

Java test [osnove programiranja] (23.05.2022.)

Napomene i uputstva

- Vreme izrade testa je 180 minuta.
- Napraviti paket sa imenom **JavaTest** unutar *src* foldera.
- Imena funkcija su **boldovana** i one moraju da su imenovane **IDENTICNO** kao sto je naglaseno.
- Poredak argumenata funkcija je takodje bitan tako da obratite paznju i na to!
- Za svaki zadatak napraviti posebnu kalsu i nazvati ih *Zadatak1*, *Zadatak2* itd. Sve funkcije jednog zadatka pisati u **istoj** klasi! Kada zavrсите sve zadatke (i proverite ih!) dizete ruku na Zoom-u (tri tackice -> reactions -> raise hand) i cekate da vas prozovem. Kada vas prozovem, zipujete predamnom projekat i postavljate ga na vas licni folder na drajvu.
- Tek kada kazem da je sve ok mozete da gasite kameru i da zaustavite share!**
- Ukoliko je *x* neka promenljiva, tada se sa *x* oznacava **vrednost** te promenjive,
- a sa *<x>* oznacava **ime** te promenjive. Npr za **int x = 5;**:

- Ispisati porku "Konacno resenje je x" --> "Konacno resenje je 5"
- Ispisati poruku "Vrednost od <x> je x" --> "Vrednost od x je 5"

Zadatak 1

Bowser je ponovo kidnapovao sve Nintendo princeze i zarobio ih u svoje zamkove. Toad je uspeo da sazna koje sve princeze su zarobljene u zamkovima i za svaki zamak je napravio spisak imena princeza u njemu. Mario nema vremena da spasava sve princeze, te se fokusira samo na spasavanje princeza Peach i Daisy. Pomozite Mario-u da odredi da li se na spisku zarobljenih princeza u zamku nalaze Peach ili Daisy.

Napisati funkciju **princezaUZamku** koja prima niz *Stringova* imena princeza i proverava da li se u tom nizu nalazi ime *"Peach"* ili ime *"Daisy"*

#	Input	Output
1	["Rosalina", "Peach", "Zelda", "Lucina", "Daisy", "Midna"]	true
2	["Peach", "Zelda", "Peach"]	true
3	["Zelda", "Midna"]	false
4	["Daisy"]	true
5	[]	false
6	["Midna", "peach", "LUCINA"]	true

Zadatak 2

Profesoru matematike se nesvidja trenutni sistem ocenjivanja. Zato je on osmislio da daje svojim djacima ocene od 1.0 do 5.0, ukljucujuci i sve medjuvrednosti poput 3.34. Stoga je odlucio da promeni i nacin zakljucivanja ocena, ali nikako ne moze da se opredeli koju sredinu da koristi. Pomozite profesoru tako sto cete mu napisati funkcije koje racunaju razlicite sredine prosljednog niza ocena:

- AR** koja racuna aritmeticku sredinu ocena

#	Input	Output
1	[5.0, 4.3, 2.7, 1.9, 4.45]	3.67
2	[2.222, 3.333, 4.444]	3.33
3	[5.0, 5.0, 5.0, 5.0]	5.0
4	[]	0

- GM** koja racuna geometrijsku sredinu ocena.
- Geometrijska sredina brojeva se racuna kao n-ti koren proizvoda svih ocena, gde je n broj ocena. n-ti koren racunati pomocu *Math.pow(broj, 1.0/n)*

#	Input	Output
1	[5.0, 4.3, 2.7, 1.9, 4.45]	3.45
2	[2.222, 3.333, 4.444]	3.20
3	[5.0, 5.0, 5.0, 5.0]	5.0
4	[]	0

- HM** koja racuna harmonijsku sredinu ocena.
- Harmonijska sredina ocena se racuna kao odnos broja ocena i sume reciprocnih vrednosti ocena. Reciprozna vrednost nekog broja n je 1/n;

#	Input	Output
1	[5.0, 4.3, 2.7, 1.9, 4.45]	3.21
2	[2.222, 3.333, 4.444]	3.07
3	[5.0, 5.0, 5.0, 5.0]	5.0
4	[]	0

Pored toga, profesor zeli da bude sto vise fer pa zeli da svakog djaka ocenjuje onom sredinom koja ce mu dati najveći prosek. Napisati funkciju **kojuSredinuKoristiti** koja prima niz ocena djaka i vraća odgovarajacu poruku.

