KeyTraining

Curs java pentru testare automata

JAVA

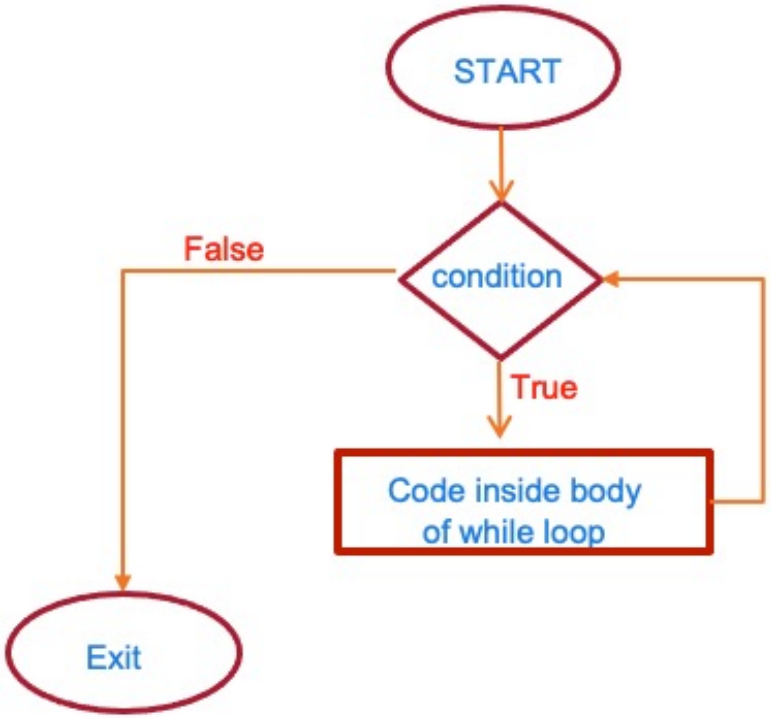
KeyTraining

Cuprins curs:

- While loop
- Do while loop
- Switch

While loop in Java

- Bucla while din Java este cea mai simplă dintre toate structurile de looping. Este o instrucțiune de buclă controlată de intrare.
- În bucla controlată de intrare, starea testului este evaluată mai întâi. Dacă condiția specificată este adevărată, atunci corpul buclei este executat.
- Dacă condiția nu este îndeplinită, atunci corpul buclei nu este executat.
- O buclă while este cea mai fundamentală declarație de buclă din Java. Execută instrucțiuni în mod repetat în timp ce condiția de testare este adevărată.


START
condition
False
True
Code inside body
of while loop
Exit

While loop in Java

```
while (test_condition) {

    //code
}
```

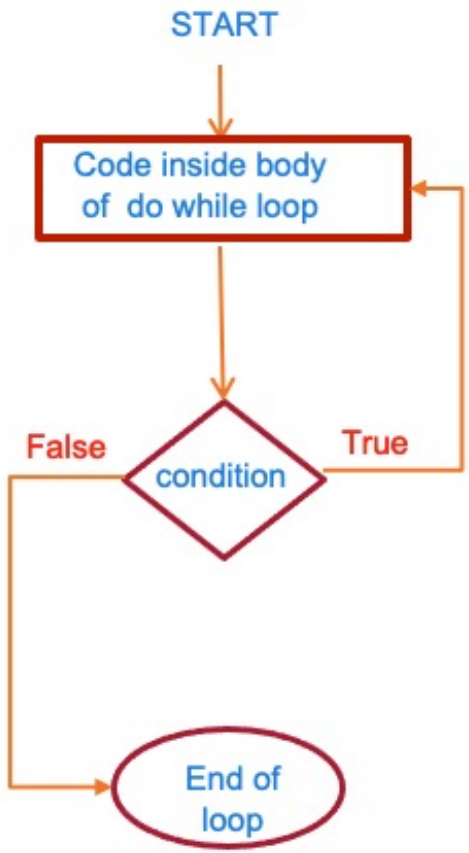
- În sintaxa de mai sus, condiția de test poate fi orice expresie booleană, care este evaluată înainte ca instrucțiunile să fie executate. Această expresie booleană controlează execuția corpului buclei.Dacă condiția nu este îndeplinită, atunci corpul buclei nu este executat.
- Când condiția de testare este adevărată, atunci se execută corpul buclei (instrucțiunile). După executarea corpului buclei, starea testului este din nou evaluată la adevărată.
- Dacă condiția specificată este adevărată, corpul buclei este executat din nou. Acest proces de execuție repetată a corpului buclei continuă până când expresia condițională devine în cele din urmă falsă.

