Istruzioni di controllo:
Parte II

- L'istruzione iterativa **for**
- L'istruzione iterativa **do-while**
- Formulare algoritmi
- La nozione di pseudocodice
- Raffinamento top-down

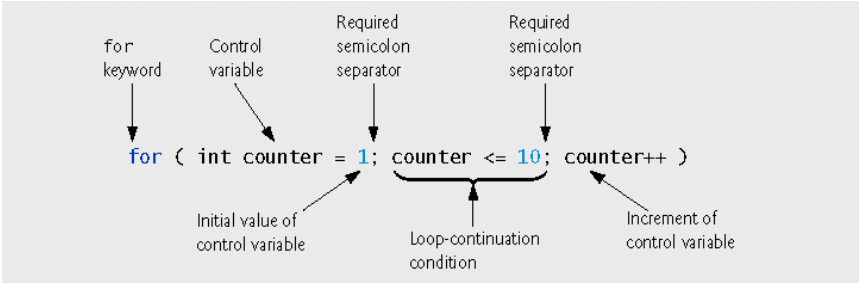
Istruzione for (1)

Java mette a disposizione l'istruzione **for** che consente di specificare l'iterazione controllata da contatore in una singola riga di codice

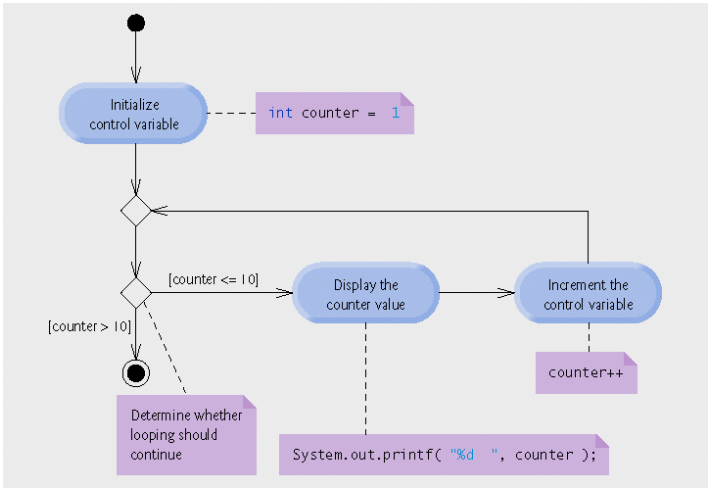
for (*inizializzazione*; *condizioneContinuazioneCiclo*; *incremento*)
istruzione;

```
/* Iterazione controllata da contatore con l'uso dell'istruzione for */

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

      for ( int counter = 1; counter <= 10; counter++ )
         System.out.printf( "%d “, counter );

   }
}  //----- fine classe ForCounter
```

Istruzione for (2)



Istruzione for (3)



Istruzione for (4)

- Scrivere un programma Java che calcoli la somma di tutti i numeri pari compresi tra 1 e 20

```
/* Somma i numeri pari compresi tra 1 e 20 */

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

        int somma = 0;
        for ( int counter = 1; counter <= 20; counter++ )  {
            if ((counter % 2) = = 0)
                somma = somma + counter;
        }

        System.out.printf( "La somma è: %d\n", somma );

}   //----- fine classe SommaPari
```

