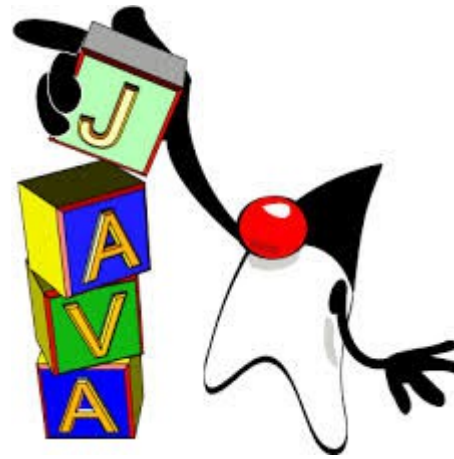




# Tehnici avansate de programare

# Desfășurarea cursului

- Scopul
- Motivația
- Modul de lucru
- Platforma de programare
- Documentația
- Evaluarea
- Laborator: probleme, proiecte, referate → ușor
- Examen: test scris → greu



# Ce este “Java” ?

- Limbaj de programare
- Platformă de lucru
- 1995
- Sun Microsystems / Oracle (2010)
- James Gosling
- Duke



# De ce Java?



# Limbaajul de programare Java

- Simplitate
- Ușurință în crearea de aplicații complexe
- Robustețe: ~~pointeri~~, administrarea automată a memoriei, GC
- Complet orientat pe obiecte
- Securitate
- **Neutralitate arhitecturală**
- **Portabilitate**
- Performanță

# Platforme de lucru Java

- **Java SE (Standard Edition)**

Aplicații desktop independente, appleturi, Java Web Start

- **Java ME (Micro Edition)**

Programarea dispozitivelor mobile

- **Java EE (Enterprise Edition)**

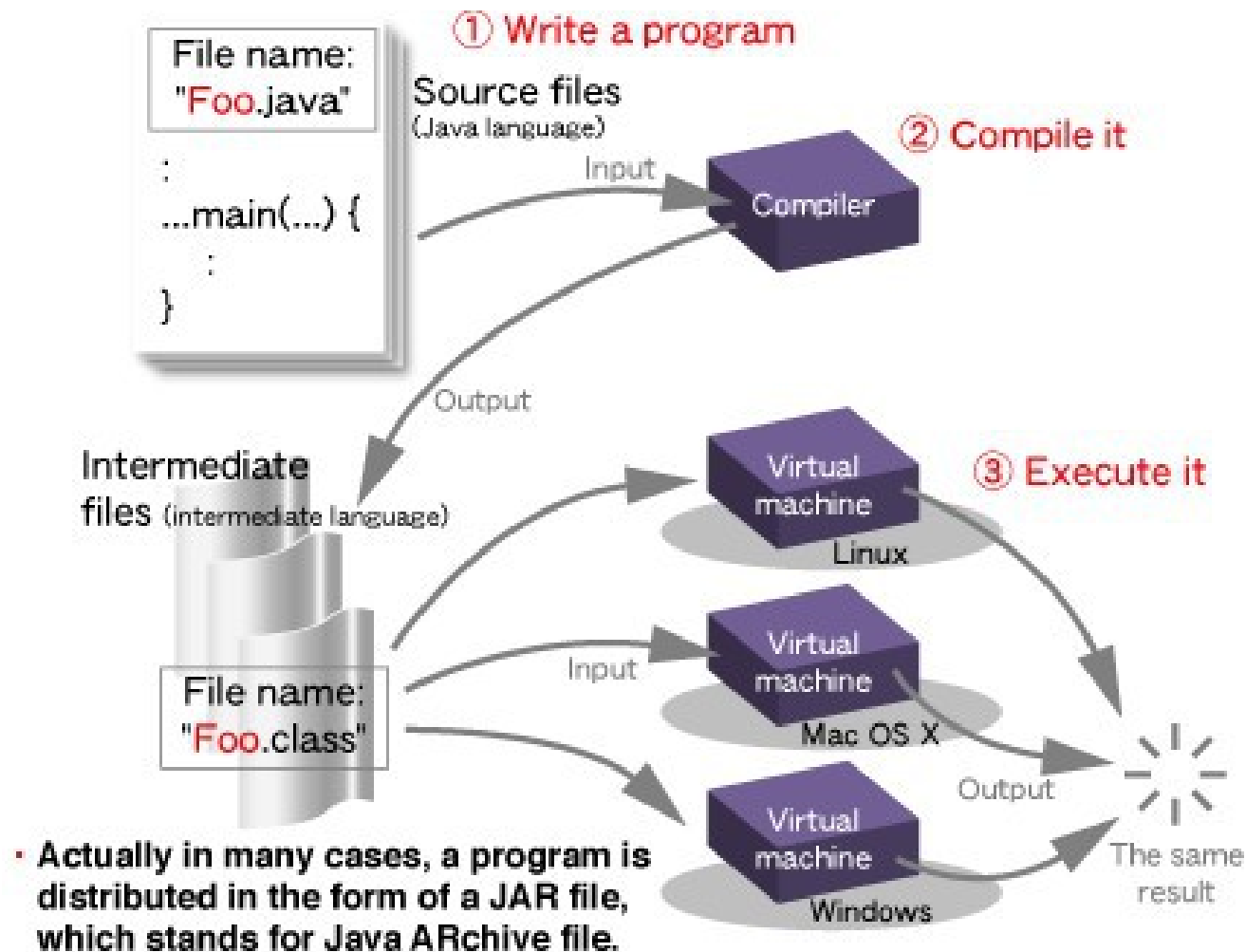
Aplicații complexe, pe mai multe niveluri pentru sisteme eterogene, aplicații și servicii Web, etc.

