POO – Laborator

Etapele dezvoltarii unui program Java

1. Conceperea si editarea programelor

Conceperea (proiectarea) programului inseamna **cel putin scrierea pe hartie a codului sau a unui pseudocod** (schita a programului scrisa in limbaj natural), **dar poate include si aplicarea unor instrumente** (diagrame, limbaje cum ar fi UML – Limbajul de Modelare Unificat) **si metodologii** (cum ar fi programarea agila/extrema sau ROSE – Ingineria Software Orientata-spre-obiecte Rational) mai complicate**.**

Urmeaza editarea codului Java. In Java tot codul este organizat in clase. Dupa editarea unei clase cu numele **<*NumeClasa*>** intr-un editor un editor de text, continutul va fi salvat intr-un fisier cu numele **<*NumeClasa*>.java**.

Atentie, limbajul Java este *case-sensitive* (face deosebirea intre litere mici si mari), **inclusiv in ceea ce priveste numele claselor si fisierelor**, iar numele fisierelor (urmate de extensia .java) trebuie sa fie identice cu numele claselor.

2. Compilarea programelor

Pentru obtinerea codului de octeti Java, trebuie compilat codul din acest fisier.

Daca se presupune utilizarea compilatorului Java (**javac**) din linia de comanda (consola standard de intrare), atunci trebuie executata urmatoarea comanda, in directorul *directorcurent*, in care se afla fisierul **<*NumeClasa*>.*java***:

```
directorcurent> javac <NumeClasa>.java
```

In urma acestei comenzi, compilatorul Java va crea genera codul de octeti (neutru din punct de vedere architectural, adica acelasi pentru orice pereche {sistem de operare, sistem de calcul} pe care e compilat) corespunzator intr-un fisier cu numele **<*NumeClasa*>.class**, in directorul curent (acelasi director in care se afla si fisierul <*NumeClasa*>.java).

Compilatorul Java genereaza cate un fisier pentru fiecare clasa compilata.

3. Lansarea programelor

Pentru executia programului, acesta trebuie lansat in interpretor (**java**), folosind comanda:

```
directorcurent> java <NumeClasa>
```

(numele clasei Java este argument pentru programul interpretor Java, numit **java**).

4. Depanarea programelor

Daca in urma compilarii apar erori, ele trebuie corectate (urmarind si **indicatiile din mesajele de eroare**), revenind la etapa conceperii si editarii. **O alternativa** este folosirea **utilitarului de depanare a programelor Java**.

Daca in urma executiei apar erori de conceptie (comportamentul programului difera de cel dorit), ele trebuie corectate revenind la etapa conceperii si editarii.

5. Generarea automata a documentatiei

O facilitate suplimentara, importanta, oferita de kitul de dezvoltare Java este generatorul de documentatie Java. Delimitatorii **`/**`** si **`*/`** sunt folositi pentru a arata ca textul trebuie tratat ca un comentariu de catre compilator, dar de asemenea ca textul este parte din documentatia clasei care poate fi generata folosind utilitarul **javadoc**.

Pentru generarea documentatiei unui program, se foloseste comanda:

```
directorcurent> javadoc <NumeClasa>.java
```

De exemplu, daca generam documentatia pentru programul Salut:

```
1   /** Clasa care ilustreaza elementele esentiale ale unui program Java.
2    *  Trebuie sa fie publica pentru ca are metoda principala.
3    *  @author  Eduard C. Popovici
4    */
5   public class Salut {
6     /** Metoda principala (punct de intrare in program).
7      *  Este prima metoda executata de JVM (Java Virtual Machine).
8      *  Primeste ca parametri argumentele din lina de comanda.
9      *  Nu returneaza nici o valoare. Trebuie sa fie 'public static'
10     */
11    public static void main(String[] args) {
12      System.out.println("Buna ziua"); // Afisarea unui text pe ecran
13    }
14  }
```

cu comanda **javadoc:**

```
directorcurent>javadoc  Salut.java
Loading source file Salut.java...
Constructing Javadoc information...
Standard Doclet version 1.5.0_07
Building tree for all the packages and classes...
Generating Salut.html...
Generating package-frame.html...
Generating package-summary.html...
Generating package-tree.html...
Generating constant-values.html...
Building index for all the packages and classes...
Generating overview-tree.html...
Generating index-all.html...
Generating deprecated-list.html...
Building index for all classes...
Generating allclasses-frame.html...
Generating allclasses-noframe.html...
Generating index.html...
Generating help-doc.html...
Generating stylesheet.css...
```

sunt create paginile Web (accesibile aici):