

5 - Istruzioni condizionali

Programmazione e analisi di dati
Modulo A: Programmazione in Java

Paolo Milazzo

Dipartimento di Informatica, Università di Pisa
<http://pages.di.unipi.it/milazzo>
milazzo@di.unipi.it

Corso di Laurea Magistrale in Informatica Umanistica
A.A. 2019/2020

Sommario

- 1 Istruzioni condizionali
- 2 Espressioni booleane
- 3 Il comando Switch
- 4 Errori tipici

Istruzioni condizionali (1)

In un programma spesso c'è bisogno di scegliere tra operazioni diverse

- Per questo esiste l'**istruzione condizionale** if-else

Un esempio:

```
import java.util.Scanner;

public class Saldo {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        double saldo;

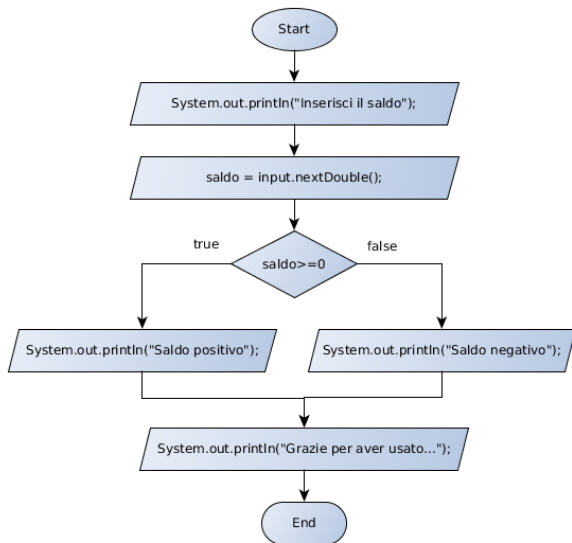
        // chiedo il saldo del conto corrente
        System.out.println("Inserisci il saldo del tuo conto");
        saldo = input.nextDouble();

        if (saldo >= 0) // se il saldo è positivo
            System.out.println("Il saldo è positivo!");
        else // altrimenti
            System.out.println("Il saldo è negativo!");

        System.out.println("Grazie per aver usato il nostro servizio");

    }
}
```

Istruzioni condizionali (2)



Istruzioni condizionali (3)

Un'istruzione condizionale if-else ha questa **forma**:

```
if (...condizione...) ...comando... else ...comando...
```

dove:

- La condizione è detta anche **guardia** dell'if
- La guardia può essere vera (**true**) o falsa (**false**)
- I due comandi alternativi sono detti anche **rami**
- Il ramo else è **opzionale** (vedremo)

Esempi di guardie "semplici": le **espressioni di confronto**

$x==10$ (la variabile x è uguale a 10) – **ATTENZ. ==, non =**

$x!=10$ (la variabile x è diversa da 10)

$x<y$ (la variabile x è minore della variabile y)

$(x/2)<=10$ (l'espressione $x/2$ è minore o uguale a 10)

$x>(10/y)$ (la variabile x è maggiore di $10/y$)

$(x/2)>=(2+y)$ (l'espressione $x/2$ è maggiore o uguale all'espress. $2+y$)

Istruzioni condizionali (4)

Se in un ramo c'è più di una istruzione ci vuole un **blocco**

- Sequenza di comandi racchiusi tra parentesi graffe {...}

```
if (saldo >= 0) {  
  
    System.out.println("Saldo positivo");  
  
    double interesseAttivo = saldo * tassoAttivo;  
    saldo = saldo + interesseAttivo;  
  
} else {  
  
    System.out.println("Saldo negativo");  
  
    double interessePassivo = saldo * tassoPassivo;  
    saldo = saldo - interessePassivo;  
  
}  
  
System.out.print("Nuovo saldo: ");  
System.out.println(saldo);
```

Istruzioni condizionali (5)

