Le istruzioni condizionali

Docente
Mario Perna
prof.perna.mario@darzo.net

A.S.
2025/2026
Materia
Informatica

Introduzione

In Java, come in qualsiasi linguaggio C-like è possibile definire una biforcazione del codice grazie al costrutto if-elsecome visto gli anni passati.

if(<condizione>)
<istruzione se la condizione è vera>
else
<istruzione se la condizione è falsa>

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int eta;

    System.out.println("Inserire la tua eta': ");
    eta = scanner.nextInt();

    if (eta >= 18)
        System.out.println("Sei maggiorenne");
    else
        System.out.println("Sei minorenne");

}
```

Più istruzioni in un ramo

Inoltre, è possibile aggiungere più istruzioni in un ramo di verità circondando le istruzioni dalle parentesi graffe {}

```
if(<condizione>){
        <istruzione 1 se la condizione è vera>
        …
        <istruzione n se la condizione è vera>
}
else{
        <istruzione 1 se la condizione è falsa>
        …
        <istruzione m se la condizione è falsa>
}
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int eta;

    System.out.println("Inserire la tua eta': ");
    eta = scanner.nextInt();

    if (eta >= 18){
        System.out.println("Sei maggiorenne");
        System.out.println("Puoi andare a votare!");
    }
    else{
        System.out.println("Sei minorenne");
        System.out.print("Devi attendere ancora " + (18 - eta));
        System.out.println(" anni per diventare maggiorenne");
    }

}
```

Istruzioni di scelta nidificate

Possiamo generare flussi di esecuzione complessi a piacere grazie alla possibilità di inserire all'interno di rami di verità ulteriori scelte condizionali.

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int eta;

    System.out.println("Inserire la tua eta': ");
    eta = scanner.nextInt();

    if (eta >= 18){
        System.out.println("Sei maggiorenne");
        System.out.println("Hai la patente? ");
        if (scanner.next().charAt(0) == 's')
            System.out.println("Puoi guidare un'auto!");
        else
            System.out.println("Non puoi guidare un'auto");
    }
    else{
        System.out.println("Sei minorenne");
        System.out.println("Non puoi guidare un'auto");
    }

}
```

Le condizioni

Possiamo costruire condizioni anche molto complicate, ma per ora ci limitiamo a vedere i casi più semplici e basilari:

- a == b : il valore contenuto in a è uguale al valore contenuto in b (in matematica sarebbe equivalente a dire a = b)
- a > b : il valore contenuto in a è maggiore al valore contenuto in b
- a < b : il valore contenuto in a è minore al valore contenuto in b
- a >= b : il valore contenuto in a è maggiore o uguale al valore contenuto in b
- a <= b : il valore contenuto in a è minore o uguale al valore contenuto in b
- a != b : il valore contenuto in a è diverso al valore contenuto in b (in matematica sarebbe equivalente a dire a ≠ b)

Gli operatori booleani

I principali operatori di tipo booleano, ovvero gli operatori che consentono di eseguire operazioni su elementi di tipo bool sono 3:

Funzione	Operatore	Significato
AND	&&	Congiunzione logica
OR	||	Disgiunzione logica
NOT	!	Negazione logica

L'operatore booleano: AND

L'operatore logico && prevede due condizioni una a sinistra e una a destra e restituisce true se e solo se entrambe le condizioni sono true:

A && B

Vediamo qualche esempio:

```
int n1 = 5;

int n2 = 10;

bool risultato1 = n1 > 3 && n2 <= 10; // true bool risultato2 = n1 < 2

&& n2 <= 10; // false bool risultato3 = n1 < 2 && n2 > 15; // false
```

A	B	A && B
false	false	false
false	true	false
true	false	false
true	true	true

L'operatore booleano: OR

L'operatore logico || prevede due condizioni una a sinistra e una a destra e restituisce truese almeno una delle due condizioni è true:

A | | B

Vediamo qualche esempio:

```
int n1 = 5;

int n2 = 10;

bool risultato1 = n1 > 3 | | n2 <= 10; // true bool risultato2 = n1 < 2 | |

n2 <= 10; // true bool risultato3 = n1 < 2 | | n2 > 15; // false
```

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

L’operatore booleano: NOT

L’operatore logico ! prevede una sola condizione a destra e restituisce truese la condizione è falsee viceversa:

!A

Vediamo qualche esempio:

```
int n1 = 5;

int n2 = 10;

bool risultato1 = !(n1 > 3); // false

bool risultato2 = !(n2 > 15); // true
```

A	!A
false	true
true	false

Combinare gli operatori booleani

Grazie all'utilizzo di diverse condizioni collegate con i vari operatori booleani possiamo esprimere qualsiasi condizione complessa ci venga in mente, vediamo vari esempi più articolati:

```
// Controllo che un numero sia pari e positivo
bool pari_positivo = num % 2 == 0 && num >= 0;
// Un numero compreso tra 10 e 20
bool tra_10_20 = num >= 10 && num <= 20;
// Un numero dispari o diverso da 20
bool dispari_o_diverso_15 = num % 2 == 1 || num != 20;
```

Operatore ternario

Anche in Java è possibile sfruttare la sintassi dell'operatore ternario:

<condizione> ? <istruzione se vero> : <istruzione se falso>

```
public static void main(String[] args) {
    int numero;
    Scanner scanner = new Scanner(System.in);
    System.out.print("Inserire un numero: ");
    numero = scanner.nextInt();

    System.out.println(numero % 2 == 0 ? "Pari" : "Dispari");

    System.out.println(numero % 2 == 0 ?
            (numero > 0 ? "Positivo pari" : "Positivo dispari") :
            (numero > 0 ? "Negativo dispari" : "Negativo dispari"));
}
```

L'istruzione switchper scelte multiple

Analizziamo il seguente codice che cerca di confrontare una variabile con diversi valori:

```
if (numero == 1)
    System.out.println("Hai inserito il giorno della settimana: Lunedi");
else if (numero == 2)
    System.out.println("Hai inserito il giorno della settimana: Martedi");
else if (numero == 3)
    System.out.println("Hai inserito il giorno della settimana: Mercoledi");
else if (numero == 4)
    System.out.println("Hai inserito il giorno della settimana: Giovedi");
else if (numero == 5)
    System.out.println("Hai inserito il giorno della settimana: Venerdi");
else if (numero == 6)
    System.out.println("Hai inserito il giorno della settimana: Sabato");
else if (numero == 7)
    System.out.println("Hai inserito il giorno della settimana: Domenica");
else
    System.out.println("Hai inserito un giorno della settimana non valido!");
```