- **Java Card, Java FX (RIA)**

# Compilat și interpretat

- **Limbaje interpretate**
  - simplitate, portabilitate
  - viteza de execuție redusă
- **Limbaje compilate**
  - viteză de execuție sporită
  - lipsa portabilității
- **Java**  
**compilat + interpretat**

# Java Virtual Machine (JVM)



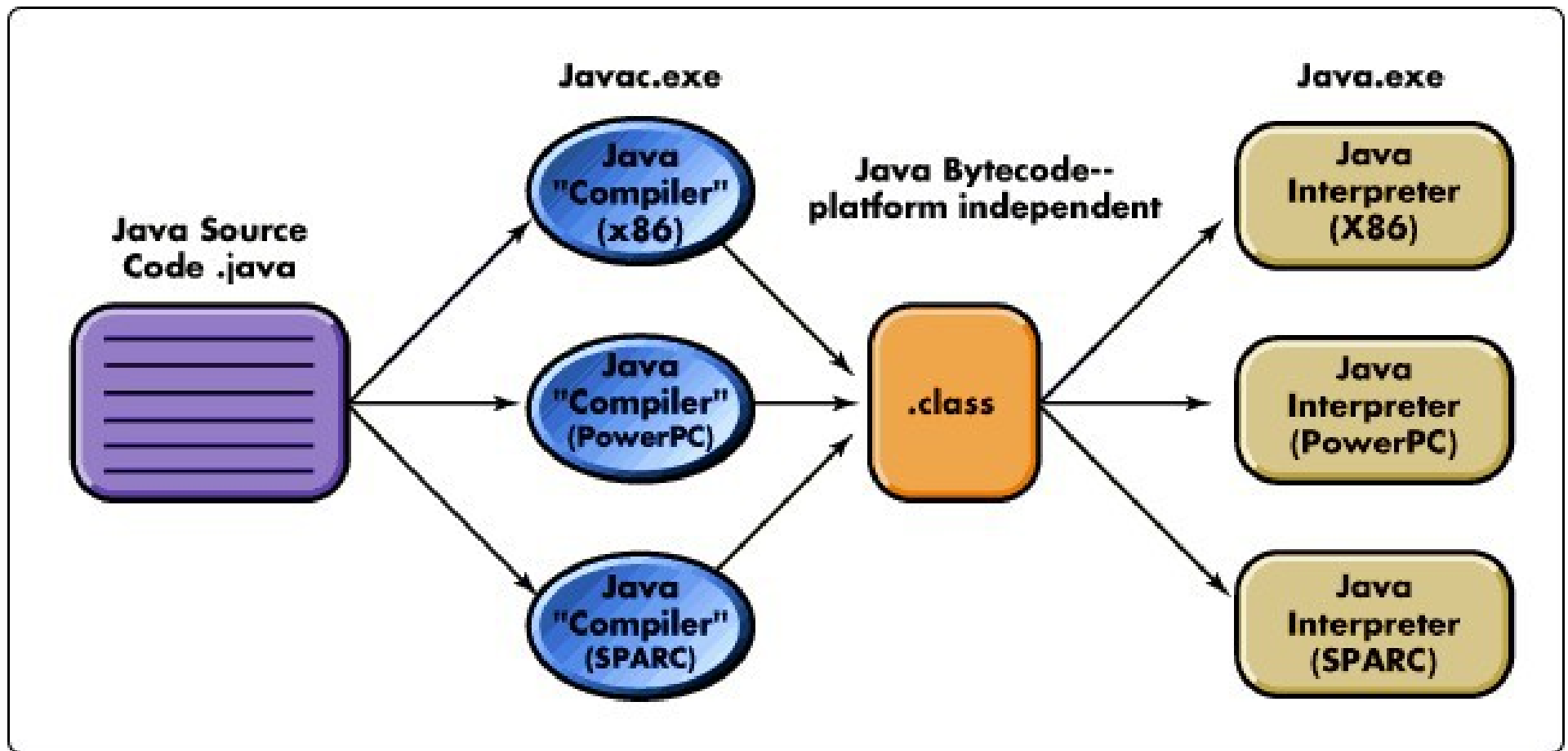


# Primul program

```
public class HelloWorld {  
  
    public static void main(String args[]) {  
        System.out.println("Hello world!");  
    }  
}
```

- *Sursa:* HelloWorld.java
- *Compilare*  
javac HelloWorld.java → HelloWorld.class
- *Rulare*  
java HelloWorld

# java, javac



# javap

## javap -c HelloWorld

Compiled from "FirstApp.java"

```
class FirstApp extends java.lang.Object{  
    FirstApp();
```

Code:

```
    0: aload_0  
    1: invokespecial #1;  
        //Method java/lang/Object."<init>":()V  
    4: return
```

```
public static void main(java.lang.String[]);
```

Code:

```
    0: getstatic #2;  
        //Field java/lang/System.out:Ljava/io/PrintStream;  
    3: ldc #3;  
        //String Hello world!  
    5: invokevirtual #4;  
        //Method java/io/PrintStream.println:(Ljava/lang/String;)V  
    8: return
```

```
}
```

## Obfuscate

# UNICODE

- Inlocuiește setul de caractere ASCII
