

Programmazione in Java e gestione della grafica

Lezione 4

La scorsa lezione.....

- Prendere valori interi in input da tastiera
- Importare classi dai packages di Java
- Definire variabili
- Operatore di assegnamento (=)
- Operatori aritmetici (+, - *, /, %)
- Operatori condizionali (==, != , <=, >=)
- Istruzione **if**

Operatori di uguaglianza e relazionali

- Operatori logici (risultato VERO o FALSO)
- Le espressioni con operatori di questo tipo si chiamano **condizioni**

Prendere delle decisioni

- Istruzione **if**

```
if ( <condizione> ) <istruzione> ;
```

Prendere delle decisioni...

- *<condizione>*
 - Espressione che può essere **true** o **false**
- Istruzione **if**
 - Se la condizione è **true**, allora viene eseguito il corpo dell'istruzione **if**
 - Il controllo ritorna sempre dopo l'istruzione **if**
 - La *condizione* dell'istruzione **if** può essere formata usando gli operatori di uguaglianza e di relazione.

```

1 // Fig. 2.15: Comparison.java
2 // Compare integers using if statements, relational operators
3 // and equality operators.
4 import java.util.Scanner; // program uses class Scanner
5
6 public class Comparison
7 {
8     // main method begins execution of Java application
9     public static void main( String args[] )
10    {
11        // create Scanner to obtain input from command window
12        Scanner input = new Scanner( System.in );
13
14        int number1; // first number to compare
15        int number2; // second number to compare
16
17        System.out.print( "Enter first integer: " ); // prompt
18        number1 = input.nextInt(); // read first number from user
19
20        System.out.print( "Enter second integer: " ); // prompt
21        number2 = input.nextInt(); // read second number from user
22
23        if ( number1 == number2 )
24            System.out.printf( "%d == %d\n", number1, number2 );
25
26        if ( number1 != number2 )
27            System.out.printf( "%d != %d\n", number1, number2 );
28
29        if ( number1 < number2 )
30            System.out.printf( "%d < %d\n", number1,

```

Test for equality, display result using printf.

Compares two numbers using relational operator <.

Outline

Comparison.java

(1 of 2)

1. Class Comparison

1.1 main

1.2 Declarations

1.3 Input data (nextInt)

1.4 Compare two inputs using if statements



Outline

Comparison.java

(2 of 2)

Program output

```
31  if ( number1 > number2 )
32      System.out.printf( "%d > %d\n", number1, number2 );
33
34
35  if ( number1 <= number2 )
36      System.out.printf( "%d <= %d\n", number1,
37
38  if ( number1 >= number2 )
39      System.out.printf( "%d >= %d\n", number1, number2 );
40
41  } // end method main
42
43 } // end class Comparison
```

Compares two numbers using relational operators >, <= and >=.

```
Enter first integer: 777
Enter second integer: 777
777 == 777
777 <= 777
777 >= 777
```

```
Enter first integer: 1000
Enter second integer: 2000
1000 != 2000
1000 < 2000
1000 <= 2000
```

```
Enter first integer: 2000
Enter second integer: 1000
2000 != 1000
2000 > 1000
2000 >= 1000
```



Errori comuni (syntax error)

- Dimenticare le parentesi () nella condizione dell'`if`.
- Confondere l'operatore di uguaglianza, `==`, con l'operatore di assegnamento, `=`
- Inserire degli spazi tra i due simboli degli operatori `==`, `!=`, `>=` , `<=`
- Invertire gli operatori `!=`, `>=` , `<=`, as in `=!`, `=>` and `=<`, is a syntax error.

Un errore logico

Mettere un `;` dopo le parentesi `()` della condizione dell'`if`.

```
if (x == y) ;  
    x=3;
```

assegna a `x` il valore 3 in ogni caso....

Precedenza e associatività

Operators	Associativity	Type
* / %	left to right	multiplicative
+ -	left to right	additive
< <= > >=	left to right	relational
== !=	left to right	equality
=	right to left	assignment

Analizziamo il codice ...

larger2.java

larger3.java

larger2_bis.java

larger5.java

larger5_bis.java

Istruzione di selezione singola:

```
if ( <condizione>)  
    <istruzione> ;
```

Istruzione di selezione doppia:

```
if ( <condizione>)  
    <istruzione>;  
else  
    <istruzione>;
```



