Poo Laboratoare[1]

Contents

[1] http://www.google.ro/

Laborator1

Programare Orientata pe Obiecte: Laborator 1

1 Mediul integrat NetBeans

1.1 Scrierea și rularea programelor Java

Pentru scrierea și rularea unei aplicații Java utilizând acest IDE se vor efectua următorii pași:

- Se va alege din meniu **File->New Project...**.
- Se va selecta **Java** din panoul *Categories*, respectiv **Java Application** din panoul *Projects*.
- Se va introduce apoi (după apăsarea butonului **Next>**) numele proiectului (**Project Name**).
- La alegere, se poate schimba locația unde se va salva proiectul, precum și directorul de lucru (prin modificarea câmpurilor **Project Location**, respectiv **Project Folder**)
- Se va bifa căsuța **Set as Main Project** și se va debifa căsuța **Create Main Class**. În acest fel, clasa care va conține metoda *main()* va avea același nume cu cel al proiectului.
- După ce proiectul a fost creat, se vor adăuga fișiere sursă după preferințe. Acest lucru se face selectând cu click dreapta **Source Packages**, apoi **New->Java Class...**.
- După introducerea numelui și apăsarea butonului **Finish**, în IDE se va deschide fișierul sursă nou creat.
- Pentru compilare se va apăsa tasta **F11** sau se va apăsa din IDE butonul având simbol un ciocan.
- Pentru rulare se va apăsa tasta **F6** sau se va apăsa din IDE butonul având simbol un triunghi verde.

Observație! În NetBeans nu se pot compila și rula fișiere de sine stătătoare. Acestea trebuie să facă obligatoriu parte dintr-un proiect.

1.2 JDK Javadoc

Se va download-a documentația JDK de pe site-ul oficial Sun[2]. Apoi, în IDE efectuați următorii pași:

- Alegeți din meniul principal **Tools->Java Platforms**
- Selectați platforma la care vreți să adăugați documentația din panoul din stânga al ferestrei de dialog (în cazul de față **JDK 1.6(Default)**).
- În tab-ul **Javadoc** apăsați butonul **Add ZIP/Folder** și apoi specificați locația fișierelor Javadoc.
- Apăsați butonul **Close**. Restartați IDE-ul. În acest moment puteți accesa documentația din **Help->Javadoc References->Java Platform SE 6**.

Pentru a afla informații despre o clasă există două metode:

1. Se va accesa documentația din meniul **Help** și se va căuta numele clasei în pachetul care o conține sau
2. Se va apăsa combinația de taste **ALT-F1** când cursorul este desupra numelui clasei.

1.3 Parametrii în linia de comandă

Pentru a folosi parametrii în linia de comandă se vor efectua următorii pași:

- Se va executa un click dreapta pe numele proiectului și se va alege **Properties** din meniul pop-up.

[2] `http://www.sun.com`

- Se va selecta ramura **Run** din arborele de configurare al proiectului și în câmpul **Arguments** se vor scrie parametrii doriți.

2 Probleme de laborator

Observație: datele de intrare se vor da fie ca parametrii în linia de comanda fie ca valori fixe ale aplicațiilor!

2.1 Problema 1

Să se introducă programul de mai jos (clasa **Test**) într-un fișier cu numele **Prob1.java** folosind mediul integrat NetBeans. Clasele care *nu* au atributul *public* pot avea un nume diferit de numele fișierului sursă (în cazul compilatorului **javac** al firmei **Sun Microsystems**).

1. *(1 punct)* Să se compileze și să se execute acest program.

```
class Test {
     public static void main( String arg[] ) {
          System.out.println( ‘‘Test Java’’ );
     }
}
```

2. *(1 punct)* Adăugați atributul *public* înaintea cuvântului cheie *class*, recompilați și executați programul. Rezolvați problema apărută prin schimbarea numelui clasei.

2.2 Problema 2

(1 punct) Să se scrie un program cu o clasă care conține două funcții:

- funcția statică **print()** cu un argument de tip **String** (care afișează șirul primit ca argument)
- funcția statică **main()** care apelează funcția **print()** pentru afișarea unui șir constant.

2.3 Problema 3

1. *(1 punct)* Să se modifice programul anterior prin definirea a două clase, fiecare cu câte o singură funcție (statică):

- o clasă pentru funcția **main()**
- o clasă pentru funcția **print()**

Observație! O metodă statică a unei clase se apelează prin **NumeClasa.numeMetodaStatica(...)**.

Să se verifice dacă ambele clase din fișier pot fi publice.

2. *(1 punct)* Să se modifice programul anterior prin crearea a două fișiere sursă, fiecare conținând o clasă cu o singură funcție.

Se va executa clasa care conține funcția **main()**.

2.4 Problema 4

(1 punct) Să se scrie un program pentru afișarea tuturor argumentelor din linia de comandă.

Argumentele se pot transmite astfel:

- dacă programul se rulează din NetBeans se vor urma instrucțiunile de la începutul laboratorului.
- daca programul se rulează din linia de comandă:

```
java numeprogram.class arg1 arg2 arg3
```

2.5 Problema 5

(2 puncte) Să se scrie o funcție recursivă pentru calculul puterii întregi a unui număr întreg și un program pentru afișarea rezultatului funcției, alături de rezultatul funcției statice **Math.pow(baza, exp)**.

2.6 Problema 6

(2 puncte) Să se scrie un program format dintr-o singură clasă cu două funcții:

- o funcție de tip **boolean** care verifică dacă un număr întreg dat este prim (constantele de tip boolean sunt cuvintele cheie *true* și *false*).
- o funcție **main()** care verifică funcția anterioară pentru toate numerele naturale mai mici ca 20.

2.7 Problema 7 (bonus)

(1 punct) Să se scrie un program pentru verificarea ipotezei lui Goldbach pentru primele **n** numere pare, prin afișarea tuturor sumelor de două numere prime prin care poate fi exprimat un număr par. Variabila **n** poate fi inițializată cu o valoare constantă.

Pentru afișarea unei expresii de forma $a = b + c$ se va scrie:

```
System.out.println ( a + ‘‘ = ’’ + b + ‘‘ + ’’ + c);
```

unde a, b, c sunt variabile numerice de orice tip (short, int, long, float, double).

2.8 Problema 8 (bonus)

(1 punct) Să se scrie un program pentru ordonarea unui vector de numere și căutarea binară în acest vector folosind metodele statice **sort()** și **binarySearch()** din clasa **Arrays**. Vectorul este inițializat cu valori constante.

Se va modifica apoi pentru adăgare de numere generate aleator folosind metoda statica **random()** din clasa **Math**, cu rezultat de tip **double**.