Dichiara la variabile somma e la inizializza a 0

Cicla facendo assumere alla variabile counter i valori tra 1 e 20

Se counter è pari aggiungi il suo valore alla somma calcolata finora

Visualizza la somma

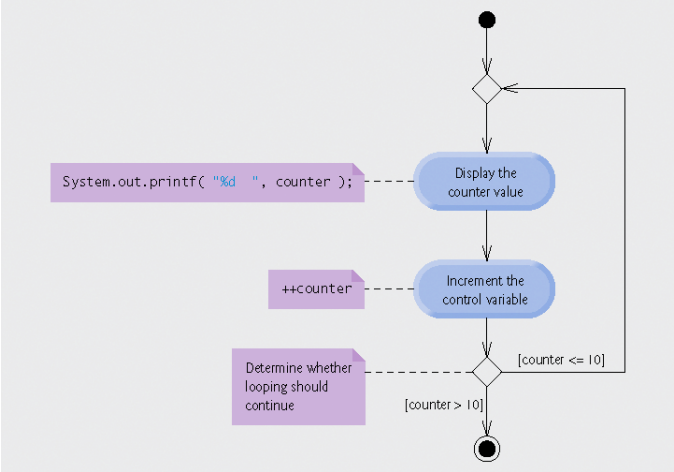
Istruzione for (5)

- Esercizio
  - Modificare il programma SommaPari in modo che calcoli la somma di tutti i numeri pari compresi tra 1 e 20 senza usare la struttura di selezione **if-else** ma solo l'istruzione **for**

Istruzione do…while (1)

- Simile all'istruzione while
- Nell'istruzione while la condizione di continuazione viene controllata all'inizio del ciclo e se la condizione è falsa il ciclo non viene eseguito
- Nell'istruzione do…while la condizione di continuazione viene controllata dopo che è stato eseguito il corpo del ciclo, quindi il corpo del ciclo viene eseguito almeno una volta

Istruzione do…while (2)



Istruzione do…while (3)

```
/* Uso dell'istruzione do…while */

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

       int counter = 1;
       do {
          System.out.printf( "%d “, counter );
          counter++;
       } while ( counter <= 10 )

    }
 }   //--- fine classe DoWhileCounter
```


Formulare algoritmi (1)

- Illustriamo come vengono formulati algoritmi attraverso il seguente esempio relativo alla classe GradeBook

Problema

Una classe di 10 studenti ha sostenuto un esame. Avendo a disposizione il voto di ogni studente, compreso tra 0 e 100, determinare la media dei voti

Formulare algoritmi (2)

- La nozione di Pseudocodice
  - Linguaggio informale usato dai programmatori per esprimere algoritmi tralasciando i dettagli sintattici dello specifico linguaggio di programmazione
  - Gli algoritmi scritti in pseudocodice non sono destinati ed essere eseguiti su un elaboratore elettronico
  - Scrivere un algoritmo in pseudocodice in maniera accurata rende facile la sua conversione in un programma Java
  - Lo pseudocodice in generale contiene la descrizione del input, del output e delle azioni da eseguire

Formulare algoritmi (3)

- Definiamo l'ordine di esecuzione delle operazioni attraverso lo pseudocodice
- Usiamo l'iterazione controllata da contatore per ricevere l'input dei 10 voti
- Usiamo una variabile *contatore* (variabile di controllo) per controllare quante volte un gruppo di istruzioni viene eseguito
- Usiamo una variabile *totale* che serve ad accumulare la somma dei voti

Formulare algoritmi (4)

- Pseudocodice

1. Inizializza il totale a 0
2. Inizializza il contatore dei voti a 1
3. Finché il contatore è minore o uguale a 10
4. | stampa il messaggio “Enter grade: ”
5. | prendi dall’input il voto successivo
6. | aggiungi il voto al totale
7. | aggiungi 1 al contatore dei voti
8. Imposta il valore della media al totale diviso 10
9. Visualizza il totale e la media

Formulare algoritmi (5)

Aggiungiamo alla classe GradeBook il seguente metodo

```
// Determina la media di 10 voti inseriti dall'utente
public void determineClassAverage()
   {
      Scanner input = new Scanner( System.in );

      int total;          // somma dei voti
      int gradeCounter;   // numero dei voti da inserire
      int grade;          // voto inserito dall'utente
      int average;        // media dei voti

      total = 0;
      gradeCounter = 1;

      while ( gradeCounter <= 10 )
      {
         System.out.print( "Enter grade: " );
         grade = input.nextInt();
         total = total + grade;
         gradeCounter = gradeCounter + 1;
      }

      average = total / 10;

      System.out.printf( "\nTotal of all 10 grades is %d\n", total );
      System.out.printf( "Class average is %d\n", average );
   } // --- fine del metodo determineClassAverage
```