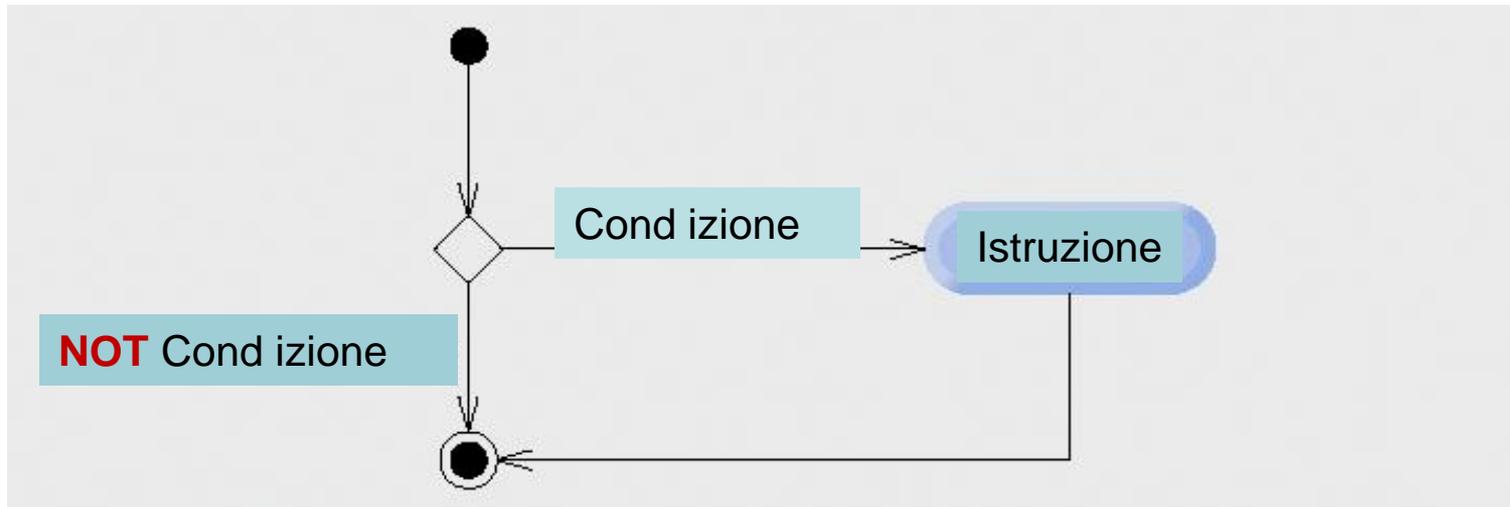


Diagramma per l'istruzione if.



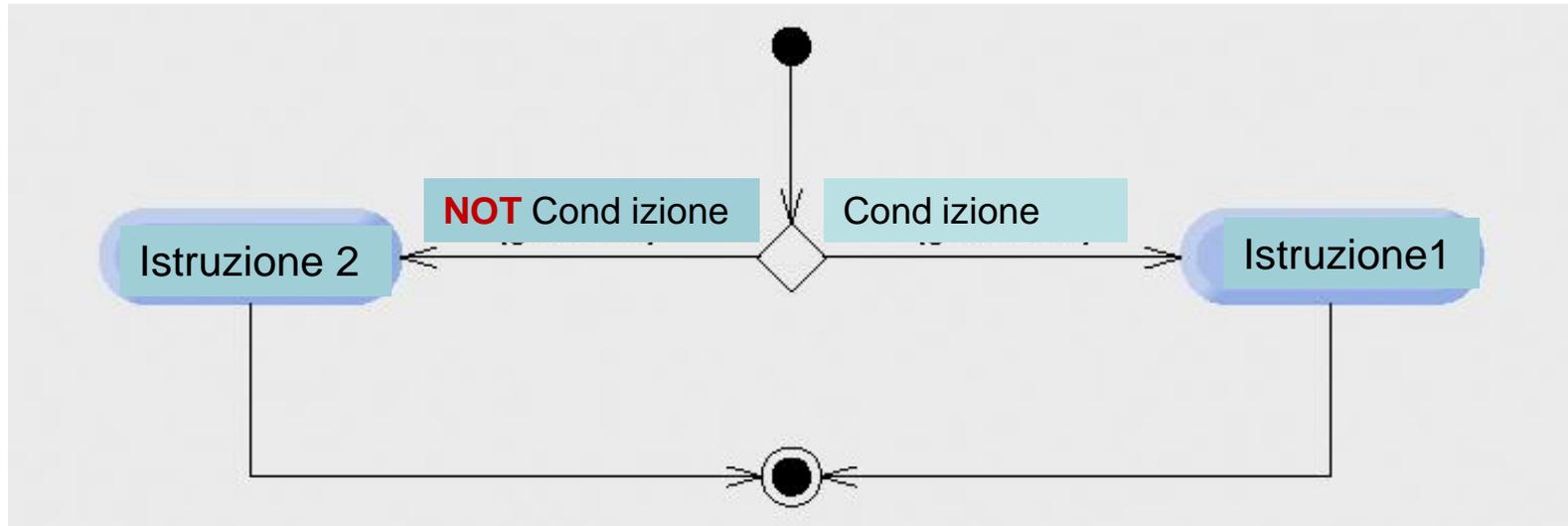


Diagramma per l'istruzione if..else.



Analizziamo il codice ...

Codice media3.java

Numeri a virgola mobile: requisiti di memoria e precisione

- **float**
 - Single-precision floating-point numbers
 - 7 cifre significative
- **double**
 - Double-precision floating-point numbers
 - 15 cifre significative

Tipi di dato float/double

- Specificatore di formato %f
 - Si usa per l'output di numeri floating-point
 - Un punto e un numero posti tra il % e la f specificano la precisione desiderata
 - Ex: %.2f stampa il numero con 2 cifre decimali

Esercizi

- Esercizio 4.1
Riscrivere `larger3.java` usando istruzione `if...else`
- Esercizio 4.2
Ampliare il programma `larger5_bis.java` in modo che calcoli sia il valore max che il valore min tra i 5 valori inseriti dall'utente

Esercizi

- Esercizio 4.3

Scrivere un programma Java che prende in input 8 valori interi e restituisce in output la somma e la media.

Il programma dovrà utilizzare soltanto 2 variabili.

- Esercizio 4.4

Scrivere un programma Java che prende in input 8 valori interi e restituisce in output quanti tra quelli inseriti sono > 100 , quanti sono >50 e ≤ 100 e quanti sono ≤ 50 .

Il programma dovrà utilizzare soltanto 4 variabili.