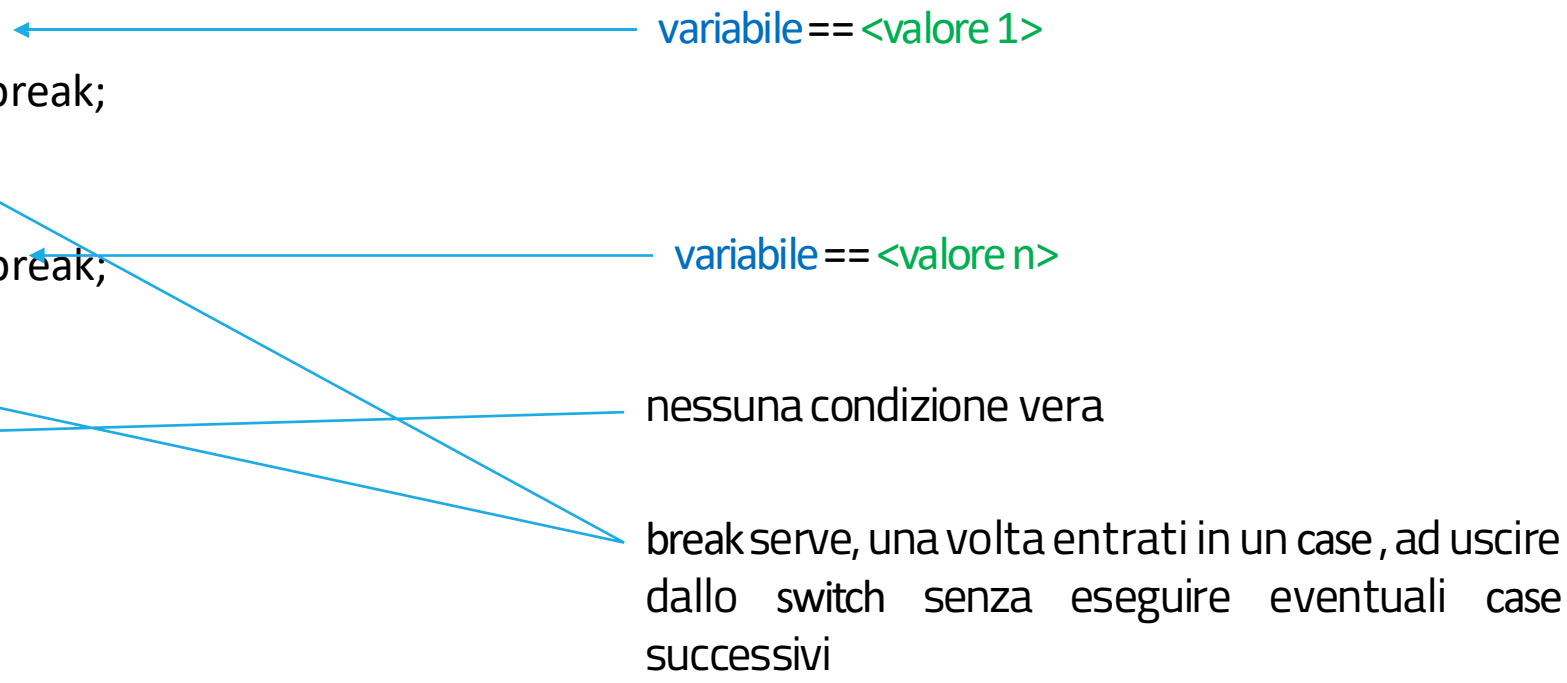
Funziona, ma possiamo fare di meglio!

L'istruzione switchper scelte multiple

Il programma appena visto è molto «verboso» e lungo da scrivere, non possiamo scrivere del codice? La risposta è SI grazie all' istruzione switch

```
switch(variabile){
        case(<valore 1>):
                <istruzioni 1> break;
        ...
        case(<valore n>):
                <istruzioni n> break;
        default:
                <istruzioni>

}
```



L'istruzione switchper scelte multiple

```
switch(numero){
    case(1):
        System.out.println("Hai inserito il giorno della settimana: Lunedi");
        break;
    case(2):
        System.out.println("Hai inserito il giorno della settimana: Martedi");
        break;
    case(3):
        System.out.println("Hai inserito il giorno della settimana: Mercoledi");
        break;
    case(4):
        System.out.println("Hai inserito il giorno della settimana: Giovedi");
        break;
    case(5):
        System.out.println("Hai inserito il giorno della settimana: Venerdi");
        break;
    case(6):
        System.out.println("Hai inserito il giorno della settimana: Sabato");
        break;
    case(7):
        System.out.println("Hai inserito il giorno della settimana: Domenica");
        break;
    default:
        System.out.println("Hai inserito un giorno della settimana non valido!");

}
```

Informazioni sull’utilizzo dei materiali didattici

Queste slides si basano su materiali originariamente elaborati dal **Prof. Andrea Melioli**, opportunamente modificati e integrati secondo specifiche esigenze riguardanti la programmazione disciplinare.

L’autore autorizza al **prof. Mario Perna** l’utilizzo, la modifica, la pubblicazione e qualsiasi altra forma di operazione sui materiali a scopo didattico e formativo.

L’uso o la diffusione di questi materiali è **vietato** senza il preventivo contatto con il proprietario del documento, Prof. Mario Perna (prof.mario.perna@darzo.net).