I comandi if-else possono essere annidati

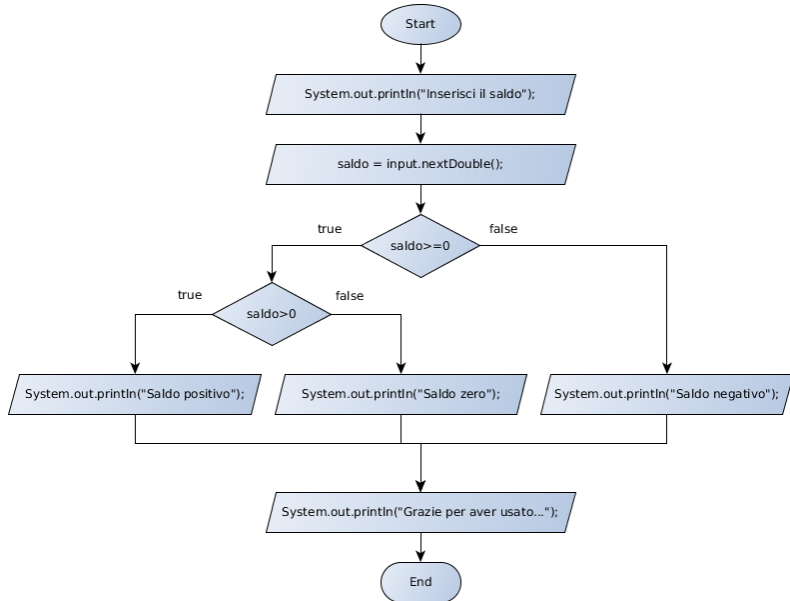
- Si può inserire un if-else in un ramo di un altro if-else

```
if (saldo >= 0)
  if (saldo > 0)
    System.out.println("Saldo positivo!");
  else
    System.out.println("Saldo zero!");
else
  System.out.println("Saldo negativo!");
```

Note:

- non servono le parentesi perchè if-else è un unico comando
- ATTENZIONE: il primo else fa riferimento al secondo if
- L'uso attento di spazi e "a capo" nel programma (**indentazione**) semplifica la lettura e la comprensione del codice

Istruzioni condizionali (6)



Istruzioni condizionali (7)

Un caso particolare di if-else annidati sono gli if-else **concatenati**

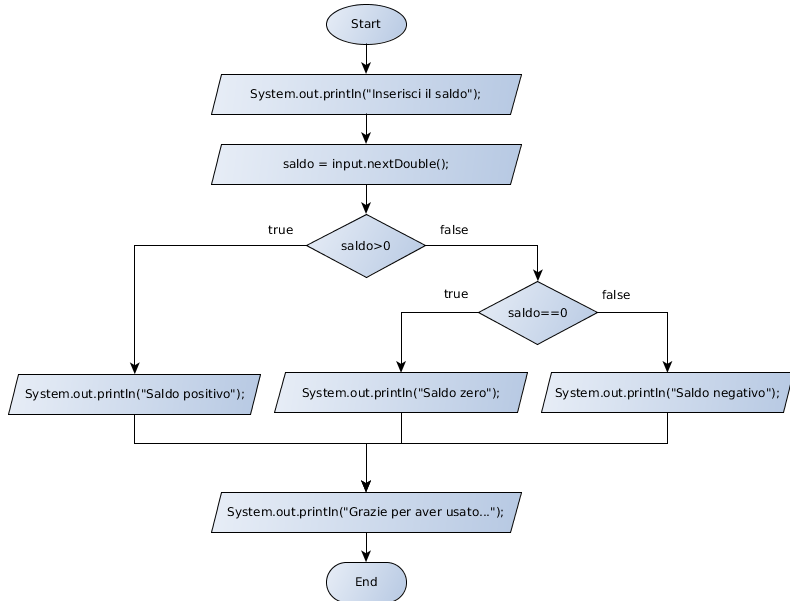
- il ramo else è costituito da un altro if
- la guardia del secondo if considera un **caso alternativo** a quello considerato dalla guardia del primo if

```
if (saldo>0)
    System.out.println("Saldo positivo!");
else if (saldo==0)
    System.out.println("Saldo zero!");
else
    System.out.println("Saldo negativo!");
```

Notare l'indentazione:

- Di solito si scrive else if nella stessa riga

Istruzioni condizionali (8)



Istruzioni condizionali (9)

Il ramo else è **opzionale**

```
if (saldo >= 0)
    System.out.println("Saldo positivo!");
System.out.println("Grazie per aver usato il nostro servizio");
```

Ma quando si annida un if in un if-else bisogna fare **attenzione**

- possono servire le parentesi graffe...

```
if (saldo >= 0) {
    if (saldo > 0)
        saldo = saldo + interesse;
}
else
    saldo = saldo - penale;
```

In quale caso la penale viene applicata se togliamo le parentesi graffe?

