



Ingegneria del Software

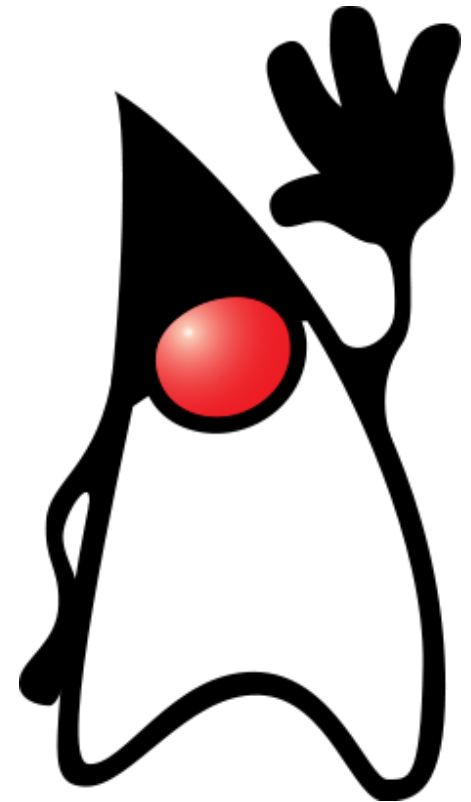
Esercitazione 1: Programmazione Java

Valerio Panzica La Manna
valerio.panzicalamanna@polimi.it

Source code at : <https://github.com/panzica/IngSoftw2014-PoliMI>

Java

- Linguaggio di Programmazione:
 - Object-oriented
 - Class-based
 - Compilato
 - A tipizzazione statica
- Caratteristica fondamentale:
Portabilità
Write Once, Run Anywhere
 - Lo stesso codice è eseguito in qualsiasi piattaforma.



Esercizio 1: Welcome

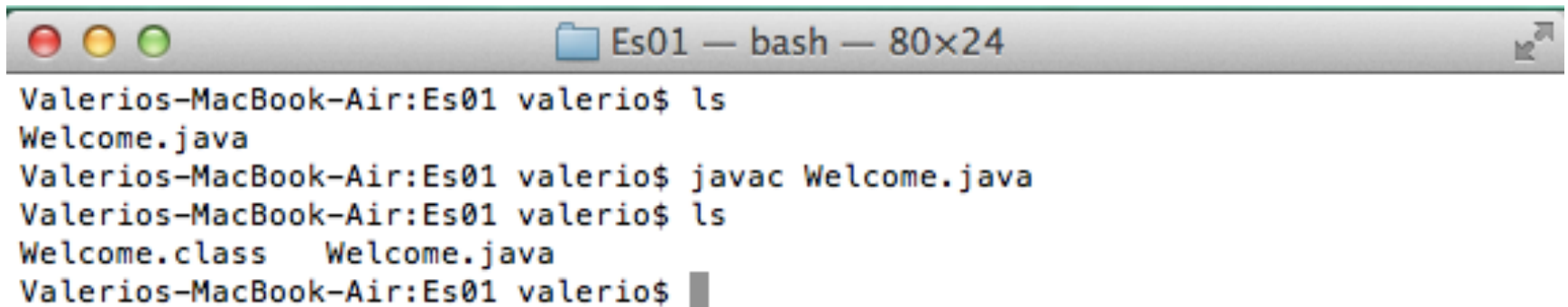
- Step 1 : Scrivere un programma Java che stampa a video:
“Benvenuto al corso di Ingegneria del Software”

Step 1

```
public class Welcome {  
    public static void main(String[] args){  
        System.out.println("Benvenuto al corso  
di Ingegneria del Software");  
    }  
}
```

