Condizioni: Switch

La parola chiave default specifica del codice da eseguire se non c'è corrispondenza tra maiuscole e minuscole:

```
1. int day = 4;
2. switch (day) {
3.   case 6:
4.       System.out.println("Today is Saturday");
5.       break;
6.   case 7:
7.       System.out.println("Today is Sunday");
8.       break;
9.   default:
10.    System.out.println("Looking forward to the Weekend");
11.}     // Outputs "Looking forward to the Weekend"
```

Si noti che se l'istruzione default viene utilizzata come ultima istruzione in un blocco switch, non necessita di un'interruzione.

Creare un sistema di inserimento che prenda tre dati in fila, un numero, una string e che setti un bool

Dopo di che un menu dovrà farci scegliere tra tre opzioni

Funzioni matematiche, Funzioni Stringa, Casting

Funzioni matematiche: Sul numero che abbiamo eseguiremo un operazione a scelta fra 4 disponibili e vedremo il risultato
Funzioni String: Devono eseguire un sub string o un concatenamento a scelta
Casting trasforma un dato da un tipo ad un 'altro e valorizza il risultato a schermo

itconsulting
Consulenza & Formazione

L'iterazione ovvero la base dei cicli

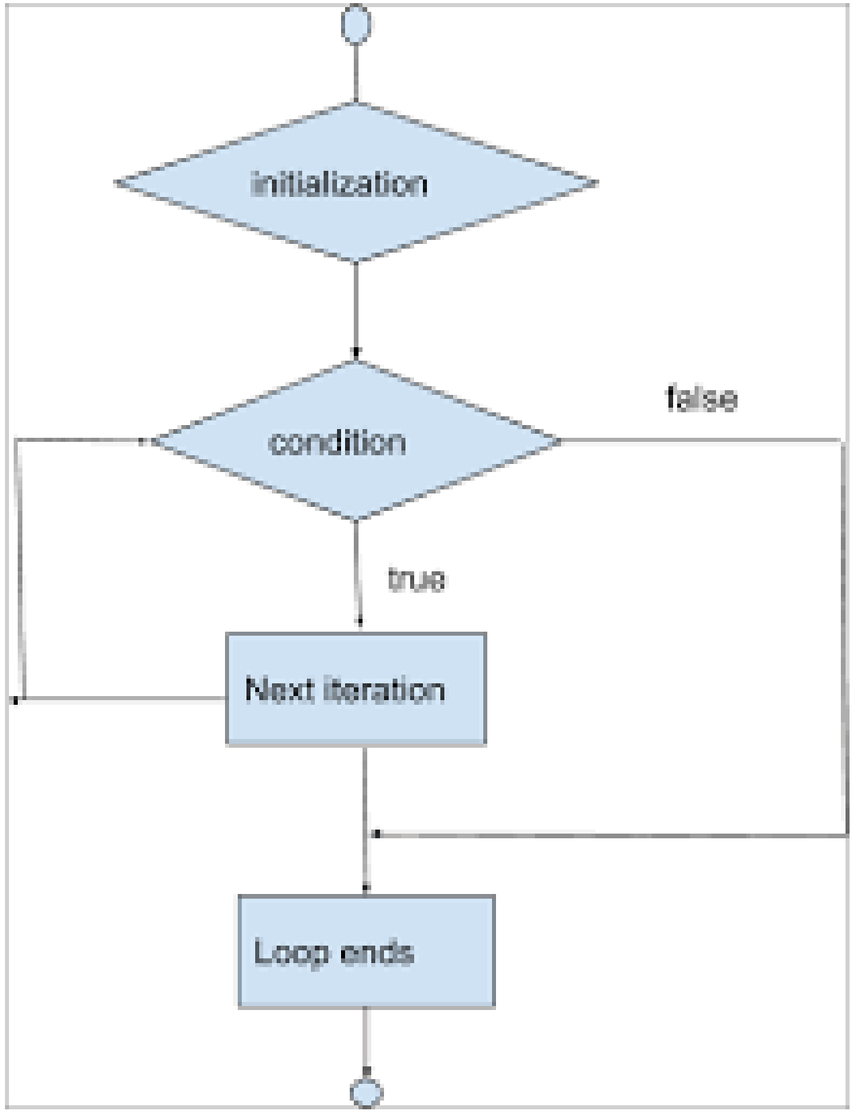
L'iterazione è l'atto di ripetere un processo con l'obiettivo di avvicinarsi a un risultato desiderato.

Ogni ripetizione del processo è essa stessa definita un'iterazione, e i risultati di una sono utilizzati come punto di partenza per quella successiva.

Diffuso è l'utilizzo negli algoritmi e nella programmazione in ambito informatico, ma anche in campi come il project management.

itconsulting
Consulenza & Formazione

L'iterazione ovvero la base dei cicli



itconsulting
Consulenza & Formazione

la base dei cicli in Java

I loop o cicli possono eseguire un blocco di codice fintanto che viene raggiunta una condizione specificata. I loop sono utili perché fanno risparmiare tempo, riducono gli errori e rendono il codice più leggibile.

