

0101010
0100101
1101010

UD1.1- INTRODUCCIÓ A LA PROGRAMACIÓ AMB JAVA

0485 - Programació
1r DAW/DAM

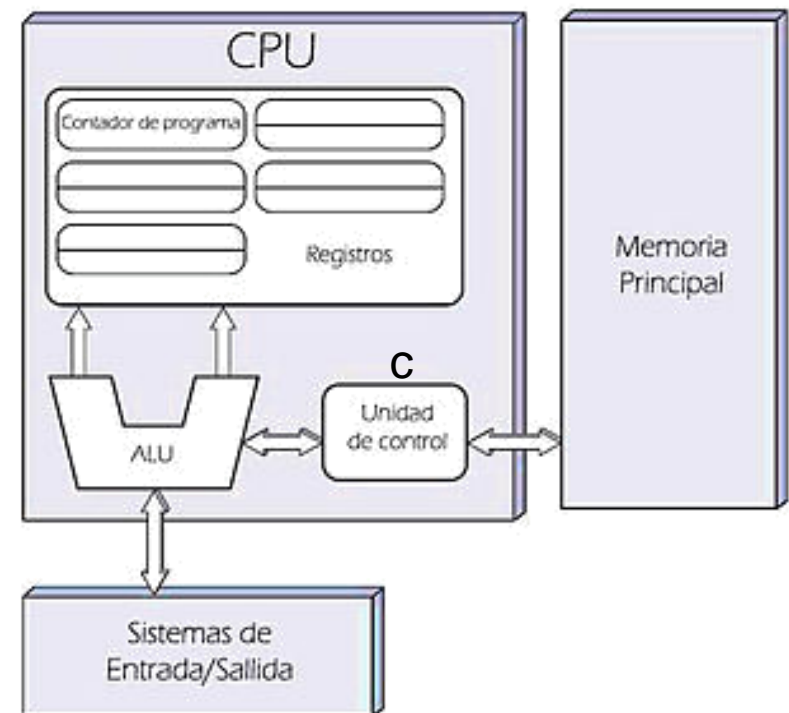
0. CONTINGUTS

- **Introducció**
- **Què es Java?**
- **Característiques bàsiques**
- **Entorn de desenvolupament**
- **Estructura de un programa**
- **Entorn de desenvolupament Integrat (IDE)**

1. INTRODUCCIÓ

- Una volta tenim dissenyat **l'algorisme** sabem: "*qué ordres*", "*en quin ordre*" i sobre "*quines dades*" **han de ser executades** per **solventar el problema**.

- ✓ Cada una d'aquestes **ordres** hauran de traduir-se **en instruccions** i quedaran vinculades a 1 o **vàries unitats funcionals** del computador.

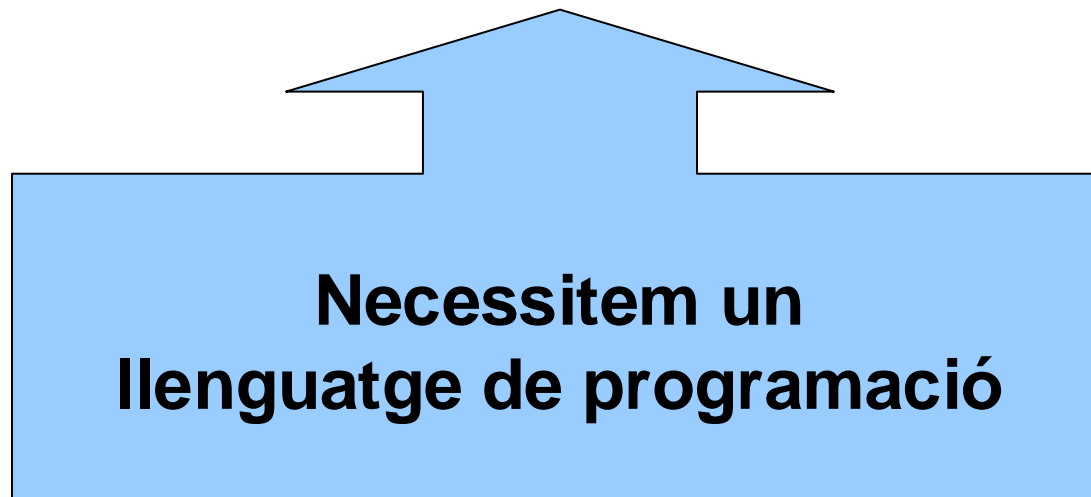


1.1 Ordres e Instruccions

Ordre	Element funcional
Llegir un número introduït per un usuari	E/S (teclat)
Guardar el resultat d'una operació	Memòria principal
Guardar una cadena introduïda per un usuari	Memòria principal
Fer una suma / resta / multiplicació	Unitat aritmètic-lògica
Mostrar el resultat d'una operació	E/S (pantalla)

