

Linguaggi di Programmazione

a.a. 13/14

docente: Gabriele Fici

gabriele.fici@unipa.it

2 - Primi passi in Java

2 - Primi passi in Java

- I programmi Java sono divisi in “*classi*”
- Le classi sono descritte all'interno di file di testo con estensione `.java`
- Ogni file `.java` deve contenere al più una classe pubblica, che deve avere per nome il nome del file (es. il file `pippo.java` contiene la classe pubblica `pippo`, ed eventualmente altre classi)
- Almeno una classe pubblica deve contenere il metodo `main`, cioè il punto di accesso al programma

2 - Primi passi in Java

- Esempio: contenuto del file `Hello.java`

```
/** Questa classe stampa un messaggio */  
  
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

2 - Primi passi in Java

- Da un terminale (o finestra DOS) aperto sulla cartella che contiene il file `Hello.java` digitare:

```
javac Hello.java
```

- Si genera così il file `Hello.class` che contiene il ByteCode (istruzioni macchina per la JVM)

- Per eseguire il programma `Hello` bisogna digitare:

```
java Hello
```

- Per generare la documentazione della classe digitare:

```
javadoc Hello.java
```

2 - Primi passi in Java

Esistono 3 tipi di commento al codice:

- Commenti su una riga:

```
// ...
```

- Commenti su più righe:

```
/* ... */
```

- Commenti per creare documentazione Javadoc:

```
/** ... */
```

2 - Primi passi in Java

Letterali:

- booleani: `true` `false`
- caratteri (simboli o codifiche *Unicode*) racchiusi fra apici:
`'a'` `'\u3abc'`
- caratteri di controllo: `'\n'` `'\t'` `'\\'` `'\''`
- stringhe, racchiuse tra virgolette: `"ciao"` `"qui\n"`

2 - Primi passi in Java

Letterali:

- interi (int: suffisso assente; long: suffisso `L` o `l`):
 - base decimale, `0` o numero che inizia per `1..9`, es: `0` `129L`
 - base ottale, inizia con `0`, es: `0127` `0777`
 - base esadecimale: inizia con `0x`, es: `0x127` `0x7ad2`
- reali (float: suffisso `f` o `F`; double: suffisso assente o `d` o `D`):
costituiti da parte intera, punto decimale, parte frazionaria, esponente preceduto da `e` o `E`; alcuni elementi possono mancare; deve esserci almeno una cifra nella parte intera o in quella frazionaria, e almeno un elemento fra punto decimale, esponente, suffisso, es:

`2.5f`, `2.5`, `2d`, `2e2f`, `2e2`, `2.`, `.2`, `0f`, `0.0`

2 - Primi passi in Java

I tipi primitivi di Java sono:

- `byte` - intero di 8 bit con segno
- `short` - intero di 16 bit con segno
- `int` - intero di 32 bit con segno
- `long` - intero di 64 bit con segno
- `float` - floating-point di 32 bit
- `double` - floating-point di 64 bit
- `boolean` - tipo enumerativo (due elementi: `true` e `false`)
- `char` - carattere di 16 bit (Unicode)

2 - Primi passi in Java

Le dichiarazioni di variabili, per i tipi primitivi, si fanno come in C:

- `int width; // dichiarazione di un intero`
- `int width=10; // dichiarazione e assegnazione`
- `int width, height; // dichiarazione multipla`
- `int 2map; // Errore! Inizia con una cifra`
- `int class; // Errore! Parola riservata`

Suggerimento: usare sempre iniziali minuscole per le variabili di tipi fondamentali!

2 - Primi passi in Java

L'attributo `final` rende la variabile immutabile (equivale alle costanti):

```
final int n = 10;  
  
n = 20;           // errore!
```

2 - Primi passi in Java

Operatori per i booleani:

- Operatori di relazione: `==`, `!=`
- Operatore di complemento logico: `!`
- Operatori logici: `&` (and), `|` (or), `^` (or esclusivo)
- Operatori `and_condizionale` e `or_condizionale` : `&&`, `||`
(valutano l'operando destro solo se quello sinistro vale `true` o `false`, rispettivamente)

2 - Primi passi in Java

Operatori per i tipi numerici:

- Operatori di confronto, che producono un risultato di tipo boolean:
 - confronto numerico: `<`, `<=`, `>`, `>=`
 - uguaglianza numerica: `==`, `!=`
- Operatori numerici, che producono un risultato di tipo dello stesso tipo numerico:
 - più e meno unari: `+`, `-`
 - incremento (unario), prefisso e postfisso: `++`
 - decremento (unario), prefisso e postfisso: `--`(si applicano a una variabile e restituiscono un valore, quello della variabile modificata o meno)
 - complemento (unario) bit a bit: `~`
 - operatori moltiplicativi: `*`, `/`, `%`
 - operatori additivi: `+`, `-`

2 - Primi passi in Java

Operatori per i tipi numerici:

- Operatori bitwise, solo per tipi interi:
 - shift a sinistra: `<<`
 - shift a destra replicando il segno: `>>`
 - shift a destra inserendo 0: `>>>`
 - operatori bit a bit: `&` (and), `|` (or), `^` (or esclusivo)

2 - Primi passi in Java

Operatore di assegnamento “`=`”:

- operandi: valore a destra, variabile a sinistra;
- produce un risultato (valore dell'espressione a destra dell'operatore);
- ha un effetto collaterale (assegnamento del risultato alla variabile a sinistra dell'operatore);
- può comparire all'interno di un'espressione, es:

```
int a, b, c, d;
```

```
a = (b = c + d) ;
```

2 - Primi passi in Java

Altri operatori di assegnamento:

`+=`

`a = a+b` si può scrivere anche `a += b`

`-=`, `*=`, etc.

idem

2 - Primi passi in Java

Altri operatori (tutti i tipi):

- Operatore condizionale:

`<condizione> ? <valore1> : <valore2>`

- Operatore di concatenazione "+":

`"pippo" + 2.54`