Esempio d'uso

Per controllare valori ricevuti in input dall'utente

```
import java.util.Scanner;

public class Divisione {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int num1,num2;

        System.out.println("Inserisci due numeri interi:");
        num1=input.nextInt();
        num2=input.nextInt();

        // controlla che num2 sia diverso da zero
        // prima di calcolare num1/num2
        if (num2==0) {
            System.out.println("ERRORE: divisione per zero");
        } else {
            System.out.print("Quoziente: ");
            System.out.println(num1/num2);
            System.out.print("Resto: ");
            System.out.println(num1%num2);
        }
    }
}
```

Sommario

- 1 Istruzioni condizionali
- 2 Espressioni booleane**
- 3 Il comando Switch
- 4 Errori tipici

Guardie complesse (1)

Abbiamo visto che possiamo esprimere la guardia di un comando `if` tramite delle semplici espressioni di confronto:

- `saldo>0`, `x==y`, ecc...

E se dobbiamo esprimere condizioni più complesse?

Esempio:

- Scrivere un programma che chiede all'utente di inserire due numeri interi (ore e minuti) e stampa "E' un orario" se le due cifre rappresentano un orario corretto, oppure stampa "Non e' un orario" altrimenti.
- Input: 15 27 Output: E' un orario
- Input: 37 97 Output: Non e' un orario

Guardie complesse (2)

Primo modo: if concatenati

```
import java.util.Scanner;

public class Orario {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int ore, minuti;

        System.out.println("Inserire ore e minuti:");
        ore = input.nextInt();
        minuti = input.nextInt();

        if (ore < 0)
            System.out.println("Non e' un orario");
        else if (ore > 23)
            System.out.println("Non e' un orario");
        else if (minuti < 0)
            System.out.println("Non e' un orario");
        else if (minuti > 59)
            System.out.println("Non e' un orario");
        else
            System.out.println("E' un orario");

    }
}
```

Guardie complesse (3)

Secondo modo: **operatore logico || (OR)**

- Combina condizioni delle quali **almeno una** deve essere vera

```
if ( (ore<0) || (ore>23) || (minuti<0) || (minuti>59) )
    System.out.println("Non e' un orario");
else
    System.out.println("E' un orario");
```

Terzo modo: **operatore logico && (AND)**

- Combina condizioni che devono essere **tutte** vere

```
if ( (ore>=0) && (ore<24) && (minuti>=0) && (minuti<60) )
    System.out.println("E' un orario");
else
    System.out.println("Non e' un orario");
```


Guardie complesse (4)

Quarto modo: variabili di tipo **boolean**

- Memorizzano il risultato di una **espressione booleana**

```
boolean ore_ok = (ore>=0) && (ore<24);
boolean minuti_ok = (minuti>=0) && (minuti<60);
if ( ore_ok && minuti_ok )
    System.out.println("E' un orario");
else
    System.out.println("Non e' un orario");
```

Quinto modo: **operatore logico ! (NOT)**

- **Nega** l'espressione booleana seguente (trasforma true in false e viceversa)

```
boolean ore_ok = (ore>=0) && (ore<24);
boolean minuti_ok = (minuti>=0) && (minuti<60);
if ( !ore_ok || !minuti_ok )
    System.out.println("Non e' un orario");
else
    System.out.println("E' un orario");
```

Espressioni booleane

Riassumendo, un'espressione booleana può consistere di:

- Letterali:
 - ▶ true
 - ▶ false
- Variabili booleane (tipo boolean):
 - ▶ isOk, trovato, x,....
- Confronti:
 - ▶ `x==0`
 - ▶ `num1<num2`
 - ▶ `(x%2)==0`
 - ▶ `((lordo-netto)/lordo)<(sconto_max/lordo)`
 - ▶
- Operazioni logiche (dove *eb1* e *eb2* sono espressioni booleane):
 - ▶ `eb1 || eb2`
 - ▶ `eb1 && eb2`
 - ▶ `!eb1`

Tavole di verità

Il risultato delle operazioni logiche è riassunto dalle seguenti **tavole di verità**

| A B | A true | A false |
|---------|--------|---------|
| B true | true | true |
| B false | true | false |

| A&&B | A true | A false |
|---------|--------|---------|
| B true | true | false |
| B false | false | false |

| A | !A |
|-------|-------|
| true | false |
| false | true |

Esempi

Assumendo

```
int i=10;  
boolean b1=true;  
boolean b2=i<5;
```

quali delle seguenti sono espressioni booleane sintatticamente corrette?

```
5  
false  
i==7  
b2||(i>7)  
(b2==true)||(i>7)  
(3+2)&&(7-4)  
!b1  
(b1&&b2)||((i+2)>11)
```

Tra quelle corrette, quali sono vere?