1.2 Què és un programa?

- **Conjunt d'instruccions** que se li proporcionen a un ordinador en un llenguatge que ell pot entendre, per a dir-li exactament què volem que faci.



1.3 Tipus de Llenguatges

Llenguatge màquina: és el llenguatge que l'ordinador entén directament:

- Les **instruccions s'expressen en binari**.
- Les **dades** es referencien mitjançant **l'adreça de memòria on es troben**. No apareixen noms.
- **Dependència del processador de l'ordinador**. Un programa escrit per a un ordinador no es pot executar en un altre (baixa portabilitat).

1011	00000010	00000001	
		++++++	Operando 1
	+++++		Operando 2
+++++			Sumar

1.3 Tipus de Llenguatges

Assemblador

- És el primer intent de substituir el **llenguatge màquina** per un més proper al nostre.



```

Add    rax, 25    // rax = rax+25
|        |        |
|        |        ++++++ número 25
|        |+++++++ Registro rax
|+++++++ Operación suma

```

Ej. Instrucción suma arquitectura x86-64 (Amd64 o Intel 64)

Contínua depenent del processador de la màquina

1.3 Tipus de Llenguatges

D'alt nivell

- ❖ Independents de l'arquitectura de l'ordinador.
- ❖ Més propers al llenguatge natural

1 instrucció en
llenguatge
d'alt nivell

Traductor

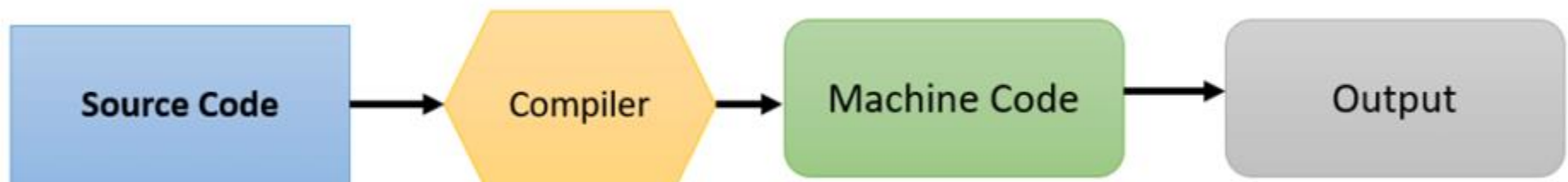
Vàries instruccions
en llenguatge màquina

```
public class Activitat4 {  
  
    public static void main(String[] args) {  
  
        int x = 144;  
        int i = 999;  
  
        int plusResult = x + i;  
        System.out.println("La suma és: " + plusResult);  
  
        int subResult = x - i;  
        System.out.println("La resta és: " + subResult);  
  
        float divResult = x / (float) i;  
        System.out.println("La divisió és: " + divResult);  
  
        int multResult = x * i;  
        System.out.println("La multiplicació és: " + multResult);  
  
    }  
}
```


1.4 Traductors

Compiladors

- **Tradueixen** completament un **programa font** (aquell que està escrit en un llenguatge d'alt nivell) i generen un **programa objecte** (semànticament equivalent) escrit en llenguatge de baix nivell.
 - El **compilador informa** a l'usuari dels **possibles errors** en el programa font. El programa objecte **sols es crearà si no n'hi ha errors** en el codi font.
 - El **programa objecte** s'emmagatzemarà en un **fitxer** que es **podrà executar sense necessitat de tornar-lo a traduir**.



1.4 Traductors

Intèrprets

- Permeten que un programa font vaja traduïnt-se i executant-se directament sentència a sentència per l'ordinador.
- Llig una sentència del codi font, la analitza i la interpreta i executant-la directament. (**No es crea cap fitxer objecte**).