While loop in Java

- Odată ce expresia condițională este falsă, bucla este terminată și controlul execuției este transferat din buclă.
- La ieșire, programul continuă cu următoarea instrucțiune imediat după închiderea acoladelor buclei while. Fiecare execuție a corpului buclei este denumită iterație (sau repetare).
- Partea buclei care conține instrucțiuni care trebuie executate în mod repetat se numește corpul buclei. Un corp de buclă poate conține una sau mai multe instrucțiuni.
- Acoladele sunt necesare numai dacă corpul buclei are două sau mai multe instrucțiuni. Cu toate acestea, este o bună practică de programare să folosiți acolade, chiar dacă corpul buclei conține o singură afirmație.

Do While loop in Java

- O buclă do-while în Java este o formă a buclei while. Este la fel ca o buclă while, cu excepția faptului că execută mai întâi corpul buclei și apoi evaluează condiția de continuare a buclei.
- În instrucțiunea while, starea testului este evaluată mai întâi înainte ca corpul buclei să fie executat. Dacă condiția este inițial falsă, corpul buclei nu va fi deloc executat. Cu toate acestea, uneori, ar putea fi necesar să se execute corpul unei bucle cel puțin o dată înainte de efectuarea testului. Astfel de situații pot fi gestionate cu ajutorul buclei do-while în Java.
- Bucla do-while își execută întotdeauna corpul cel puțin o dată, înainte ca testul să fie evaluat, deoarece expresia condiționată de test se află în partea de jos a buclei.


START
Code inside body
of do while loop
False
condition
True
End of
loop

Do While loop in Java

- Din flowchart-ul do-while de mai sus, este clar că mai întâi corpul buclei este executat, iar apoi expresia condițională buclă este evaluată pentru a determina dacă se continuă sau se termină bucla.
- Dacă evaluarea este adevărată, corpul buclei este executat din nou. Acest proces continuă atât timp cat condiția de testare este adevărată.
- Când condiția specificată este falsă, bucla este terminată, controlul execuției iese din bucla do-while și trece la următoarea instrucțiune care apare imediat după instrucțiunea while.

Do While loop in Java

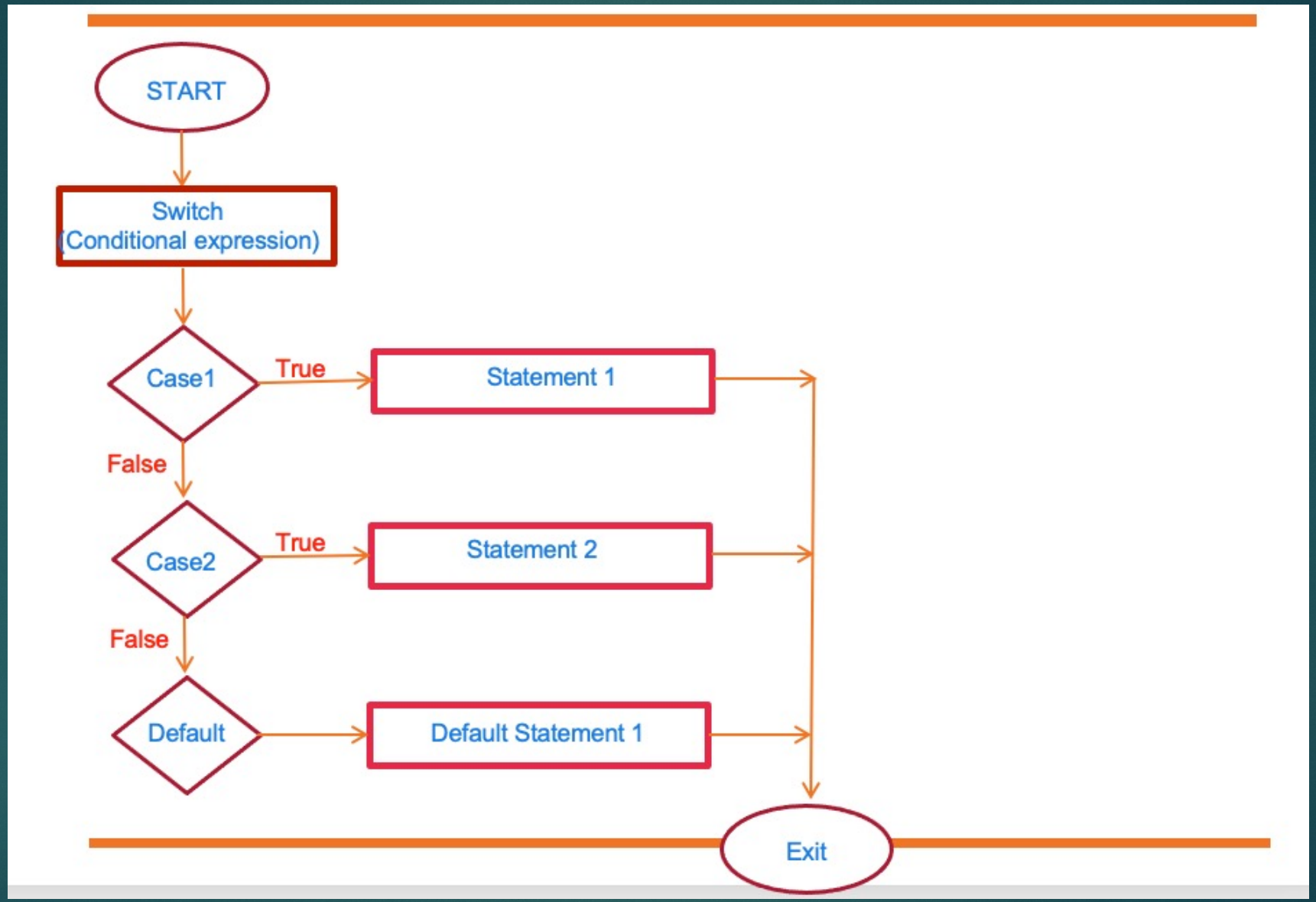
```
do {
    // Loop body;
    //Increment/decrement;

} while (test_condition expression );
```