- Un caracter se reprezintă pe **2 octeți**
- **65536** semne, `\uxxxx`
- Compatibil ASCII: primele 256 caractere sunt cele din ASCII
- Structurat în **blocuri**: Basic Latin, Greek, Arabic, Gothic, Currency, Mathematical, Arrows, Musical, etc.
- `public class приветмир { }`
- `System.out.println(" 好世界 ");`

# Sintaxa

- Similară cu C++
- Cuvinte cheie
- **Literali:** "Hello World", 'J', 'a', 'v', 'a', 10, 010, 0xA, ob11, 12.3, 12.3d, 12.3f, 12e3, 123L, true, false, null
- **Separatori:** ( ) { } [ ] ; , .
- **Operatori**

(char)65 + "na" + "are" + (8 >> 2) + " mere"

<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>

# Comentarii

```
/* To change this template file, choose Tools | Templates
```

```
and open the template in the editor. */
```

```
/**
```

```
 * Clasa principala a aplicatiei
```

```
 * @author Duke
```

```
 */
```

```
public class HelloWorld {
```

```
    /**
```

```
     Metoda de unde porneste executia aplicatiei
```

```
     @param args the command line arguments
```

```
     */
```

```
    public static void main(String args[]) {
```

```
        // TODO code application logic here
```

```
        System.out.println("Hello World!"); // Gata!
```

```
    }
```

```
}
```