Esempio d'uso (1)

Un programma che controlla temperatura corporea e battito cardiaco

```
import java.util.Scanner;

public class Salute {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Inserisci il la temperatura corporea");
        double febbre = input.nextDouble();

        System.out.println("Inserisci i battiti cardiaci al minuto");
        int battito = input.nextInt();

        boolean febbreNo, febbreBassa, febbreAlta;
        boolean battitoOK, battitoAccelerato;

        febbreNo = febbre < 37.0;
        febbreBassa = (febbre >= 37) && (febbre < 38);
        febbreAlta = febbre >= 38;

        battitoOK = battito < 80;
        battitoAccelerato = battito >= 80;

        ....continua.....
    }
}
```

Esempio d'uso (2)

```
....continua.....  
  
    if (febbreNo && battitoOK)  
        System.out.println("Tutto OK");  
    else if ((febbreBassa && battitoAccelerato)|| (febbreAlta))  
        System.out.println("Stato febbrile");  
    else  
        System.out.println("Stato febbrile lieve");  
    }  
}
```

In quali casi stampa "Stato febbrile lieve" ?

Sommario

- 1 Istruzioni condizionali
- 2 Espressioni booleane
- 3 Il comando Switch**
- 4 Errori tipici

Quanti if...

In alcuni casi un comando `if` può diventare un po' lungo da scrivere:

Esempio: un programma che legge un numero e lo trasforma in un giorno

```
import java.util.Scanner;

public class Giorno {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Inserisci un numero [1-7]");
        int giorno = input.nextInt();

        if (giorno==1) System.out.println("Lunedì");
        else if (giorno==2) System.out.println("Martedì");
        else if (giorno==3) System.out.println("Mercoledì");
        else if (giorno==4) System.out.println("Giovedì");
        else if (giorno==5) System.out.println("Venerdì");
        else if (giorno==6) System.out.println("Sabato");
        else if (giorno==7) System.out.println("Domenica");
        else System.out.println("Numero errato");

    }
}
```


Il comando "switch" (1)

Un if concatenato che confronta

- il risultato di una **espressione di tipo int o char** (vedremo)
- con un numero di **letterali** alternativi (ad es. 1,2,3,...)

può essere sostituito da un comando switch

```
switch (giorno) {  
  case 1: System.out.println("Lunedì"); break;  
  case 2: System.out.println("Martedì"); break;  
  case 3: System.out.println("Mercoledì"); break;  
  case 4: System.out.println("Giovedì"); break;  
  case 5: System.out.println("Venerdì"); break;  
  case 6: System.out.println("Sabato"); break;  
  case 7: System.out.println("Domenica"); break;  
  default: System.out.println("Numero errato");  
}
```

Il **comando break** fa saltare l'esecuzione alla parentesi graffa chiusa

- Evita, ad esempio, che dopo aver eseguito il caso 3 venga eseguito anche il 4
- Tra l'etichetta case N: e il comando break ci può essere più di un comando senza bisogno di parentesi graffe aggiuntive

Il comando “switch” (2)

Il comando switch consente di accorpare casi in maniera abbastanza semplice

```
switch (giorno) {
  case 1: case 2: case 3: case 4: case 5:
    System.out.println("Giorno lavorativo"); break;
  case 6:
    System.out.println("Giorno prefestivo"); break;
  case 7:
    System.out.println("Giorno festivo"); break;
  default:
    System.out.println("Numero errato ");
}
```

Sommario

- 1 Istruzioni condizionali
- 2 Espressioni booleane
- 3 Il comando Switch
- 4 Errori tipici**

Istruzioni condizionali: errori tipici (1)

Vediamo alcuni **errori tipici** nell'uso delle istruzioni condizionali:

- Uso dell'operatore di assegnamento invece che di confronto

```
if (x=0)
    System.out.println("x vale 0");
```

- if con comando vuoto

```
if (x==0);
    System.out.println("x vale 0");
```

- if-else senza blocchi (parentesi graffe)

```
if (x<=y) // manca {
    System.out.print(x);
    System.out.print("e' minore o uguale di");
    System.out.println(y);
// manca }
else // manca {
    System.out.print(x);
    System.out.print("e' maggiore di");
    System.out.println(y);
// manca }
```

Istruzioni condizionali: errori tipici (2)

- omissione di else

```
if (y==0)
    System.out.println("Divisione per zero non consentita");
// manca else
    System.out.println(x/y);
```

- omissione di break

```
switch (giorno) {
    case 1: case 2: case 3: case 4: case 5:
        System.out.println("Giorno lavorativo"); // manca break;
    case 6:
        System.out.println("Giorno prefestivo"); // manca break;
    case 7:
        System.out.println("Giorno festivo"); // manca break;
    default:
        System.out.println("Numero errato");
}
```