Curs 1 POO
© Conf.dr. Crenguta M. Puchianu

Introducere

- Limbajul de programare Java
- Proprietatile limbajului Java
- Scrierea, compilarea si executarea programelor
- Variabile

2022

Limbajul de programare Java

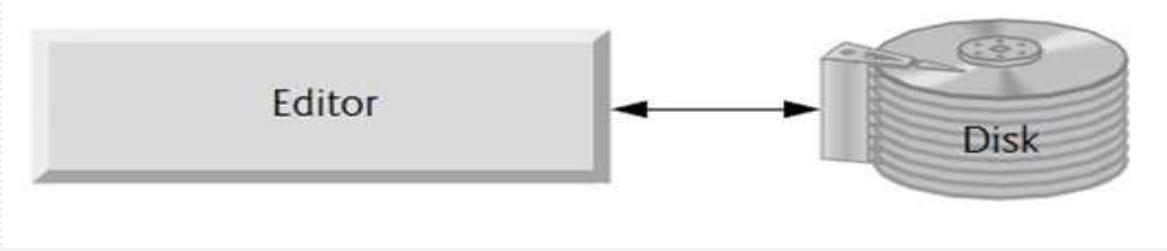
- Java a fost dezvoltat de Sun Microsystems in cadrul unui proiect inceput in 1991, condus de James Gosling.
- Un scop principal al lui Java este de a scrie programe ce pot fi executate pe calculatoare/telefoane mobile ce au diferite sisteme de operare. Aceasta calitate se numeste portabilitate software ("write once, run anywhere ").
- Sun Microsystems a fost cumparat de Oracle in 2009.
- Putem scrie in programele noastre fiecare clasa sau putem importa clase predefinite in libraria Java numita Java APIs (Application Programming Interfaces)

Proprietati ale limbajului Java

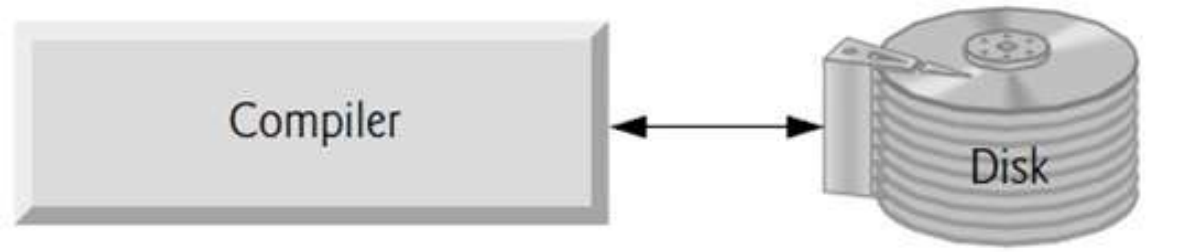
- Un program Java este format din module numite clase.
- In timpul executiei, un program este format dintr-o multime de obiecte care comunica prin mesaje
- Obiect = structura (date) + comportament (metode)
- Clasa = multime de obiecte
- Clasele sunt organizate in structuri arborescente pe baza relatiei UML (Unified Modeling Language) de generalizare/specializare
- programele Java pot fi distribuite în reţea şi deci adaptate folosirii lor în internet şi în intranet
- Programele Java sunt interpretate. Este utilizată o Maşină Virtuală Java (JVM) care interpretează un cod compilat al programelor
- Este un limbaj concurent pentru ca include facilităţi ce permit crearea şi gestiunea de fire de execuţie concurente.
- Limbaj Bytecode

Programe Java

- Procesul de a scrie si de a executa un program Java:
- **Pas 1**. Scriem programul. Acest pas consta din editarea unuia sau mai multor fisiere cu un editor text. Fiecare fisier este memorat pe disc cu extensia .java.



