Laborator 3. Metode statice. Metode supraincarcate. Metode recursive

1. Sa se scrie un program care memorează doua numere întregi citite din ferestre JOptionPane si contine metode care calculeaza si returnează suma, diferenta, produsul, maximul si minimul dintre numere. Sa se testeze execuţia metodelor apelându-le din metoda main a clasei si afişând valorile furnizate de metode.

2. Sa se implementeze urmatoarele metode:

- metoda "transformaInGradeCelsius" returneaza temperatura in grade Celsius echivalenta cu temperatura Fahrenheit folosind formula: C=5/9 * (F-32) ;
- metoda "transformaInGradeFahrenheit" returneaza temperatura in grade Fahrenheit echivalenta cu temperatura Celsius folosind formula: F=9/5 *C+32 ;

Folositi aceste metode pentru a scrie un program care primeste din linia de comanda sau dintr-o fereastra de tip JOptionPane doua valori reale ce reprezinta temperaturi Celsius, respectiv Fahrenheit si afiseaza temperatura echivalenta in celalalt sistem de temperaturi.

3. Scrieti o metoda numita afiseazaPatratStea care afiseaza un patrat umplut cu simbolul "*" a carui latura este specificata ca parametru al metodei. Incorporati aceasta metoda intr-un program care preia din linia de comanda o valoare intreaga pentru latura si realizeaza afisarea patratului prin apelarea metodei patratStea.

4. Modificati metoda "afiseazaPatratStea" din problema anterioara pentru a desena patrate formate cu orice caracter specificat in semnatura metodei.

5. Un numar intreg se numeste numar perfect daca divizorii sai (inclusiv 1), mai mici strict decat numarul respectiv, adunati dau numarul. De exemplu 6=1+2+3 este un numar perfect. Scrieti o metoda "perfect" care determina daca parametrul numar este perfect. Folositi aceasta metoda intr-un program care determina si afiseaza toate numerele perfecte intre 1 si 1000. Care sunt numerele perfecte intre 1 si 10 000 ?

6. Valoarea lui e^x poate fi calculată cu următoarea formulă:

$$e^x = \sum x^n/n! = 1 + x + x^2/2! + x^3/3! + \ldots, n=0,\infty$$

Să se scrie un program care calculeaza valoarea lui e^x cu ajutorul acestei formule. Cum seria este infinită, calculul se termină când termenul general este mai mic decât ε (a cărui valoare este preluată din linia de comandă): $| x^n/n! | < \varepsilon$. ε reprezintă precizia ce va aproxima valoarea e^x. Programul va conţine metode pentru calculul nerecursiv al factorialului şi al sumei.

7. Se consideră următorul program:

```
1. public class Test1{
2. public float oMetoda (float a, float b){return 0.0; }
3.
4. }
```

Care din următoarele metode sunt corecte dacă se adaugă (câte una) pe linia 3?

a) public int oMetoda (int a, int b){}
b) public float oMetoda (float a, float b){}
c) public float oMetoda (float a, float b, int c) throws Exception{}
d) public float oMetoda (float c, float d){}
e) public float oMetoda (int a, int b, int c){}

8. Să se scrie trei metode supraîncărcate care calculează volumul unei sfere, cub, respectiv al unui tetraedru regulat. Metodele conţin un parametru care reprezintă raza sferei, respectiv latura cubului/tetraedrului regulat. Acest parametru va fi preluat din linia de comandă sau dintr-o fereastra de tip JOptionPane.

   Observaţii. 1. Pentru constanta π se va folosi constanta `Math.PI`.

   2. $Vol(cub)=a^3$, $Vol(sfera)=4\pi R^3/3$; $vol(tetraedru)=a^3\sqrt{2}/12$

9. Se preiau din lina de comanda numerele n şi k (numere naturale n>k). Calculaţi recursiv C_n^k utilizând formula de recurenţă: $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$.

10. Sa se scrie un program care calculeaza si afiseaza c.m.m.d.c dintre 2 numere intregi preluate din linia de comanda, folosind urmatoarea definitie recursiva:

$$\mathbf{cmmdc(u,v)}=\begin{cases}\mathbf{cmmdc(v,\ u\ mod\ v),\ daca\ v \neq 0}\\ \mathbf{u\ daca\ v=0}\end{cases}$$

11. Să se scrie un program care calculeaza şi afişeaza a^b dupa formula:

12.$P(a,b) = \begin{cases} 1, daca\ b = 0, \\ a * P(a, b-1), altfel \end{cases}$

13.Fie definitia funcţiei lui Ackerman, $ac: \mathbb{N} x \mathbb{N} \rightarrow \mathbb{N}$

$$ac(m,n) = \begin{cases} n+1, daca\ m = 0 \\ ac(m-1, 1), daca\ n = 0 \\ ac\big(m-1, ac(m, n-1)\big), daca\ m, n > 0 \end{cases}$$

Sa se scrie un program care calculeaza si afiseaza ac(2,2).