

PROIECTAREA SI IMPLEMENTAREA ALGORITMILOR – LABORATOR 1

Platforma Java

Pagina Web de la care puteți porni căutarea documentației originale oferite de firma Sun Microsystems pentru utilizatorii limbajului Java www.java.com.

Se va lucra cu NetBeans: (download JDK 8 & NetBeans)

- 1) <http://www.oracle.com/technetwork/java/javase/downloads/index.html#close>
- 2) Selectati NetBeans with JDK8
- 3) Bifați Accept License Agreement și alegeți versiunea de NetBeans corespunzătoare SO dumneavoastră

Mediul integrat NetBeans

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

1. Se va alege din meniul File opțiunea New Project.
2. Se va selecta Java din panoul Categories, respectiv Java Application din panoul Projects.
3. După apăsarea butonului Next, se va introduce numele proiectului - Project Name.
4. La alegere, se poate schimba locația unde se va salva proiectul, precum și directorul de lucru (prin modificarea câmpurilor Project Location, respectiv Project Folder).
5. Se va bifa căsuța Set as Main Project și se va debifa casuța Create Main Class. In acest fel, clasa care va conține metoda main() va avea același nume cu cel al proiectului.
6. După ce proiectul a fost creat, se vor adăuga fișiere sursă după preferințe. Acest lucru se face selectând, cu click dreapta, Source Packages, apoi din meniul New se va alege Java Class.
7. După introducerea numelui și apăsarea butonului Finish, în IDE se va deschide fișierul sursă nou creat.
8. Pentru compilare, se va apăsa tasta F1 sau se va selecta, din IDE, butonul având ca simbol un ciocan.
9. Pentru rulare, se va apăsa tasta F6 sau se va selecta, din IDE, butonul având simbol un triunghi verde.

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

Observații

1. Echivalentul bibliotecilor din C++, în Java, sunt pachetele. Acestea conțin clase ce pot fi utilizate fără a le defini, dacă se folosește comanda: import nume_pachet; (exemplu: import java.util.Scanner;), la începutul programului.

2. Majoritatea instrucțiunilor și tipurilor de date elementare din C++ se folosesc și în Java.
3. Ordinea de scriere a metodelor și datelor nu are importanță.

Exemplu

```
class exemplu{  
  
    public static void main(String[] args){  
  
        System.out.println("exemplu java");  
  
    }  
  
}
```

Programul anterior conține o singură clasă (evident cea principală), numele fișierului ce-l conține trebuie să fie exemplu.java. Efectul acestui program este afișarea textului exemplu java în consola input/output.

Observații

1. Operatorii sunt la fel ca în C++.
2. Instrucțiunile liniare, alternative și repetitive din C++ sunt cu aceeași sintaxă și în Java.
3. Mulțimile de valori pentru aceste tipuri de date și operatorii sunt la fel ca în C++.

Citirea datelor

Citirea datelor în Java poate fi realizată folosind clasa **Scanner** definită în pachetul **java.util.Scanner**. Etapele ce trebuie parcurse pentru citirea unor date sunt următoarele:

- 1) Importarea unui pachet:

Linia

```
import java.util.Scanner;
```

va fi scrisă la începutul programului.

- 2) Definirea unui obiect:

```
Scanner cin=new Scanner(System.in);
```

înainte de citirea datelor

- 3) Folosirea metodelor de citire a datelor:

cin.nextInt(); returnează un număr de tip int citit de la consolă

cin.nextDouble(); returnează un număr de tip double citit de la consolă

cin.nextFloat(); returnează un număr de tip float citit de la consolă

cin.nextLine(); returnează un șir de caractere citit de la consolă

Aplicatie rezolvata cu instructiuni liniare si alternative

Se dă n număr natural cu cel mult 3 cifre. Afișați pătratul și cubul lui n.

Soluție:

```
import java.util.Scanner;

public class P1 {

    public static void main(String[] args) {

        int n;

        Scanner cin=new Scanner(System.in);

        System.out.print("n=");

        n=cin.nextInt();

        System.out.println("patratul lui "+n+" este "+n*n);

        System.out.println("cubul lui "+n+" este "+n*n*n); } }
```

Programe Java cu instructiuni repetitive: *Se dă n număr natural nenul cu maxim 9 cifre. Afișați suma cifrelor lui n.*

Soluție:

```
import java.util.Scanner;

public class cifre { public static void main(String[] args) {

    int n,S;

    Scanner cin=new Scanner(System.in);
```

```

System.out.print("n=");

n=cin.nextInt();

S=0;

while(n>0){

S=S+n%10;

n=n/10; }

System.out.println(S); } }

```

Aplicatii propuse:

1. Se dau de la tastatura 2 cifre nenule a si b. Sa se afiseze pe ecran cele 2 numere care se pot forma cu acestea, in ordine crescatoare.

Ex: a=5 si b=2 Se vor afisa cele 2 numere formate in aceasta ordine: 25 52

Ex2: a=0, b=3 Se vor afisa valorile: 3 si 30

2. Se citește de la tastatură un număr natural n cu exact 4 cifre. Să se verifice dacă numărul este:

Echilibrat - spunem că un număr este echilibrat dacă numărul de cifre pare este egal cu numărul de cifre impare.

Pitic - spunem că un număr este pitic dacă toate cifrele sunt mai mici decât 4.

Generos - spunem că un număr este generos dacă suma cifrelor lui e mai mare decât $n+2$.

Corect - spunem că un număr este corect dacă suma și produsul cifrelor sunt pătrate perfecte.

3. Se citește de la tastatură o valoare naturală $x(x>10)$. Dacă valoarea citită are cifra unităților egală cu cifra zecilor se vor afișa primele două numere naturale mai mari decât valoarea x. În caz contrar se va afișa cifra maximă dintre cifra zecilor și a unităților. Exemplu: Pentru $x=3455$ se va afișa 3456 3457 Pentru $x=3485$ se va afișa 8.

Referinte:

BAZELE PROGRAMARII - JAVA DUPA C++

Autor: Doru Anastasiu Popescu