

Programmazione 1 - Modulo C

Java: Visibilità di una variabile locale

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

Ottobre 2020



Blocchi

- Abbiamo visto che se in un ciclo **while** devono essere eseguiti più comandi ci vuole un **blocco**
- Un blocco é una sequenza di comandi racchiusa tra parentesi graffe

```
while (x>0){  
    somma=somma+x;  
    x--;  
    System.out.println("Valore di x: "+x+" Valore di somma: "+somma+ " invariante  $x(x+1)/2 + \text{somma}$ : "+(x*(x+1)/2+somma));  
}
```

Blocchi

- Il corpo del metodo **main** é un altro esempio di blocco;

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

- i blocchi possono essere **annidati**

```
public static void main (String[] args){  
    // dichiarazione ed inizializzazione  
    int x=10,somma=0;  
    while (x>0){  
        somma=somma+x;  
        x--;  
        System.out.println("Valore di x: "+x+" Valore di somma: "+somma+ " invariante  $x*(x+1)/2 + somma = x*(x+1)/2 + somma$ ");  
    }  
    System.out.println("Valore finale di somma: "+somma);  
}
```

Variabili locali

- le variabili dichiarate in un blocco sono **locali al blocco** e, quindi, **scompaiono dopo l'esecuzione del blocco**.

```
if ( saldo >=0 ) {  
    System.out.println(" Saldo positivo ");  
    double interesse = saldo * tasso ;  
    saldo = saldo + interesse ;  
}  
  
System.out.println( interesse ); // ERRORE !
```

Visibilità Variabili locali

- Si può definire una nozione di **visibilità (o scope)** di una variabile:

La **visibilità (o scope)** di una variabile è la **porzione di programma** in cui tale variabile può essere utilizzata

- La visibilità di una **variabile locale** (cioé definita in un blocco) é la porzione di programma che va **dalla dichiarazione** della variabile stessa alla **fine del blocco** che la contiene;

Visibilità Variabili locali

- Limitare la visibilità di una variabile al blocco che la contiene consente di:
 - ▶ **riutilizzare nomi** di variabili in parti diverse del programma;
 - ▶ anche con tipi diversi.

```
boolean leggiIntero = SIn.readLineBoolean();  
if ( leggiIntero ) {  
    int val = SIn.readLineInt();  
    System.out.print ("Il numero letto e ':"+val);  
}  
else {  
    double val = SIn.readLineDouble()  
    System . out . print ("Il numero letto e ':"+val);  
}
```

Visibilità Variabili locali

- Le variabili dichiarate in un blocco non possono avere il nome di variabili esterne al blocco stesso;
- Il compilatore segnalerebbe un errore

```
int a =0;
if (true){
    int a =0; // ERRORE : variabile già ' dichiarata
    int b =10;
    System.out.println (a+ b );
}
```