Compilazione

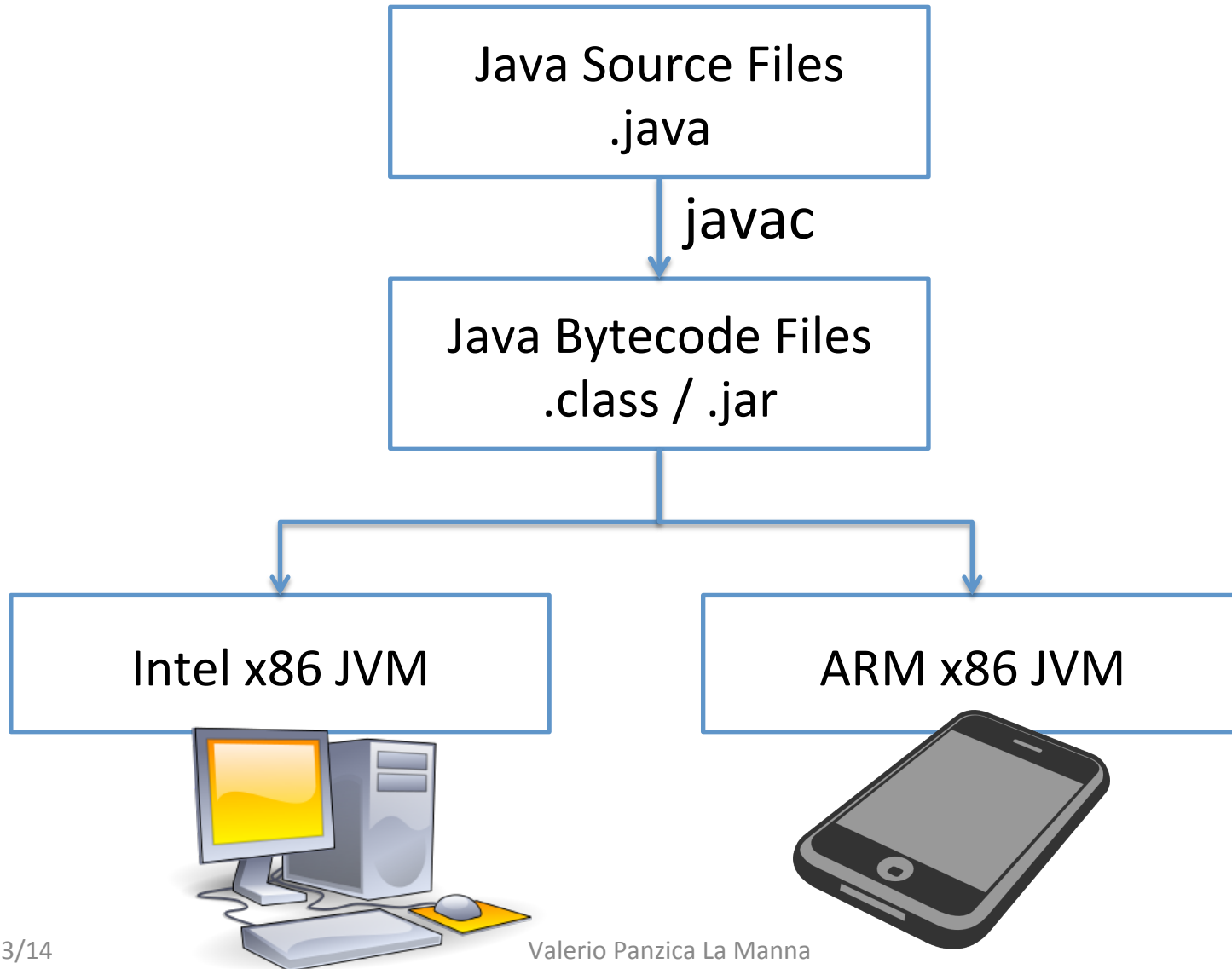
- Utilizziamo il programma javac
javac nomeFile.java



```
Valerios-MacBook-Air:Es01 valerio$ ls
Welcome.java
Valerios-MacBook-Air:Es01 valerio$ javac Welcome.java
Valerios-MacBook-Air:Es01 valerio$ ls
Welcome.class  Welcome.java
Valerios-MacBook-Air:Es01 valerio$
```

The image shows a macOS terminal window titled "Es01 — bash — 80x24". The terminal displays the following commands and output: a directory listing showing "Welcome.java", the execution of "javac Welcome.java", another directory listing showing both "Welcome.class" and "Welcome.java", and a final prompt line.