2. QUÈ ÉS JAVA?

- **Java** és un **llenguatge de programació de propòsit general** i, per tant, és vàlid per a desenvolupar qualsevol tipus d'aplicacions professionals.
- Els programes “**executables**”, creats pel compilador de Java, són **independents de l'arquitectura**.
- **S'executen** en una **gran varietat de dispositius** amb diferents **microprocessadors** i **sistemes operatius**.



2.3 Característiques

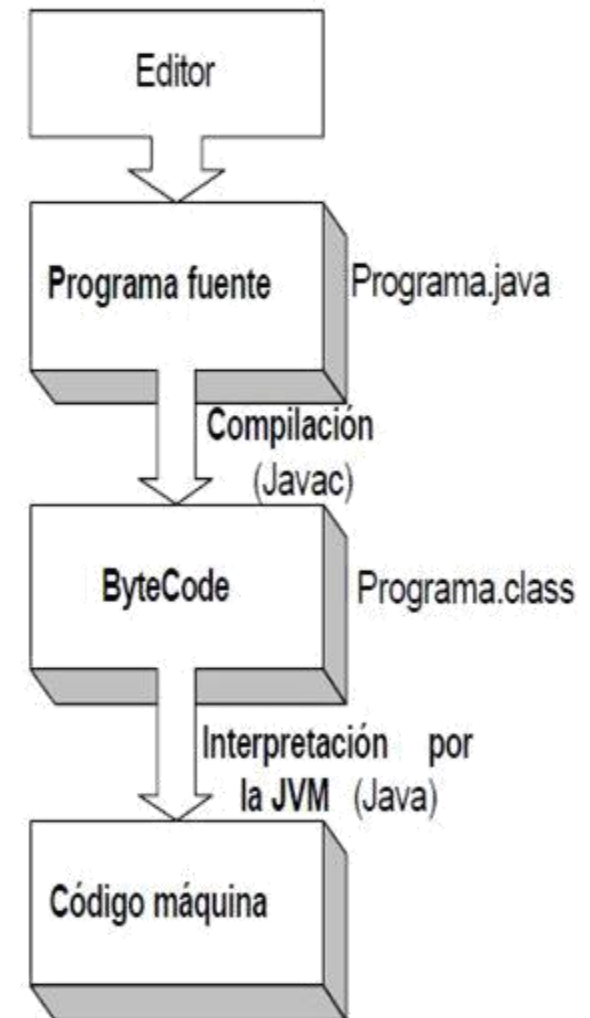
- És intrínsecament **orientat a objectes (OO)**.
- Llenguatge amb **tipat fort, estàtic i explícit**.
- Funciona **perfectament en xarxa**.
- Aprofita **característiques de la majoria** dels llenguatges moderns **evitant els seus inconvenients**.

*P.E **NO** té **punters** que haja de manejar el programador encara que els gestiona internament de forma transparent.*