Il ciclo while scorre in maniera ripetitiva e sequenziale attraverso un blocco di codice fintanto che una condizione specificata è true :

Sintassi

```
1. while (condition) {
2.   // code block to be executed
3. }
```

itconsulting
Consulenza & Formazione

la base dei cicli in Java

Nell'esempio seguente, il codice nel ciclo verrà eseguito, più e più volte, finché una variabile (i) è minore di 5:

```
1. int i = 0;
2. while (i < 5) {
3. System.out.println(i);
4.   i++;
5. }
```

Nota: non dimenticare di aumentare la variabile utilizzata nella condizione, altrimenti il ciclo non finirà mai!

itconsulting
Consulenza & Formazione

Il ciclo Do/While

Il ciclo do/while è una variante del while.

Questo ciclo eseguirà il blocco di codice una volta, prima di verificare se la condizione è vera, quindi ripeterà il ciclo finché la condizione è vera.

Sintassi

```
1. do {
2.   // code block to be executed
3. }
4. while (condition);
```

L'esempio seguente utilizza un ciclo do/while.

Il ciclo verrà sempre eseguito almeno una volta, anche se la condizione è falsa, perché il blocco di codice viene eseguito prima che la condizione venga verificata:

```
1. int i = 0;
2. do {
3.   System.out.println(i);
4.   i++;
5. }
6. while (i < 5);
```

itconsulting
Consulenza & Formazione

Test il ristorante

Andiamo a creare un primo menù che ci fa entrare o chiudere subito ogni nuovo ingresso randomizzeremo un budget per il cliente e terremo il conto dei clienti e fine app stamperemo totale speso e nr clienti
Andiamo a creare un menu con 4 scelte Compra, vedi, aggiungi, Esci

Compra farà vedere una lista di piatti che puoi comprare stampando , nome, prezzo (int) e ingredienti (minimo 2) e darà la possibilità di comprarlo se il nostro budget e sufficiente la disp. qui non si vedrà

Vedi: stamperà la lista piatti e le loro disponibilità una volta finito un piatto dovrà non essere più ordinabile

Aggiungi: Se si conosce una password, si avranno due scelte, resetta disponibilità che mette tutte le disponibilità come all'Inizio e aggiungi piatto

Esci, ci riporta al primo menu e ci permetta di riiniziare o terminare

PAUSA FINO A E 15

itconsulting
Consulenza & Formazione

Il ciclo For

Quando sai esattamente quante volte vuoi eseguire il looping di un blocco di codice, usa il for invece di un loop while:

Sintassi

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

L' istruzione 1 viene eseguita (una volta) prima dell'esecuzione del blocco di codice.

L' istruzione 2 definisce la condizione per l'esecuzione del blocco di codice.

L' istruzione 3 viene eseguita (ogni volta) dopo che il blocco di codice è stato eseguito.

L'esempio seguente stamperà i numeri da 0 a 4:

```
for (int i = 0; i < 5; i++) {
  System.out.println(i);
}
```

itconsulting
Consulenza & Formazione

Loop nidificati

È anche possibile inserire un loop all'interno di un altro loop.

Questo è chiamato un ciclo annidato . Il "ciclo interno" verrà eseguito una volta per ogni iterazione del "ciclo esterno":

```
1. // Outer loop
2. for (int i = 1; i <= 2; i++) {
3.   System.out.println("Outer: " + i); // Executes 2 times
4.
5.   // Inner loop
6.   for (int j = 1; j <= 3; j++) {
7.     System.out.println(" Inner: " + j); // Executes 6 times (2 * 3)
8.   }
9. }
```

Loop nidificati

PAUSA FINO A E 05

itconsulting
Consulenza & Formazione

for-each

C'è anche un ciclo " for-each ", che viene utilizzato esclusivamente per scorrere gli elementi in un array :

Sintassi

```
for (type variableName : arrayName) {
  // code block to be executed
}
```

L'esempio seguente emette tutti gli elementi nell'array cars che spiegheremo più avanti , utilizzando un ciclo " for-each ":

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
  System.out.println(i);
}
```

itconsulting
Consulenza & Formazione

for-each

C'è anche un ciclo " for-each ", che viene utilizzato esclusivamente per scorrere gli elementi in un array :

Sintassi

```
for (type variableName : arrayName) {
  // code block to be executed
}
```

L'esempio seguente emette tutti gli elementi nell'array cars che spiegheremo più avanti , utilizzando un ciclo " for-each ":

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
  System.out.println(i);
}
```

itconsulting
Consulenza & Formazione

Istruzione di rottura

Hai già visto l' break istruzione usata in un capitolo precedente .
Era usato per "saltare fuori" da una dichiarazioneswitch.

L'istruzione break può essere utilizzata anche per uscire da un ciclo .

Questo esempio interrompe il ciclo quando i è uguale a 4:

```
1. public class Main {
2.   public static void main(String[] args) {
3.     for (int i = 0; i < 10; i++) {
4.       if (i == 4) {
5.         break;
6.       }
7.       System.out.println(i);
8.     }  }}
```

itconsulting
Consulenza & Formazione

Java Continue

L'istruzione continue interrompe un'iterazione (nel ciclo), se si verifica una condizione specificata, continua con l'iterazione successiva nel ciclo.

Questo esempio salta il valore di 4:

```
1. public class Main {
2.   public static void main(String[] args) {
3.     for (int i = 0; i < 10; i++) {
4.       if (i == 4) {
5.         continue;
6.       }
7.       System.out.println(i);
8.     } }}
```