- Genera il file java bytecode .class



Esecuzione

- Utilizziamo il programma java
java nomeFile

```
Valerios-MacBook-Air:Es01 valerio$ ls
Welcome.java
Valerios-MacBook-Air:Es01 valerio$ javac Welcome.java
Valerios-MacBook-Air:Es01 valerio$ ls
Welcome.class  Welcome.java
Valerios-MacBook-Air:Es01 valerio$ java Welcome
Benvenuto al corso di Ingegneria del Software
```

- Esegue il metodo main della classe

Esercizio 1: Welcome

- Step 1 : Scrivere un programma Java che stampa a video:
“Benvenuto al corso di Ingegneria del Software”
- Step 2: Modificare il programma:
 - L’utente inserisce il proprio nome. Es. Valerio
 - Il programma stampa a video:
 - “Ciao Valerio, Benvenuto al corso di Ingegneria del Software”.


```
import java.util.Scanner;

public class WelcomeWithName {
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        System.out.println("Inserisci il tuo nome:");
        String nome = scanner.nextLine();
        scanner.close();
        String welcome = "Benvenuto al corso di Ingegneria
del Software";
        System.out.println("Ciao " + nome + ", " +
welcome);
    }
}
```

Oggetti e Classi

- Esercizio 2: Modellare in Java una Bicicletta
- Domande da porsi (o da porre al cliente):
 - Come possiamo definire lo **stato** di una bicicletta?
 - Che **funzionalità** deve offrire la bicicletta?

Oggetti e Classi

- Esercizio 2: Modellare in Java una Bicicletta
- Domande da porsi (o da porre al cliente):
- Come possiamo definire lo **stato** di una bicicletta?
 - Una bicicletta ha una certa velocità alla quale si sta muovendo e questa è conseguenza della marcia e del ritmo di pedalata.
- Che **funzionalità** deve offrire la bicicletta?
 - Monitorare la velocità, la marcia utilizzata, e il ritmo di pedalata.
 - Aumentare e Diminuire la marcia della bicicletta e il ritmo di pedalata.
 - Frenare.
- **PROVIAMO**

Implementare un Client

- Finora è stata solamente definita la classe Bike.
 - Non abbiamo istanziato oggetti di tipo Bike
 - Non abbiamo un main dal quale far partire l'applicazione
- 2 Possibilità:
 1. Metodo main dentro la classe.
 2. Creare una classe client. VEDIAMO

```
public class BikeClient {  
    public static void main(String[] args){  
        // crea un oggetto Bike utilizzo il costruttore di default  
        Bike b1 = new Bike();  
        // modifico lo stato dell'oggetto tramite metodi  
        b1.changeCadence(2);  
        b1.incrementGear();  
        b1.incrementGear();  
        b1.printState();  
        b1.brake();  
        System.out.println("The speed now is "+ b1.getSpeed());  
    }  
}
```

Compiliamo ed Eseguiamo BikeClient

```
Valerios-MacBook-Air:01 valerio$ javac es02/BikeClient.java
Valerios-MacBook-Air:01 valerio$ java es02/BikeClient
gear: 2, speed: 4, cadence: 2
The speed now is 0
Valerios-MacBook-Air:01 valerio$ █
```

Packages

```
Valerios-MacBook-Air:01 valerio$ javac es02/BikeClient.java
Valerios-MacBook-Air:01 valerio$ java es02/BikeClient
gear: 2, speed: 4, cadence: 2
The speed now is 0
Valerios-MacBook-Air:01 valerio$
```

- Le classi java sono organizzate in cartelle chiamate **package**
- Il package va definito anche nel codice **package es02;**

