|  |  |
| --- | --- |
| logo | OBJEKTNO ORIJENTISANO PROGRAMIRANJE**ISPIT** Januar, 2023. |

**PRAKTIČNI TEST**

1. Na programskom jeziku Java napisati program koji za 100 celih brojeva koji se unose sa komandne linije proverava da li su istovremeno deljivi sa 2 i 5. Za svaki broj koji zadovoljava ovaj uslov ispisati odgovarajuću poruku. Od brojeva koji zadovoljavaju uslov kreirati niz deljivo\_25. Posle učitavanja svih brojeva na komandnoj liniji formatirano štampati broj elemenata niza deljivo\_25 kao i sumu svih elemenata ovog niza. Napomena: za učitavanje koristiti Scanner klasu i metodu nextInt(). **[25]**
2. U Javi napisati program koji kreira natklasu Predmet sa dve promenljive šifra\_predmeta i broj\_bodova tipa int i promenljivu naziv\_predmeta tipa String i parametarskim konstruktorom koji vrši inicijalizaciju promenljivih. Klasa takođe definiše metodu Stampaj, koja na izlazu štampa rečenicu „Koeficijent nije definisan“ i vraća celobrojnu vrednost 0. Izvedite iz klase Predmet dve potklase ObavezniPredmet i IzborniPredmet. Svaka od ovih klasa treba da redefiniše metodu Stampaj, tako da vraća vrednost težinskog koeficijenta predmeta koji se za obavezni i izborni predmet se računa kao proizvod promenljivih šifra\_predmeta, broj\_bodova i brojeva 3 i 2, respektivno. U glavnom delu programa na osnovu klasa Predmet, ObavezniPredmet i IzborniPredmet kreirati tri objekta i za te objekte pozvati metodu Stampaj.  **[25]**
3. Kreirati klasu kosinus\_x kojom se izračunava vrednost funkcije cos(x) prema sledećem izrazu:

$$\cos(\left(x\right))=\sum\_{n=0}^{\infty }\left(-1\right)^{n}\frac{x^{2n}}{\left(2n\right)!} $$

 Odrediti vrednost funkcije cos(π/8) u 20 koraka (n=20) i formatirano štampati dobijenu vrednost funkcije u svakom koraku. Napomena: za izračunavanje faktorijela koristiti metodu faktorijel() iz paketa Metode.Matematika.  **[25]**

1. Napisati klasu ProveraParnosti koja sadrži javnu statičku metodu provera\_Parnosti koja vraća TRUE ako je broj koji je unet kao paran, a FALSE ako je neparan. Metoda baca proveravani izuzetak ako je uneti broj jednak nuli. Napisati i test klasu TestProveraParnosti koja poziva metodu klase ProveraParnosti. **[25]**

**TEORIJSKI TEST**

1. Redefinisane metode moguće su samo ako su i  metoda identični.
2. U procesu nasleđivanja, pored redefinisanja i preklapanja metoda mogu se:  **,**

 **.**

1. U procesu nasleđivanja potklasa **može** ili **ne može** postati natklasa drugoj potklasi. (Zaokrižiti tačan odgovor)
2. Rezervisana reč **super** ima dva pojavna oblika:

 ili .

1. Mogu se definisati više metoda sve dok im se deklaracija parametara razlikuju, u tom slučaju metode su .
2. Za pretraživanje znakovnih nizova koriste se metode  i .
3. Zamena podniza se obavlja metodom , a za umetanje znakovnog niza koristi se metoda.
4. U paketu java.applet su definisana tri interfejsa: ,i.
5. U klasi Applet definisane su sledeće četiri metode za upravljanje radom apleta: , , , \_\_\_\_\_\_\_\_\_\_\_.
6. Prozor Java aplikacije se formira u tri koraka:
7. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
9. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
10. Metode koje pružaju reference na pojedina okna su: ,,,.
11. Kod generičke metode se parametri tipa deklarišu **pre** ili **posle** tipa rezultata metode? (Zaokružiti tačan odgovor)
12. Da li konstruktori mogu biti generički ako klasa nije generička? **DA, NE**
13. Popunite tabelu:

|  |  |
| --- | --- |
| **Metode klase Thread** | **Značenje** |
| setName()/getName() |  |
| getPriority()/setPriority() |  |
| isAlive() |  |
| join() |  |
| run() |  |
| sleep() |  |
| start() |  |

1. Programska nit se može kreirati na dva načina pomoću objekta klase Thread:

 ili  **.**

1. Naredbom **throws** se svi tipovi izuzetka koje metoda može da baca izuzimajući
i  ali ih i .
2. Ako klasa želi da realizuje neki interfejs, u njenu deklaraciju treba uključiti naredbu .
3. Polimorfizam funkcioniše samo sa objektima .
4. Kod Jave su poznati sledeći specifikatori pristupa: , i .
5. Popuniti tabelu:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Specifiator pristupa** | **private** | **bez specifikatora** | **protected** | **public** |
| ista klasa |  |  |  |  |
| potklasa istog paketa |  |  |  |  |
| klasa istog paketa koje nisu podklase |  |  |  |  |
| potklase iz drugog paketa |  |  |  |  |
| klase iz drugih paketa koje nisu potklase |  |  |  |  |

Da – dozvoljava se pristup Ne – Ne dozvoljava se pristup

**Napomena:** Svako pitanje nosi 5 poena.

Predmetni profesor