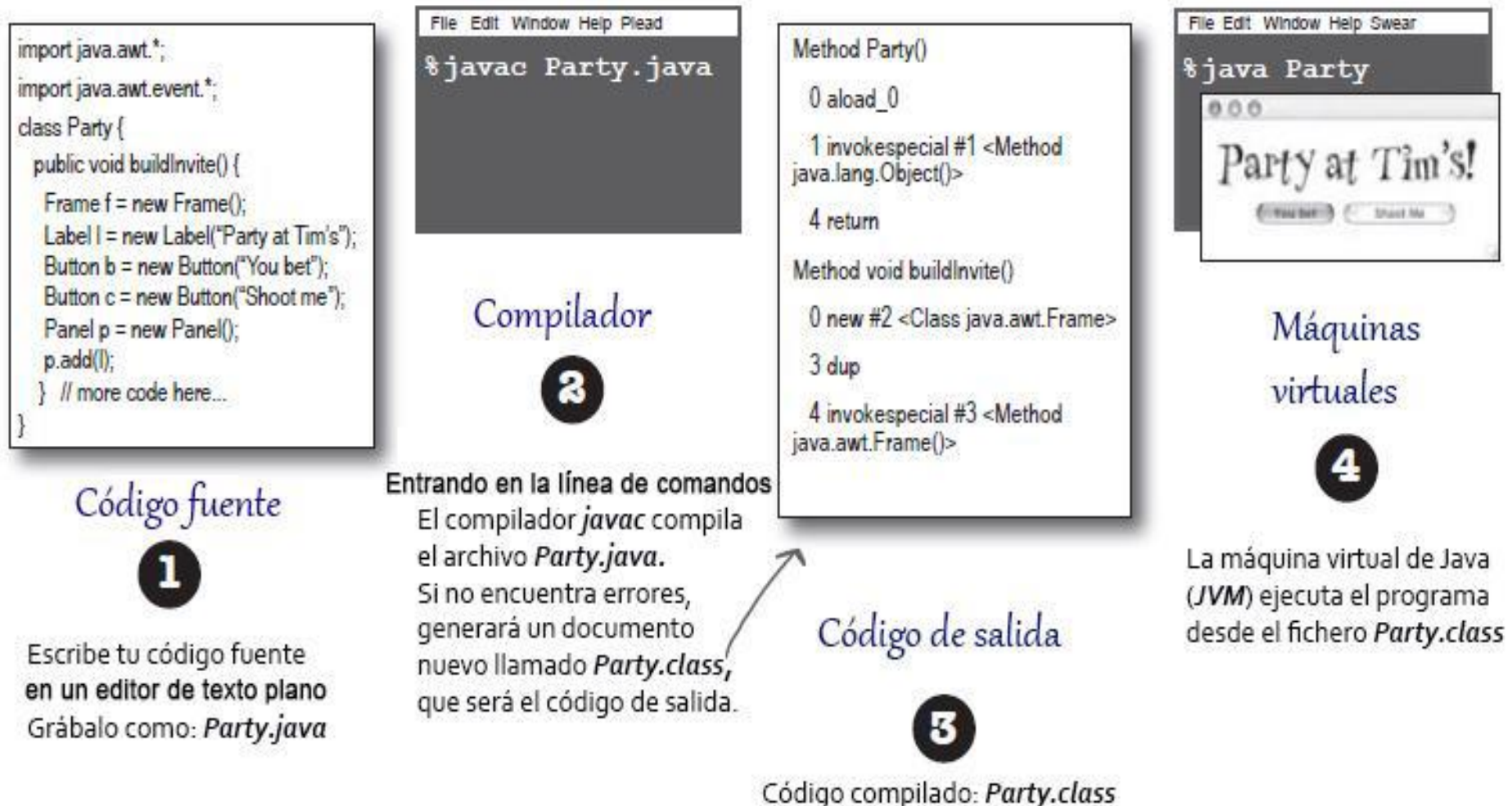
- Té **una gran funcionalitat** gràcies a les seues llibreries (classes).

2.4 És Java compilat o interpretat?

