3); Red<Par<Character,Integer>> r = new Red<>(p1,p2,p3);** //sad red sadrži 3 elementa p, p1, p2 koji su dodani u red redosljedom definiranim u konstruktoru. Implementirajte funkcije **public void enqueue(T element); public T dequeue();** koje dodaju element na kraj reda (ukoliko nema dovoljno memorije u polju, polje treba proširiti s dodatnih 100 elemenata), odnosno uklanjaju element s početka reda. Omogućite ispisivanje reda i pretvorbu u listu **public List<T> toList();** Definirajte funkciju koja vraća broj elemenata u redu.

Definirajte sučelje **Operacije** koje ima deklaraciju sljedećih funkcija:

```
public double suma(); //vraća sumu vrijednosti p.y svih parova p u redu
public double prosjek(); // vraća prosjek vrijednosti p.y svih parova p u redu
public Par<Object, Integer> najbrojniji(); //vraća par (x,y), za x koji se najviše puta javlja u redu. x je prvi element para (Objekt) a y (Integer) broj pojavljivanja tog elementa u redu (smijete koristiti kontejner iz java.util).
public T najmanjiPrioritet(); //vraća par (x,y) tipa T koji ima najmanji prioritet y.
```

Implementirajte generičku klasu **PrioritetniRed** koja nasljeđuje klasu **Red**, implementira sučelje **Operacije** i koju više ne možemo nasljeđivati. Definirajte konstruktore (konstruktor koji prima više parametara implementirajte koristeći funkciju **enqueue**) te nadjačajte funkciju **public void enqueue(T element);** tako da dodaje elemente u red poštujući prioritet definiran kao drugi element para element (element s najvišim prioritetom je prvi u redu).

2. (6 bodova) Napravite novi projekt i dodajte prethodnu implementaciju kao biblioteku. Definirajte barem 4 različita para (neka barem 2 imaju istu prvu komponentu) te ih ubacite u red i u prioritetni red te ispišite oba. Pretvorite prioritetni red u listu, te omogućite da radi sljedeća naredba:

```
List<Par<Character,Integer>> l = pred.toList(); l.forEach(new PromijeniPrioritet(2));
Ukoliko smo imali red (('a',4), ('b',3), ('c',2), ('d',1)), tada nakon poziva imamo red (('a',16), ('b',9), ('c',4), ('d',1)) //dakle potencirali smo svaki prioritet potencijom 2 (potencija je parametar koji se definira kao u primjeru). Napravite funkciju public static void ispisi i definirajte ArrayList s ispravnim argumentom tipa tako da radi:
```

```
ArrayList<argument_tipa> sve = new ArrayList<>();
sve.add(red); sve.add(pred); //red je običan red a pred je prioritetni red
ispisi(sve); //ispisuje oba reda, svaki u svom retku
Zapišite oba reda u datoteku „izlaz.txt”.
```