Corso di base JAVA

Mauro Donadeo
mail: mauro.donadeo@gmail.com

Iterazioni

HT Lab

Complementi

Enunciato switch

Una sequenza di confronti in un'**unica variabile intera** con diverse **alternative costanti** può essere realizzata con l'enunciato switch

```
1  int x;
2  int y;
3  ...
4  if (x == 1)
5      y = 1;
6  else if (x == 2)
7      y = 4;
8  else if (x == 4)
9      y = 16;
10 else
11     y = 0;
```

```
1  int x;
2  int y;
3  ...
4  switch (x){
5      case 1: y = 1; break;
6      case 2: y = 4; break;
7      case 4: y = 16; break;
8      default: y = 0; break;
9  }
```

L'enunciato switch

- **Vantaggio:** non bisogna ripetere il nome della variabile;
- **Svantaggio:**
  - non si può usare se la variabile da confrontare non è intera
  - Non si può usare se uno dei valori da confrontare non è costante
  - in ogni `case` deve terminare con un enunciato `break`, altrimenti viene eseguito anche il corpo del `case` successivo. *Fonte di molti errori*

Errori con gli operatori relazionali

Alcune espressioni "naturali" con operatori relazionali sono errate, ma per fortuna il compilatore le rifiuta

- **if(0 <= x <= 1) NON FUNZIONA.**
- **if(0 <= x && x <= 1) OK**
- **if(x && y > 0) NON FUNZIONA**
- **if(x > 0 && y > 0) OK**

Problema

Riprendiamo un problema visto nella prima lezione, per il quale abbiamo individuato un algoritmo senza realizzarlo:

- **Problema:** Avendo depositato ventimila euro in un conto bancario che produce il 5% di interessi all'anno, capitalizzati annualmente, quanti anni occorrono affinché il saldo del conto arrivi al doppio della cifra iniziale?
- Abbiamo bisogno di un programma che ripeta degli enunciati (capitalizzazione degli interessi annuali, incremento del conto degli anni), finché non si realizza la condizione desiderata

Enunciato while

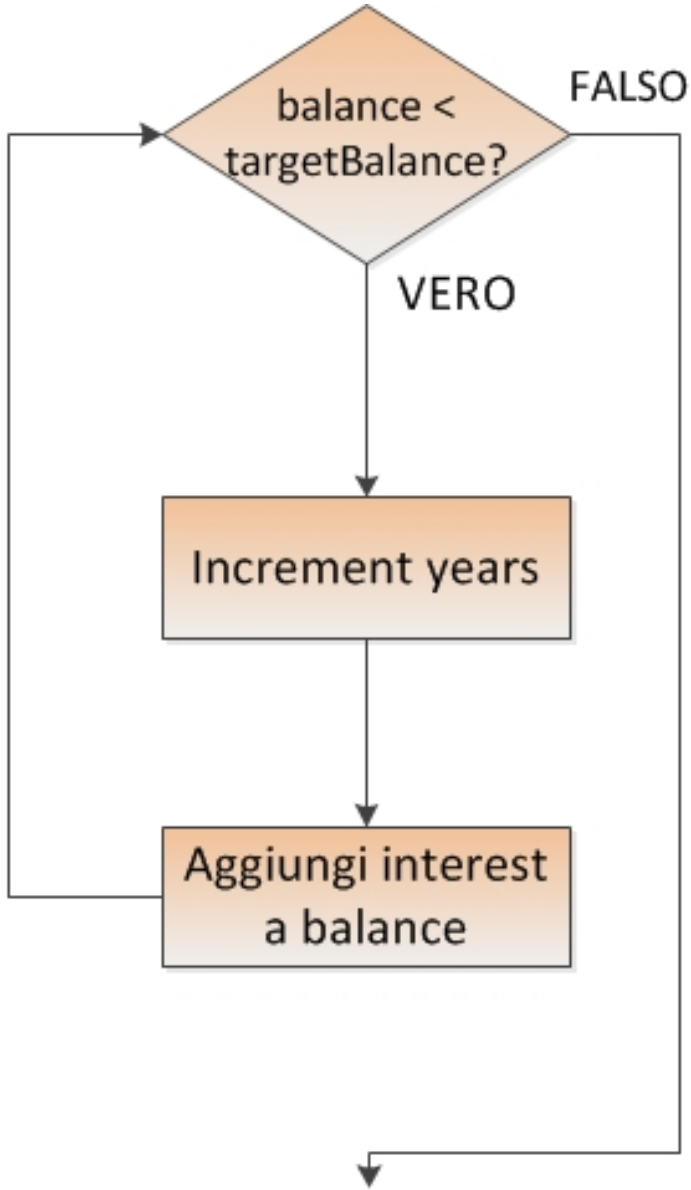
L'enunciato **while** consente la realizzazione di programmi che devono eseguire ripetutamente una serie di azioni finché è verificata una condizione