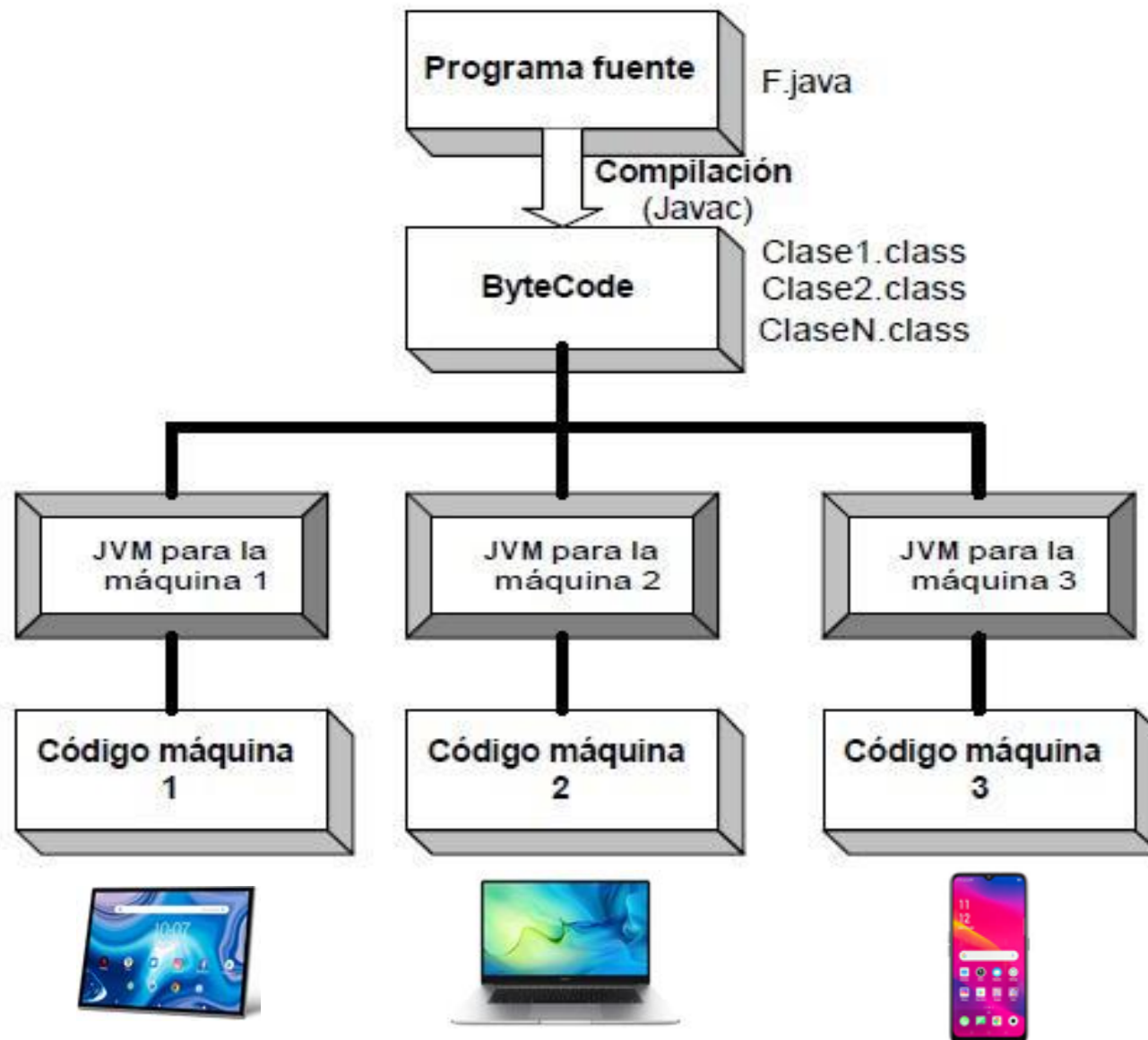
- Encara que estrictament parlant és interpretat, **necessita un procés previ de compilació.**
- Una vegada “compilat” el programa, **es crea un fitxer que emmagatzema *bytecodes*** (pseudocodi bàsicament al nivell de llenguatge màquina).
- **Per executar-lo, és necessari un “intèrpret”, la JVM (*Java Virtual Machine*)**
- És possible **compilar el programa** en una màquina amb **Linux** i **executar-la** en una altra amb **Windows** utilitzant la màquina virtual Java
- **La JVM** llig els *bytecodes* i els tradueix a instruccions executables pel processador.



