

Laborator 1 - Programare Avansata pe Obiecte

Vlad Popescu

vladpopescu193@gmail.com

vlad.popescu@endava.com

Setup:

1) Install JDK (Opțional pentru primul laborator, BlueJ vine la pachet cu JDK) Verifica Instalare-Java.docx pentru pași.

2) Install Git

- Download and install Git from:
<https://git-scm.com/download/win>

3) Create GitHub account

- Create from here <https://github.com/>
- Create a repository named **pao-labs**
- Clone repository locally
- At the end of the lab push work to github

4) Install BlueJ IDE

- Install from <https://www.bluej.org/versions.html>

Java:

- Este un limbaj de programare orientat pe obiect
- Este un limbaj case sensitive
- Programele Java sunt interpretate => este utilizata o masina virtuala Java (JVM) care interpretează un cod compilat al programelor
`.java` (fisier cod sursa) ---> `javac` (compilator) ---> `.class` (bytecode) ---> `java` (interpretor, emulator de cod bytecode = JVM) ---> `OS` (sistem de operare)
- Este un limbaj portabil => este independent de platforma pe care ruleaza
- Are funcția de garbage collector, care se ocupă de eliberarea memoriei ocupate de obiecte care nu mai sunt folosite

Cuvinte rezervate in Java:

abstract	double	int	strictfp
boolean	else	interface	super
break	extends	long	switch
byte	final	native	synchronized
case	finally	new	this
catch	float	package	throw
char	for	private	throws
class	goto*	protected	transient
const*	if	public	try
continue	implements	return	void
default	import	short	volatile
do	instanceof	static	while

- Java are un set de cuvinte rezervate, care nu pot fi folosite ca nume de clase, interfețe, variabile sau metode
- Excepție fac true, false, null care nu sunt cuvinte cheie, dar nu pot fi nici ele folosite ca nume în aplicații

Metoda Main:

- Un program scris în limbajul Java este alcătuit dintr-un număr de clase.
- Una dintre acestea trebuie să conțină o metodă numită main. Clasa care conține metoda main se numește clasa principală. Evident, dacă un program este format dintr-o singură clasă, atunci aceasta trebuie să conțină neapărat main.

```
package com.company;  
  
public class Main {  
  
    public static void main(String[] args) {  
        // write your code here  
    }  
}
```

Conventii nume clase:

- Nu pot fi folosite cuvinte cheie
- Numele nu poate începe cu o cifră, dar poate să conțină cifre
- Nu poate conține spații sau operatori (sau caractere speciale: #)

Primitive și Clase Wrapper:

- În Java există 2 categorii importante de tipuri de date: **referințe** și **primitive**.
- Valorile numerice, caracterele și valorile logice(true sau false) sunt utilizate ca primitive, deoarece sunt mai eficiente, ca viteza de procesare, sintaxa și efecte asupra memoriei.

- Exista scenarii când este necesar sa stochezi valori primitive în interiorul unor obiecte. Pentru aceasta nevoie, Java oferă un set de clase folosite la ambalarea (wrap) valorilor primitive într-un obiect.
- Pe lângă rolul principal (gestionarea valorilor primitive), clasele wrapper contin un set de metode utilitare utilizate pentru a converti valori la tipuri diferite sau de a transforma şiruri de caractere în valori numerice.

Tip primitiv	Clasa wrapper
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Operatori de comparare:

- Operatorii sunt utilizaţi pentru a exprima diferite formule elementare de calcul sau prelucrare a informaţiei
- Operatorii se pot clasifica în următoarele categorii:
 - Operatori aritmetici(+, -, *, /, %, ++, --)
 - Operatori relaţionali (<, <=, ==, >=, >, !=)
 - Operatori booleani (&&, ||, !, &, |, ^)
 - Operatori la nivel de bit (&, |, ^, ~, <<, >>, >>>)
 - Operatorul de concatenare a String-urilor (+)
 - Operatori speciali (instanceof, ?:)
 - Operatori de atribuire (=, +=, -=, *=)

Pachete:

- Un pachet este o colecție de clase și interfețe
- Scopul pachetelor in java :
 - Organizarea claselor și interfețelor
 - Evitarea conflictelor de nume
 - Controlul accesului

- Exemple de pachete:

java.lang - clasele de baza ale limbajului Java (acest pachet este importat by default)

java.io - intrari/iesiri, lucrul cu fisiere

java.util - clase și interfețe utile

java.sql - lucrul cu baze de date

java.math - operații matematice cu numere mari

javax.swing - interfata grafica cu utilizatorul

Instrucțiunile limbajului Java:

- Instrucțiuni decizionale
- Instrucțiuni repetitive
- Instrucțiuni de salt

Exercitii:

1. Scrieți un program care sa afișeze toate numerele pare din intervalul $[0, n]$, unde n este un număr citit de la tastatura.
2. Scrieți un program care sa compare doua numere a și b citite de la tastatura si sa afișeze numărul mai mare.

3. Fiind dat un număr n citit de la tastatura, sa se afiseze toți divizorii sai.
4. Scrieți o metoda care sa calculeze factorialul unui numar n citit de la tastatura
5. Scrieți un program care sa intoarca *true* dacă un număr este prim, și *false* în caz contrar.
6. Se citesc două numere a și b de la tastatura, sa se afișeze valoarea a^b .
7. Sa se afiseze al n -lea termen din seria Fibonacci, unde n este citit de la tastatura