Cambiamo la cartella

```
Valerios-MacBook-Air:01 valerio$ javac myes02/BikeClient.java
Valerios-MacBook-Air:01 valerio$ java myes02/BikeClient
Exception in thread "main" java.lang.NoClassDefFoundError: myes02/BikeClient (wrong name: es02/BikeClient)
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:800)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:449)
    at java.net.URLClassLoader.access$100(URLClassLoader.java:71)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:361)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:425)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:358)
    at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:482)
Valerios-MacBook-Air:01 valerio$ █
```

- Errore!
- Non trova il file es02/BikeClient

IDE

- Gli IDE (Integrated Development Environment) vi sollevano da tutte le difficoltà relative alla compilazione e alla gestione di cartelle e packages
- Inoltre vi offrono moltissime funzionalità come l'autocompletamento del codice
- Tuttavia il concetto di package rimane presente in Java e risulta importante per la visibilità delle classi (vedremo)
- In laboratorio userete l'IDE Eclipse. Tuttavia online sono presenti tantissimi altri IDE.
- Netbeans è il principale concorrente di Eclipse.

- Esercizio: Modificare la classe Bike affinché sia possibile istanziare una bicicletta con velocità, marcia e cadenza definite dall'utente.
- Fino ad adesso:

```
private int gear = 1;  
private int cadence;  
private int speed;
```

Esercizio 2.1

- Modificare la classe Bike affinché sia possibile istanziare una bicicletta con velocità, marcia e cadenza definite dall'utente.
- Fino ad adesso:

```
private int gear = 1;  
private int cadence;  
private int speed;
```
- Creiamo un costruttore

// Modifica: creazione di un costruttore

```
public Bike(int gear, int speed, int cadence ){  
    this.gear = gear;  
    this.speed = speed;  
    this.cadence = cadence;  
}
```

// Modifica: aggiungo costruttore di default vuoto

```
public Bike(){  
  
}
```

- La keyword **this** si riferisce all'oggetto e permette di risolvere ambiguità.

Aggiorniamo BikeClient

```
System.out.println("Creating a new object b2");  
Bike b2 = new Bike(3,10,20);  
System.out.println("The initial speed of b2 is: " +  
b2.getSpeed());
```

Esercizio 2.2

- Modificare la classe Bike affinché ogni bicicletta nel sistema sia identificata univocamente da un id.
- Due variabili di tipo Bike saranno uguali se e solo se avranno lo stesso id.
- Modificare Bike e testare con BikeClient
- Vediamo

Metodo equals

```
@Override //aggiungo annotazione  
public boolean equals(Object o){  
    Bike other = (Bike) o; //casting  
necessario per accedere a id  
    return other.id == this.id;  
}
```

- Nota bene: Il metodo equals è un metodo che viene usato convenzionalmente per dire se due determinati oggetti sono uguali o meno.
- Il programmatore può benissimo scegliere un altro nome, tuttavia sarà una sua convenzione e non potrà usufruire dei vantaggi che Java mette a disposizione per equals (li vedremo in dettaglio)

Tipi di riferimento

- Tutte le variabili che sono istanze di classi Java sono di tipo riferimento. Tali variabili contengono l'indirizzo di memoria degli oggetti ai quali si riferiscono.

Es 2.3 Cosa Stampa?

```
public class TipiDiRiferimento {  
    public static void main(String [] args){  
        Bike b1 = new Bike(2,10,4);  
        Bike b2 = b1;  
        System.out.println("b1 e b2 sono uguali? " +  
b1.equals(b2) );  
        System.out.println("b1 e b2 sono lo stesso oggetto?  
" + (b1 == b2));  
        // stampo lo stato di b1  
        b1.printState();  
        // stampo lo stato di b2  
        b2.printState();  
        // incremento la marcia di b2  
        b2.incrementGear();  
        // stampo lo stato di b1  
        b1.printState();  
    }  
}
```