2.6 Com treballa Java?(II)

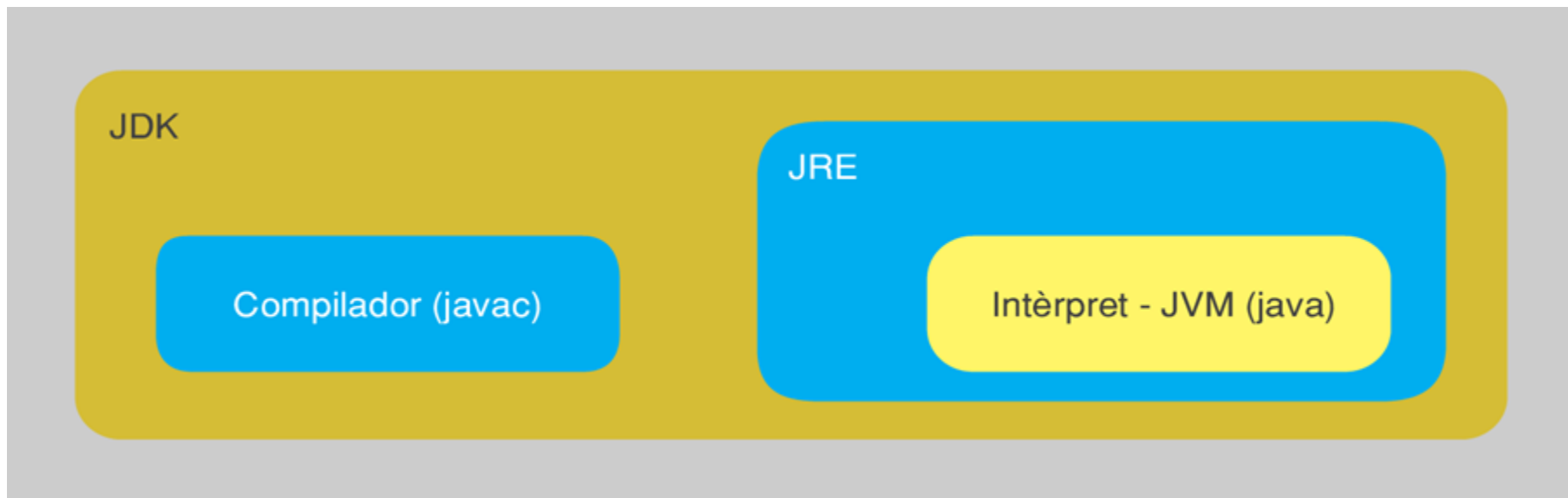


2.7 La màquina virtual Java(I)



3. L'ENTORN DE DESENVOLUPAMENT (JDK)

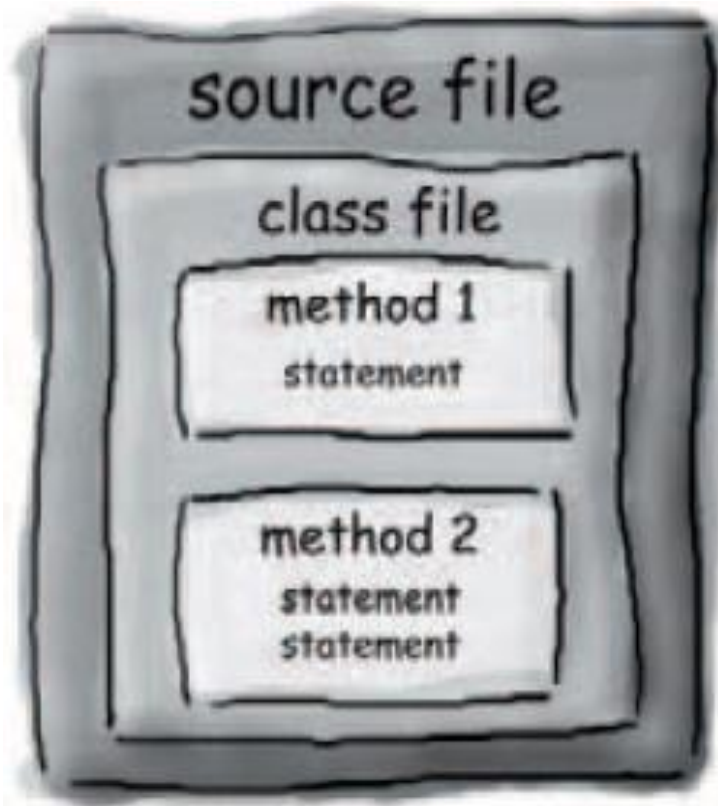
- **L'eina bàsica** per començar a desenvolupar aplicacions en Java és el **JDK** (*Java Developer's Kit*).
- Consisteix bàsicament en un compilador i un intèrpret (JVM) per a la línia de comandes.