Algoritmo che risolve il problema

1. All'anno 0 il saldo è 20000
2. **Ripetere** i passi 2 e 3 **finché** il saldo è minore del doppio di 20000, poi passare al punto 4.
3. aggiungere 1 al valore anno corrente;
4. il nuovo saldo è il valore del saldo precedente moltiplicato per 1.05.
5. il risultato è il valore dell'anno corrente.

Il ciclo while

- Sintassi:
  - **while(condizione)**
    enunciato
- Scopo:
  - eseguire un enunciato finché la condizione è vera
- Il **corpo** del ciclo **while** può essere un enunciato qualsiasi, quindi anche un blocco di enunciati
- L'enunciato **while** realizza un **ciclo**



```
1  public class Investment{
2      public Investment(double aBalance, double aRate){
3          balance = aBalance;
4          rate = aRate;
5          years = 0;
6      }
7      public void waitForBalance(double targetBalance){
8          while (balance < targetBalance){
9              years++;
10             double interest = balance * rate / 100;
11             balance = balance + interest;
12         }
13     }
14     public double getBalance(){return balance;}
15     public int getYears(){return years;}
16     private double balance;
17     private double rate;
18     private int years;
19 }
```

```
1  public class InvestmentTester{
2      public static void main(String[] args){
3          final double INITIAL_BALANCE = 10000;
4          final double RATE = 5;
5          Investment invest = new Investment(
               INITIAL_BALANCE, RATE);
6          invest.waitForBalance(2 * INITIAL_BALANCE);
7          int years = invest.getYears();
8          System.out.println("The investment doubled after
               "+ years + " years");
9      }
10 }
```

Cicli infiniti

Esistono errori logici che impediscono la terminazione di un ciclo, generando un **ciclo infinito** L'esecuzione del programma continua **ininterrottamente**

```
1 int year = 0;
2 while (year < 20){
3     double interest = balance * rate / 100;
4     balance = balance + interest;
5 }
```

Cicli infiniti

Esistono errori logici che impediscono la terminazione di un ciclo, generando un **ciclo infinito** L'esecuzione del programma continua **ininterrottamente**

```
1 int year = 0;
2 while (year < 20){
3     double interest = balance * rate / 100;
4     balance = balance + interest;
5 }
```

```
1 int year = 20;
2 while (year > 0){
3     year++;
4     double interest = balance * rate / 100;
5     balance = balance + interest;
6 }
```

Ciclo for

For al posto di while

Molti cicli hanno questa forma:

i = inizio;
while(i < fine)
{*enunciati*; *i++*;}

Sostituzione

Per comodità esiste il ciclo for equivalente

for(i = inizio; i < fine; i++)

Non è necessario che l'incremento sia di una sola unità, né che sia positivo, né che sia intero.

for(inizializzazione; condizione; aggiornamento)
enunciato

Scopo

Eseguire un'**inizializzazione**, poi **ripetere** l'esecuzione di un enunciato ed effettuare un **aggiornamento** finché la **condizione** è vera;

Nota

L'inizializzazione può contenere la definizione di una variabile, che **sarà visibile soltanto all'interno del ciclo**

```
1 for (int y = 1; y <= 10; y++){
2     //corpo
3 }
4 //qui y non e' piu' definita.
```