Dichiara le variabili richieste

1. Inizializza il totale a 0
2. Inizializza il contatore dei voti a 1

3. Finché il contatore è minore o uguale a 10
4. stampa il messaggio “Enter grade: ”
5. prendi dall’input il voto successivo
6. aggiungi il voto al totale
7. aggiungi 1 al contatore dei voti

8. Imposta il valore della media al totale diviso 10

9. Visualizza il totale e la media

Formulare algoritmi (6)

Aggiungiamo la classe GradeBookTest

```
// Crea un oggetto GradeBook e invoca determineClassAverage()

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

        GradeBook myGradeBook = new GradeBook(
                "CS101 Introduction to Java Programming" );

        myGradeBook.displayMessage();
        myGradeBook.determineClassAverage();

    } // fine main

} // fine classe GradeBookTest
```

Crea un nuovo oggetto di classe GradeBook e lo assegna alla variabile myGradeBook. Il nome del corso è passato come parametro di tipo stringa al costruttore

Chiama il metodo displayMessage di myGradeBook che visualizza un messaggio di benvenuto

Chiama il metodo determineClassAverage di myGradeBook che per calcolare la media dei 10 voti inseriti dall'utente

Formulare algoritmi (7)

- Esercizio
  - Riformulare il codice del metodo determineClassAverage() utilizzando l'istruzione iterativa **for**
- Domanda
  - Dobbiamo modificare il codice della classe GradeBookTest?

Formulare algoritmi (8)

- Problema
  Scrivere un programma che calcola la media di un numero arbitrario di studenti deciso di volta in volta a tempo di esecuzione
- Domanda
  - Come fa il programma a sapere quando sono finiti i voti inseriti da input?
- Suggerimenti
  1. Utilizzare l'iterazione controllata da un valore *sentinella*. Quando l'utente inserisce il valore sentinella indica la fine dell'input dei dati
  2. Utilizzare –1 come valore sentinella

Formuliamo prima l'algoritmo in pseudocodice e poi proponiamo la soluzione in Java

Formulare algoritmi (9)

- Formuliamo lo pseudocodice con una tecnica nota come raffinamento top-down per passi successivi utile per il progetto di programmi ben strutturati
- Top: *determinare la media dei voti di una classe*
- Primo raffinamento: utilizziamo una struttura sequenziale per dividere il programma in tre fasi logiche
  1. Inizializzare le variabili
  2. Eseguire l'input, la somma e il conteggio dei voti
  3. Calcolare e visualizzare la media dei voti

Formulare algoritmi (10)

Secondo raffinamento

1. Inizializzare le variabili: occorre stabilire quali variabili servono ed il loro valore iniziale (stesse variabili del problema precedente)
   - Inizializzare il totale a 0
   - Inizializzare il contatore dei voti inseriti a 0

Formulare algoritmi (11)

Secondo raffinamento

2. Eseguire input, somma e conteggio dei voti:
   - Effettuare l'input del primo voto (può essere il valore sentinella)
   - Finché l'utente non ha digitato il valore sentinella
     - Aggiungere il voto corrente al totale
     - Aggiungere 1 al contatore
     - Effettuare l'input del prossimo voto (può essere il valore sentinella)

Formulare algoritmi (12)

Secondo raffinamento

3. Calcolare e visualizzare la media dei voti:
   - Se il contatore non è uguale a 0
     - Impostare il valore della media al totale diviso contatore
     - Visualizzare la media
   - Altrimenti visualizzare il messaggio “non è stato inserito alcun voto”

Questo controllo è fondamentale per evitare di eseguire una divisione per 0 che darebbe luogo ad un errore logico fatale!