3. L'ENTORN DE DESENVOLUPAMENT (JDK)

- El *kit* conté bàsicament:
 - El **compilador**: **javac**
 - El depurador: **jdb**
 - L'**intèrpret**: **java** i **javaw**
 - El generador de documentació: **javadoc**
 - Un desensamblador de classes: **javap**
 - El generador d'arxius font i de capçalera (.c i .h) per a classes natives en C: **javah**

4. ESTRUCTURA D'UN PROGRAMA



- **Source file:** fitxer font
- **Class file:** fitxer de classe
- **Method:** mètode
- **Statements:** sentències

4.1 Què conté un fitxer .java?

```
public class Dog {
```

```
}
```

clase

- Conté, almenys, la definició d'una classe. El contingut ha d'anar entre { i }.
- Extensió **.java**.
- Hi ha una convenció per donar noms als fitxers:
 - **UpperCamelCase**
 - inicial de cada paraula en majúscula (preferentment sense accents ni espais)
 - Exemples: **HolaMundo.java**, **Prova.java**, **ElMeuPrograma.java**,...

4.2 Què conté una classe?

```
public class Dog {  
    void bark() {  
  
    }  
}
```

método

- Conténdrà, **almenys un mètode**.
- Per a declarar una classe: class **NomClasse**. El contingut anirà entre { i }.
- El **nom de la classe** ha de coincidir exactament amb el **nom del fitxer** que conté el codi.
 - Tindrà l'extensió **.java**.

MOLT IMPORTANT!!! Java és sensible a majúscules (*case sensitive*). **Holamon, HolaMon i holaMon són diferents.**

4.3 Què conté un mètode?

```
public class Dog {  
    void bark() {  
        statement1;  
        statement2;  
    }  
}
```

sentencias

- Conjunt de **sentències** / **instruccions** que realitzen una acció.