**javadoc** - generatorul automat de documentație

# Tipuri de date

## Tipuri primitive

- aritmetice: byte (1), short (2), int (4), long (8)
- reale: float (4), double (8)
- caracter: char (2)
- logic: boolean (true, false)

**Tipuri referință:** clase, interfețe, adnotări, enum  
~~pointer, struct, union~~

# Variabile

## Declarare [+ Inițializare]

```
byte a;
```

```
int valoare = 100;
```

```
final double PI = 3.14;
```

```
long numarElemente = 12345678L;
```

```
String bauturaMeaPreferata = "apa";
```

Java naming conventions



# Variabile (cont.)

```
class Exemplu {  
    int a; //Variabile membre  
    public void metoda(int b) { //Argumente ale metodelor  
        a = b; //Variabile locale unei metode  
        int c = 10;  
        for(int d=0; d < 10; d++) { //Variabile locale unui bloc de cod  
            c --;  
        }  
        try {  
            a = b/c;  
        } catch(ArithmeticException e) { System.err.println(e.getMessage()); }  
        //Tratarea exceptiilor  
    }  
}
```

# Controlul execuției

- Instrucțiuni de decizie  
if-else, switch-case
- Instrucțiuni de salt  
for, while, do-while
- Instrucțiuni pentru tratarea excepțiilor  
try-catch-finally, throw
- Alte instrucțiuni  
break, continue, return

# Tablouri de elemente

- Declarare

```
int[] a; byte b[];
```

- Instantiere

```
a = new int[10]; char c = new char[100];
```

- Inițializare

```
String culori[] = {"Rosu", "Galben"};
```

```
metoda( new String[]{ "Rosu", "Galben" } );
```

- Dimensiunea unui tablou

```
a.length și nu a.length()
```

# Tablouri multidimensionale

- Tablouri de tablouri

```
int[][] m2d = new int[10][20];
```

```
int[][][] m3d = new int[10][20][30];
```

- Copierea tablourilor

```
System.arraycopy
```

```
int a[]; int b[]; ... ce efect are a = b?;
```

- Metode utile pentru tablouri

```
java.util.Arrays
```

```
- binarySearch, equals, fill
```

# Șiruri de caractere

- char[]

```
char data[] = {'a', 'b', 'c'};
```

- String **Immutable Object**

```
String s = "abc"; String s = "a" + "b" + "c";
```

```
String s = new String("abc");
```

```
String s = new String(data);
```

- StringBuilder, StringBuffer

```
StringBuilder sb = new StringBuilder("a");
```

```
sb.append("b").append("c");
```

# Testarea egalității

- Tablouri

```
int a[] = {1, 2};
```

```
int b[] = {1, 2};
```

```
a == b / a.equals(b) / Arrays.equals(a,b)
```

- Șiruri de caractere

```
String s1 = new String("abc");
```

```
String s2 = new String("abc");
```

```
s1 == s2 / s1.equals(s2) / s1.compareTo(s2)
```

```
"abc" == "abc" 
```

# Argumente de la linia de comandă

```
public class Argumente {  
    public static void main (String args[]) {  
        if (args.length < 3) {  
            System.out.println("Numar insuficient de argumente!");  
            System.exit(-1); // termina aplicatia  
        }  
        String sir = args[0];  
        int numarIntreg = Integer.parseInt(args[1]);  
        double numarReal = Double.parseDouble(args[2]);  
    }  
}
```

java Argumente "Hello" 2014 1.7

# Bibliografie

- ***The Java Tutorials***

<http://docs.oracle.com/javase/tutorial/>

- ***The Java Language Specification***, James Gosling, Bill Joy, Guy Steele, Gilad Bracha

- ***The Java Virtual Machine Specification***

Tim Lindholm, Frank Yellin

- ***Curs practic de Java***, C. Frăsinaru

- <http://www.infoiasi.ro/~acf/java>

- <http://cfrasinaru.dyndns.org/acf>