Formulare algoritmi (13)

Pseudocodice

1. Inizializzare il totale a 0
2. Inizializzare il contatore dei voti inseriti a 0
3. Effettuare l’input del primo voto (può essere il valore sentinella)
4. Finché l’utente non ha digitato il valore sentinella
   - 4.1 Aggiungere il voto corrente al totale
   - 4.2 Aggiungere 1 al contatore
   - 4.3 Effettuare l’input del prossimo voto (può essere il valore sentinella)
5. Se il contatore non è uguale a 0
   - 5.1 Impostare il valore della media al totale diviso contatore
   - 5.2 Visualizzare la media
6. Altrimenti visualizzare il messaggio “non è stato inserito alcun voto”

Formulare algoritmi (14)

```
// Determina la media di un numero arbitrario voti inseriti dall'utente

public void determineClassAverage1()  {
    Scanner input = new Scanner( System.in );
    int total;              // somma dei voti
    int gradeCounter;       // numero dei voti inseriti
    int grade;              // voto inserito dall'utente
    double average;         // media dei voti

    total = 0;
    gradeCounter = 0;

    System.out.printf( "Inserisci un voto oppure –1 per uscire: ");
    grade = input.nextInt();

    while ( grade != -1 ) {
        total = total + grade;
        gradeCounter = gradeCounter + 1;
        System.out.print( "Enter grade: " );
        grade = input.nextInt();
        }

    if (gradeCounter != 0) {
        average = total /gradeCounter;
        System.out.printf("La media è %d\n", average);
    }

        else  System.out.printf( "Class average is %d\n", average );
} //   --- fine metodo determineClassAverage
```

Dichiara le variabili richieste

1. Inizializza il totale a 0
2. Inizializza il contatore dei voti inseriti a 0

3. Inserisci il primo voto

4. Finché non viene inserito il valore sentinella
   4.1 aggiungi il voto al totale
   4.2 aggiungi 1 al contatore dei voti
   4.3 prendi dall'input il voto successivo

5. Se il contatore dei voti è diverso da 0
   5.1 Calcola la media
   5.2 Visualizza la media

6. Altrimenti visualizza messaggio

Formulare algoritmi (15)

- I 10 studenti di una classe hanno affrontato l'esame finale di un corso. Scrivere un programma Java che:
  - Effettua l'input dei risultati (i possibili valori di input sono: 1 = passato e 2 = bocciato)
  - Conta il numero di risultati di ogni tipo
  - Visualizza il riepilogo dei risultati che indichi il numero di studenti promossi e bocciati

Formulare algoritmi (16)

```
// Analizza ed elabora i risultati di un esame

import java.util.Scanner; // importa la classe Scanner

public class Analysis {
   public void processExamResults()  {
      Scanner input = new Scanner( System.in );

      int passes = 0;              // numero di esami passati
      int failures = 0;            // numero di esami falliti
      int studentCounter = 1;      // contatore studenti
      int result;                  // singolo risultato

      while ( studentCounter <= 10 ) {
         System.out.print( "Inserisci risultato (1 = passato, 2 = fallito): " );
         result = input.nextInt();
         if ( result = = 1 )
               passes = passes + 1;
         else
               failures = failures + 1;
         studentCounter = studentCounter + 1;
      } // --- fine ciclo while

      System.out.printf( "Passati: %d\nFalliti: %d\n", passes, failures );

   } // --- fine metodo processExamResults
} // --- fine classe Analysis
```

Dichiara le variabili richieste

Conta il numero di risultati di ogni tipo

Visualizza il riepilogo dei risultati

Formulare algoritmi (17)

```
// Programma di test per la classe Analysis

public class AnalysisTest
{
   public static void main( String args[] )
   {
      Analysis A = new Analysis(); // crea un oggetto Analysis
      A.processExamResults(); // invoca il metodo che elabora i risultati
   } // --- fine main

} // --- fine classe AnalysisTest
```

Formulare algoritmi (18)

- Esercizio
  - Riformulare il codice del metodo processExamResult() utilizzando l'istruzione iterativa **for**
- Domanda
  - Dobbiamo modificare il codice della classe AnalysisTest?

Riepilogo controllo in Java