- Deoarece condiția specificată este testată în partea de jos a buclei, bucla do ... while oferă o buclă controlată la ieșire.
- Condiția de testare trebuie să fie expresie booleană.
- Nu este nevoie să folosim acolade dacă există o singură afirmație în bucla do ... While dar este best practice sa folosim ☺.

Switch

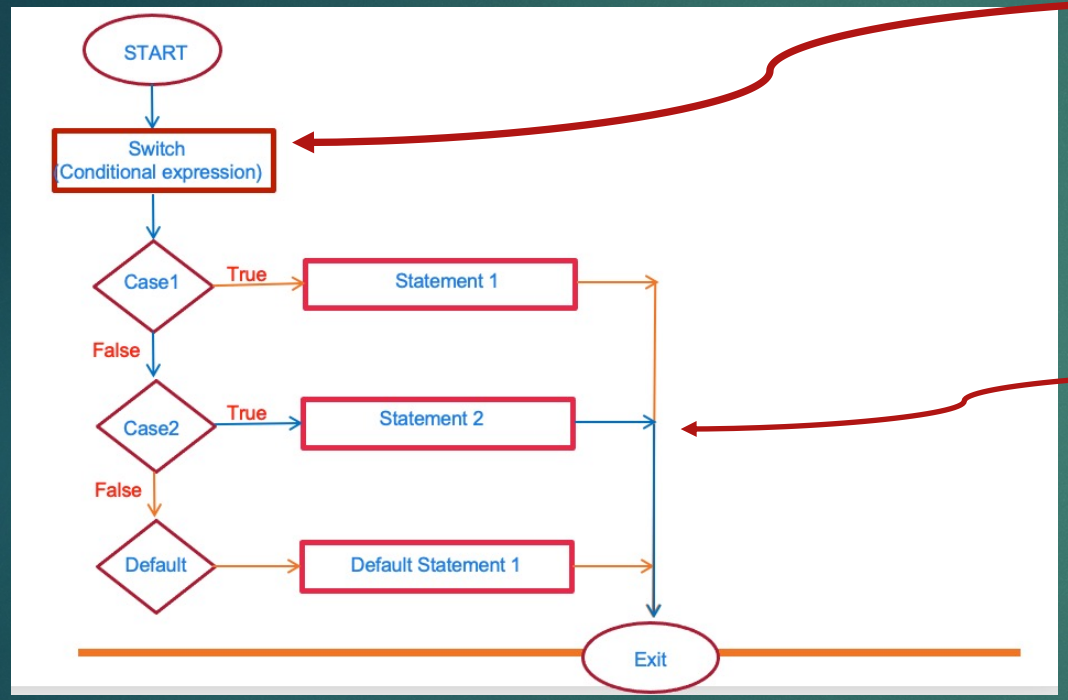
- O instrucțiune switch în Java este o instrucțiune de control condițional (sau instrucțiune de decizie multiway) care execută o instrucțiune din mai multe condiții.
- Folosește rezultatul unei expresii pentru a evalua ce instrucțiuni să execute. Este o alternativă la declarația if-else-if.
- Cu alte cuvinte, o instrucțiune switch execută instrucțiuni bazate pe valoarea unei variabile sau a unei expresii, în raport cu o listă de valori de caz.
- Dacă se găsește o potrivire, se execută un bloc de instrucțiuni corespunzător cazului respectiv.