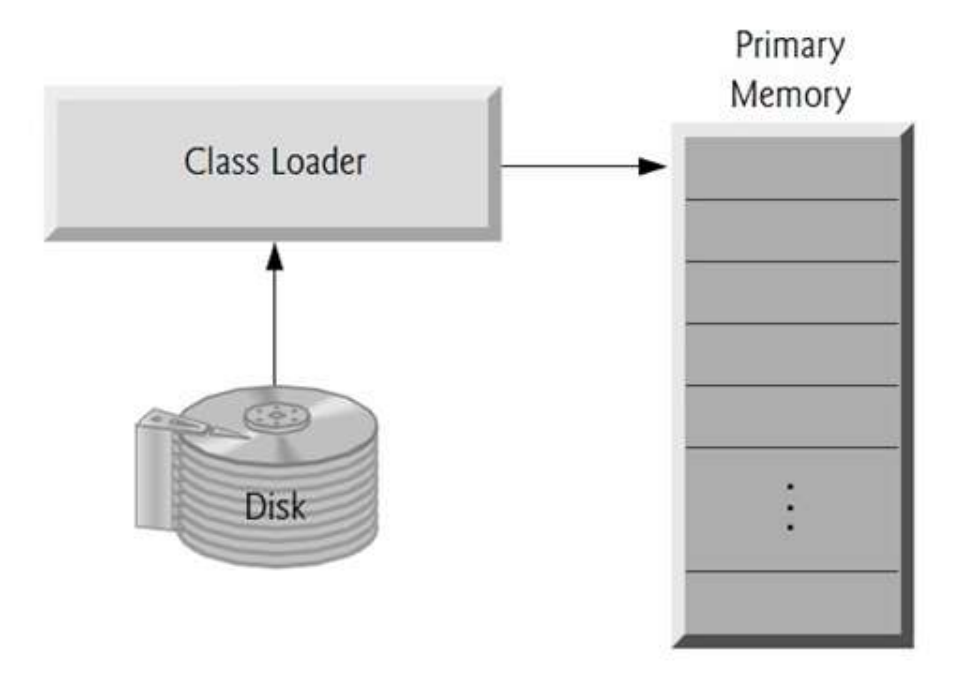
- **Pas 2**. Programul Java este compilat in bytecode. Compilatorul produce fisiere cu extensia .class .



Programe Java (cont.)

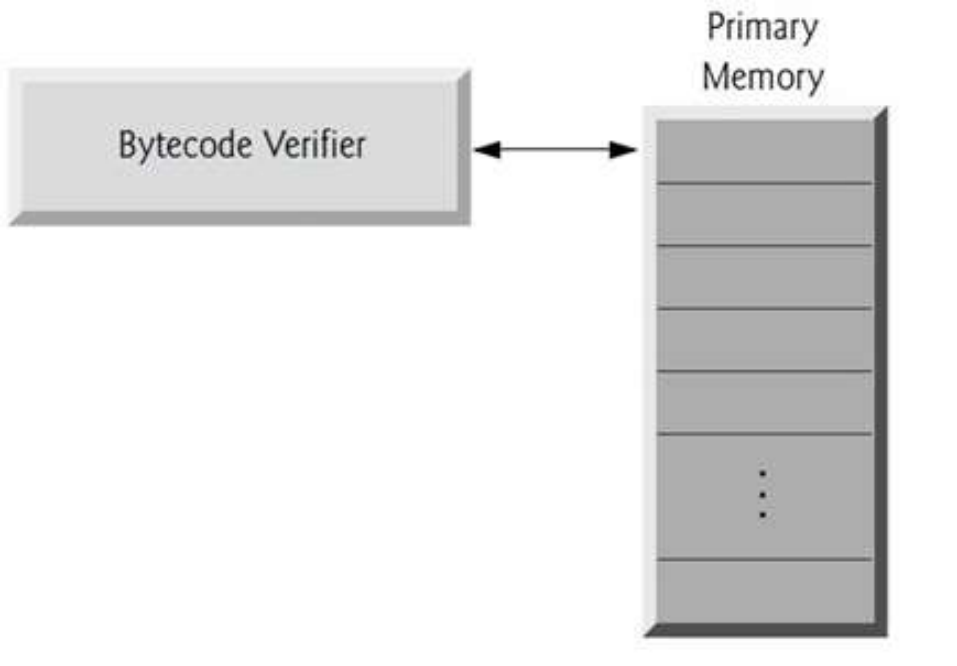
- **Pas 3**. Incarcarea programului in memorie. In aceasta etapa, Java Virtual Machine (JVM) incarca programul in memorie sa ruleze. Incarcatorul de clase al MVJava ia fisierele .class care contin bytecode si le transfera in RAM.

Fisierele .class pot fi incarcate din calculatorul local sau de pe retea (locala sau Internet).



Programe Java (cont.)

- **Pas 4**. Verificare bytecode. In aceasta etapa, clasele sunt incarcate si verificatorul de bytecode examineaza bytecode-urile lor, pentru a se asigura ca sunt valide si nu incalca restrictiile de securitate ale limbajului Java.



Programe Java (cont.)

- **Pas 5**. Executia programului.
- In aceasta faza, JVM executa bytecode-urile programului, realizand astfel actiunile specificate de program.
- In versiunile initiale ale Java, JVM era un interpretor simplu pentru bytecodes.
- Acest lucru a facut ca majoritatea programelor Java sa ruleze incet deoarece JVM interpreta si executa cate un bytecode in fiecare moment de timp.
- Cu arhitecturile de calculator curente, calculatoarele pot executa instructiuni in paralel.
- MVJava ruleaza la ora actuala bytecode folosind o versiune a interpretarii numita just-in-time compilation (JIT) .

Programe Java (cont.)