- "AR"; ako je aritmeticka sredina strogo veca od preostale dve
- "GM"; ako je geometrijska sredina strogo veca od preostale dve
- "HM"; ako je harmonijska sredina strogo veca od preostale dve
- "Nebitno"; ukoliko su sve tri jednake
- Ukoliko su dve jednake i vecce od trece, onda vratiti onu koja je prva u poretku AR - GM - HM (Ako je HM = GM > AR, vratiti poruku "GM")

#	Input	Output
1	[5.0, 5.0, 5.0]	"AR"
1	[1.0, 2.1, 3.3, 4.1, 5.0]	"AR"
1	[]	"AR"

Zadatak 3

Vas zadatak je da napravite software za EasternUnion menjacnicu. EasternUnion menjacnice posluju samo sa valutama dalekog istoka. Te valute su *yuan*, *yen*, *won* i *dong*. Svaka menjacnica ima novac sa kojim raspolaze. Sve valute se prosledjuju kao *String*, a vrednosti novca kao **double**. Potrebno je napisati funkciju:

- konvertuj** koja prvo prima podatak **u** koju valutu se konvertuje, zatim **iz** koje valute se konvertuje i na kraju **koliko** novca se konvertuje, a vraća podatak koliko je dobijeno novca izabrane valute

Vrednost valuta je:

- 1 yuan = 19.11 yen
- 1 yuan = 190.31 won
- 1 yuan = 3461.83 dong

Pretpostaviti da su vrednosti simetricne (1-1) i tranzitivne i na taj nacin odredite kako izgledaju ostale konverzije.

#	Input	Output
1	"Yuan", "Yen", 123.0	6.436
2	"Dong", "Yen", 5000.0	905763.997
3	"WON", "dong", 100.0	5.497
5	"Yen", "Yen", 434.3	434.3
6	"Yen", "Won", 0.0	0.0

Zadatak 4

Napisati funkciju **najduziStringizTriNiza** koja prima tri *String* niza i vraća najduzi *String* iz ta 3 niza. Pretpostaviti da je takva niska jedinstvena, kao i da je duzina svakog niza >=1.

#	Input	Output
1	["Niska jedan", "Niska dva", "Niska t"], ["123", "12", "1"], ["qwe", "abc"]	"Niska jedan"
2	["abc", "0123456789"], ["ABCDEFG", "aBcDEfG"], ["defg"]	"0123456789"
3	[""], ["ABCD"], ["abcde"]	"abcde"

Zadatak 5

Napisati naredne funkcije:

- dupliFaktorijel** koja prima **long** pozitivan ceo broj i vraća njegov dupli faktorijel (oznaka !!). Dupli faktorijel se racuna kao:
 - n!! = n * (n-2) * (n-4) * ... * (1 ili 2, u zavisnosti odakle se kreće)

#	Input	Output
1	5	15
2	6	48
3	1	1
4	2	2
5	10	3840

- xProcenatY** koja prima dva nenegativna cela broja x i y, oba ne veca od 100, i vraća odgovarajucu poruku:
 - "<x> %<y> je vecce od <y> %<x>"; ako je x *procenat od y strogo vecce* od y *procenat od x*
 - "<x> %<y> je manje od <y> %<x>"; ako je x *procenat od y strogo manje* od y *procenat od x*
 - "<x> %<y> je jednako <y> %<x>"; ako je x *procenat od y isto* kao y *procenat od x*

Napomena: Znak % se interpretira kao procenat, a ne moduo (ostatak prilikom deljenja)!

#	Input	Output
1	25, 50	"<x> %<y> je jednako <y> %<x>"
2	10, 100	"<x> %<y> je jednako <y> %<x>"
3	16, 25	"<x> %<y> je jednako <y> %<x>"

- prvihNFibonacci** koja prima pozitivan ceo broj n i ispisuje u *jednom redu* prvih n brojeva u Fibonacijevom nizu.

Fibonacijev niz pocinje sa 0 i 1 i svaki sledeci broj se dobija kao zbir prethodna dva:

- [0, 1, 1, 2, 3, 5, 8, 13, 21 ...

Gornji prikaz prikazuje prvih 9 brojeva fibonacijevog niza

#	Input	Output
1	6	0 1 1 2 3 5
2	10	0 1 1 2 3 5 8 13 21 34
3	47	0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986 102334155 165580141 267914296 433494437 701408733 1134903170 1836311903

Zadatak 6

Pronaci gresku u resenju narednog zadatka. Odgovor ispisati kao komentar u *praznoj* klasi *Zadatak6*.

Napisati funkciju koja proverava da li je cetvorocifren broj palindrom. Broj je palindrom ako je isiti i kada se cita sa levo na desno, i kada se cita sa desno na levo. Npr:

- 9229 - jeste palindrom
- 1231 - nije palindrom

Korisnik prosledjuje jednocifren, dvocifren, trocifren ili cetvorocifren broj. Ako broj nije cetvorocifren, dopuniti ga sa vodecim nulama do cetvorocifrenog. Npr za prosedjeni broj 132, posmatrati ga kao 0132.

```
public static boolean palindrom(int x){
    int prvaCifra = x / 1000;
    x = x % 1000;
    int drugaCifra = x / 100;
    x = x % 100;
    int trecaCifra = x / 10;
    int cetvrtaCifra = x % 10;
    if ((trecaCifra * cetvrtaCifra) / (prvaCifra * drugaCifra) == 1){
        return true;
    }
    return false;
}
```