START
Switch
(Conditional expression)
Case1
True
Statement 1
False
Case2
True
Statement 2
False
Default
Default Statement 1
Exit

```
var nota = "B";

switch(nota){

case "A":
 System.out.println("Great job!");
    break;
case "B":
    System.out.println("Good job!");
    break;
case "C":
    System.out.println("Not that good !");
    break;
case "D":
    System.out.println("Kind of bad!");
    break;
default:
    System.out.println("Te rugam introdu o nota v
    break;
}
```




```java
var nota = "D";

switch(nota){

case "A":
 System.out.println("Great job!");
    break;
case "B":
    System.out.println("Good job!");
    break;
case "C":
    System.out.println("Not that good !");
    break;
case "D":
    System.out.println("Kind of bad!");
    break;
default:
    System.out.println("Te rugam introdu o nota valida!");
    break;
}
```

Switch

- Când se execută instrucțiunea switch, valoarea expresiei este comparată succesiv cu fiecare valoare de caz, cum ar fi case-1, case-2-,. . .default

- Dacă se găsește o potrivire, se execută secvența de instrucțiuni care se regăsesc in instrucțiunile expresiei.
- Dacă valoarea expresiei nu este egală cu case-1, case-2,. . . . atunci niciuna dintre secvențele de instrucțiuni nu va fi executată.
- În acest caz, se execută clauza default, iar apoi sunt executate instrucțiunile implicite.

Switch

- Dacă nu se potrivește niciun caz și nu este prezentă nicio valoare default, atunci nu mai are loc nicio execuție.
  - Instrucțiunea break este utilizată în interiorul blocului de cod pentru a termina o secvență de instrucțiuni. Este opțional.
- În acest caz, se execută clauza default, iar apoi sunt executate instrucțiunile implicite.

Cand folosim Switch?

- O instrucțiune switch poate fi utilizată pentru a selecta un bloc din mai multe blocuri de instrucțiuni.
- Adică, poate fi folosit pentru a executa o instrucțiune din condiții multiple. In general poate fi o alegere mai bună decât if-else-if.
- Când trebuie să executați o instrucțiune din două alternative, utilizați instrucțiunea if-else.

instrucțiuni Neconditionate

- Când fluxul de execuție sare la o altă parte a codului fără a efectua niciun test condiționat, se numește instrucțiuni necondiționate sau execuție necondiționată în Java.
- Următoarele declarații necondiționate sunt disponibile în Java:
- break statement
- continue statement
- return statement

instrucțiuni Neconditionate

- Aceste declarații transferă controlul execuției către o altă parte a programului.
- De exemplu, să presupunem că există o listă de 100 de nume și că trebuie să căutăm un anumit nume din listă.
- Când căutarea este finalizată și numele dorit este găsit, bucla de program scrisă pentru căutare trebuie să fie încheiată.
- În acest scop, Java permite un salt de la o instrucțiune la sfârșitul sau începutul unei bucle, precum și ieșirea dintr-o buclă.

Break statement

- O instrucțiune break în Java este utilizată pentru a sparge o instrucțiune de loop sau switch.
- Când o instrucțiune break este executată în interiorul unei instrucțiuni loop sau switch, bucla este terminată imediat într-o condiție specificată și controlul programului continuă următoarea instrucțiune imediat după buclă.
- Când buclele sunt nested, instrucțiunea break rupe doar bucla interioară.
- Instrucțiunea Java break poate fi utilizată în toate tipurile de bucle(loop), cum ar fi for loop, while loop și do-while loop.
- Sintaxa generală pentru instrucțiunea break este următoarea: break;

Continue statement

- Instrucțiunea Continue în Java este o altă afirmație similară, cum ar fi instrucțiunea break care este utilizată în interiorul unei bucle pentru a repeta următoarea iterație a buclei.
- Când este întâlnită instrucțiunea continue, instrucțiunile ulterioare din buclă sunt omise (adică nu sunt executate) într-o condiție specificată, iar controlul execuției continuă cu următoarea repetare a buclei.
- În cuvinte simple, este folosit pentru a continua bucla. Sintaxa generală a declarației continue în Java este după cum urmează: continue;

Continue statement

- Instrucțiunea Java continue este o altă instrucțiune de salt utilizată în structura de control a buclei, care este utilizată atunci când trebuie să trecem imediat la următoarea iterație a buclei. Această declarație poate fi utilizată în toate tipurile de bucle, cum ar fi for loop, while loop și do-while loop.