- Un prim exemplu de program Java:

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

- Fisierul .java trebuie sa aiba acelasi nume cu clasa publica, Test.java.
- Orice program contine cel putin o clasa ce trebuie sa fie principala. Cuvantul cheie *class* incepe declararea clasei si este urmata imediat de numele clasei (Test).
- Cuvintele cheie sunt rezervate pentru Java si trebuie scrise numai cu litere mici.

Programe Java (cont.)

- In mod normal, definirea unei clase contine una sau mai multe metode. Pentru un program Java, o unica clasa contine metoda *main* ce trebuie sa definite ca in exemplul de mai sus; altfel, Masina Virtuala Java (MVJ) nu poate rula aplicatia.

- Metodele realizeaza calcule ce pot return o valoare la sfarsitul calculului. Cuvantul cheie void indica faptul ca metoda nu returneaza nimic.

- Parametrul *String[ ] args* specificat in parenteze este obligatoriu cand declaram metoda main. Acest parametru este o variabila tablou ce contine parametrii liniei de comanda, daca ei au fost enumerati in linia de comanda.
- Paranteza { spune ca incepe corpul metodei si paranteza } inchide corpul metodei.

- Linia 3 din programul Java anterior, reprezinta apelarea metodei println care realizeaza actiunea de afisare a sirului delimitat de ghilimele. Variabila *System.out* refera un obiect al clasei PrintStream ce contine un flux de iesire pe bytes transmis din memoria pe ecranul monitorului (i.e. iesirea standard). Aceasta metoda permite aplicatiilor Java sa afiseze informatii in fereastra de iesire.

Programe Java (cont)

- Metoda *System.out.println* afiseaza o linie de text in fereastra de afisare. Sirul din paranteze () este parametrul actual al metodei. Dupa ce este afisat sirul “Buna ziua“ se plaseaza cursorul la inceputul liniei urmatoare.

- O metoda similara este *print* care lasa cursorului la sfasitul textului afisat.

- Metoda *System.out.printf* afiseaza date formatate. Instructiunea

```
System.out.printf( "%s\n%s\n", "Welcome to ", "Java Programming!" );
```

Contine apelarea acestei metode pentru afisarea sirurilor "Welcome to " si "Java Programming "

Compilarea si executarea programelor Java in linia de comanda

- Versiunile < JDK 11

Pas 1. Deschiderea liniei de comanda

cmd> Command prompt > Click sau Enter

Pas 2. Schimbarea folder-ului curent cu cel care contine fisierul sursa Test.java

cd C:\ProgrameJava si Enter

Pas 3. Copierea caii folder-ului unde se afla instalat JDK:

C:\Program Files\Java\jdk-18.0.2.1\bin

Pas 4. Executarea compilatorului Java in linia de comanda:

```
C:\ProgrameJava>"C:\Program Files\Java\jdk-18.0.2.1\bin"\javac Test.java
```

Compilarea si executarea programelor Java in linia de comanda (cont)

Pas 5. Daca nu sunt erori de compilare, se executa in linia de comanda interpretorul Java:

```
C:\ProgrameJava>"C:\Program Files\Java\jdk-18.0.2.1\bin"\java Test
```

- Versiunile >= JDK 11

Pas 1. Deschiderea liniei de comanda

cmd> Command prompt > Click sau Enter

Pas 2. Schimbarea folder-ului curent cu cel care contine fisierul sursa Test.java

cd C:\ProgrameJava si Enter

Compilarea si executarea programelor Java in linia de comanda (cont)

Pas 3. Executarea fisierelor sursa fara compilarea lor:

```
C:\ProgrameJava>java Test.java
```

Variabile

- O **variabilă** este o componentă a programului care identifică un spaţiu în memoria calculatorului alocat la runtime. În spaţiul respectiv programul poate memora date necesare calculului.

- În textul programului, variabilele apar de obicei ca nişte identificatori. Prin **identificator** într-un limbaj de programare se înţelege un nume simbolic dat unei componente a programului: variabilă, clasă, metodă, etc.

- Regula: Un identificator Java este o secventa de litere, cifre, $ si _ ce incepe cu o litera, $ sau _.

- Notam ca Java este "case sensitive", insemnand ca compilatorul face diferenta intre litere mari si mici.

- **Conventii de nume**
- Clase: Persoana, SistemDeCalcul
- Metode: verificaNume, cautaPersoana
- Constante: CONSTANTA, MAXIM, PI
- Interfete: IServicii, Taxe
- Variabile: nume, prenume, forma, culoare, oVariabila

Cuvinte cheie

Anumiţi identificatori, numiţi *cuvinte-cheie* (key words), au semnificaţii precise în cadrul programului şi sunt rezervaţi doar scopului de a reprezenta aceste semnificaţii. Ei nu pot fi utilizaţi pentru a desemna variabile, metode, clase sau alte componente introduse de programator, sunt "cuvinte rezervate".

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