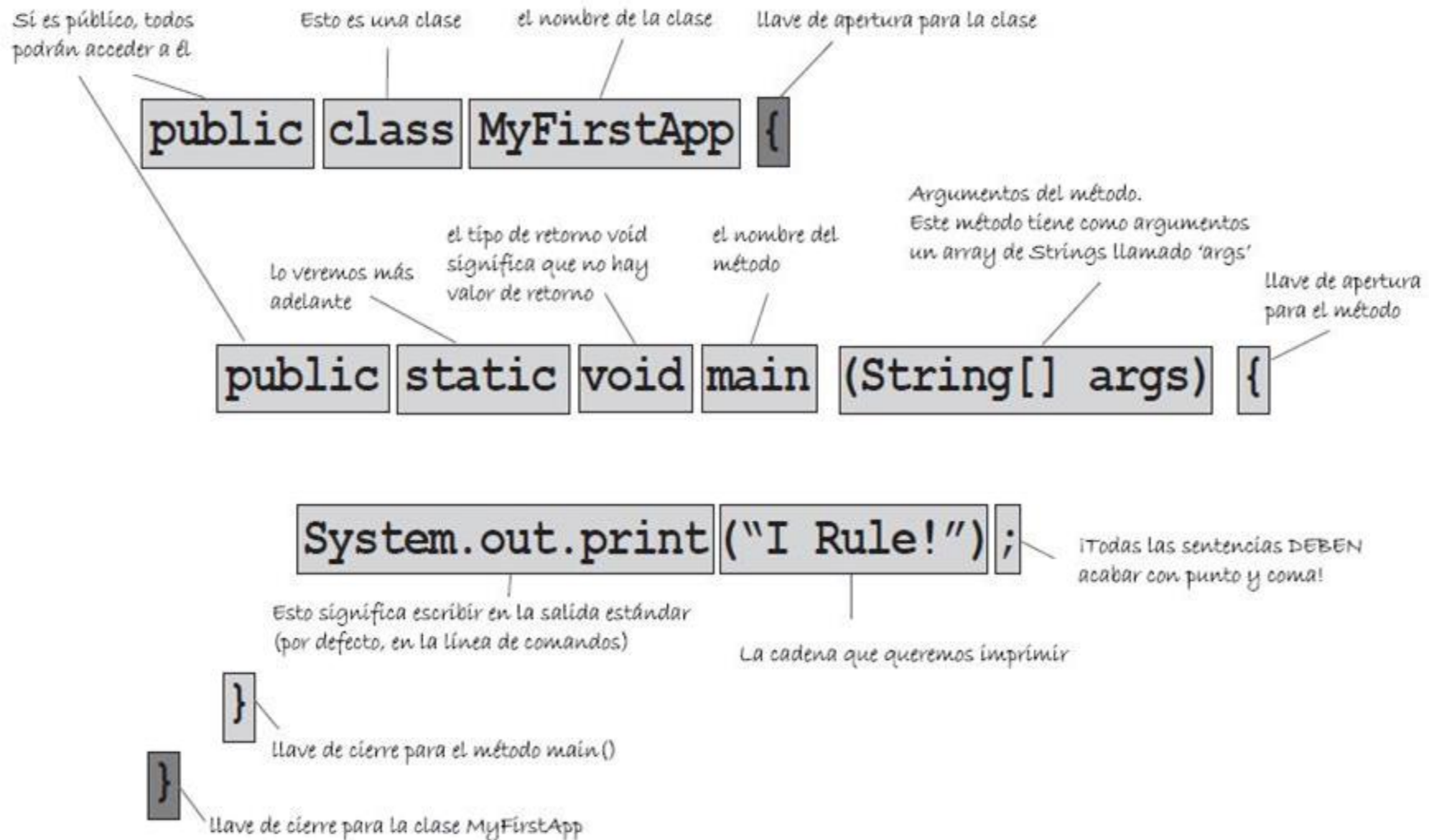
4.4 Execució de programes

- Quan **JVM arranca** (a l'executar), busca la classe creada i, concretament, busca el mètode `main()`

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

- Cada aplicació Java té, almenys, **una classe i un mètode**.
 - **Correcte:** un `main()` per aplicació
 - **Incorrecte:** un `main()` per classe

4.5 Anatomía d'una classe



5. ENTORN DE DESENVOLUPAMENT INTEGRAT (IDE)

- Faciliten el desenvolupament de programes mitjançant:
 - L'automatització de tasques de **compilació/execució**
 - Anàlisi **d'errors en temps de compilació**
 - **Auto-completat**
 - ...

Nosaltres utilitzarem de moment ...



Activitat Prèvia

- **Activitat G.-** Instal·la l'IDE **Apache Netbeans** al teu ordinador. Crea i executa un programa que mostre per pantalla:

“Hola món {{NOM ALUMNE}}”

```
public class HolaMon {  
    public static void main(String[] args) {  
        System.out.println(x: "Hola món -EL TEU NOM-");  
    }  
}
```

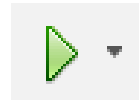
Activitat Prèvia

La classe principal s'anomenarà "HolaMon"

- (1)
- (2) *Java with Ant -> Java App*
- (3) *Project*

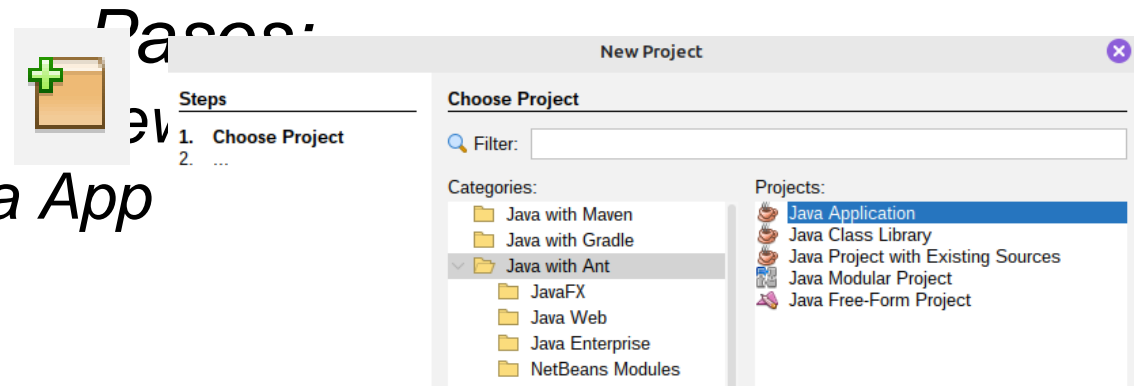
☒ Create Main Class

- (4) *Escriu codi*
- (5) *Executa*



Output - HolaMon (run) x

```
run:
Hola món -EL TEU NOM-
BUILD SUCCESSFUL (total time: 0 seconds)
```



Una volta executat el programa, investiga tots el arxius associats al projecte creat... quins fitxers corresponen al codi font? quins al bytecode?

5.1 Activitat Prèvia

- Això és tot ... de moment :-)

FinAlgoritmo