

La programmazione strutturata - II

Stefano Mizzaro

Dipartimento di matematica e informatica
 Università di Udine
<http://www.dimi.uniud.it/mizzaro/>
 mizzaro@dimi.uniud.it
 Programmazione, lezione 5
 9 ottobre 2007

Oggi

- Riassunto
- Input: **Leggi.java**
- Sviluppo incrementale
- Ancora programmazione strutturata
- Iterazione

Stefano Mizzaro - Prog. strutt. 2 2

Riassunto (1/2)

- Esempi
- I mattoni di un programma Java:
 - Parole riservate
 - Identificatori
 - Variabili
 - Letterali
 - Costanti
 - Operatori
 - Espressioni
 - Assegnamento, dichiarazioni di variabile
 - Commenti
 - Tipo
- Puntualizzazioni ed esempi
 - Costanti
 - Valori booleani o logici
 - Operatore condizionale ?:
- Conversione di tipo
- Valori temporanei delle espressioni

Stefano Mizzaro - Prog. strutt. 2 3

Riassunto (2/2)

- Strutture di controllo della programmazione strutturata
 - Sequenza
 - ; e {}
 - Selezione
 - if, if/else, switch/case
 - Iterazione (ripetizione)
 - while, do/while, for
 - (break, continue)

Stefano Mizzaro - Prog. strutt. 2 4

Sequenza e selezione

```
I1;
I2;
I3;
```

```
if (C)
  I1
else
  I2
```

```
if (C)
  I
```

Stefano Mizzaro - Prog. strutt. 2 5

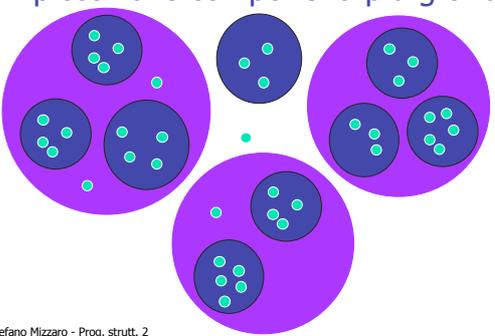
if e while

```
if (C)
  I
```

```
while (C)
  I
```

Stefano Mizzaro - Prog. strutt. 2 6

Analisi sistematica: dai mattoni più piccoli alle componenti più grandi



Stefano Mizzaro - Prog. strutt. 2

7

Scaletta

- Riassunto
- Input: **Leggi.java**
- Sviluppo incrementale
- Ancora programmazione strutturata
- Iterazione

Stefano Mizzaro - Prog. strutt. 2

8

L'input

- Come inserire dati durante l'esecuzione
- La **System.in.read()** non è il massimo:
 - Legge un solo carattere
 - Ne restituisce il codice ASCII/Unicode
- È possibile "estendere" il Java con istruzioni di input più comode
 - Lo abbiamo fatto per voi... scaricate il file **Leggi.java**, sul sito Web del corso

Stefano Mizzaro - Prog. strutt. 2

9

Uso di Leggi.java

- **Leggi.java** va messo (per ora) **nella stessa directory** del programma che lo usa!!
- Potete scrivere nei vostri programmi nuove istruzioni per la lettura di:
 - **boolean, byte, short, int, long, float, double, char, String**

```
byte b;
short s;
int i;
long l;
float f;
double d;
boolean t;
char c;
String x;

b = Leggi.unByte();
s = Leggi.unoShort();
i = Leggi.unInt();
l = Leggi.unLong();
f = Leggi.unFloat();
d = Leggi.unDouble();
t = Leggi.unBoolean();
c = Leggi.unChar();
x = Leggi.unoString();
```

Stefano Mizzaro - Prog. strutt. 2

10

Esempio

- Programma per trovare il massimo fra due numeri interi

```
...
int a, b;
a = Leggi.unInt();
b = Leggi.unInt();
if (a>b)
    System.out.println(
        "Il massimo e' il primo");
else
    System.out.println(
        "Il massimo e' il secondo");
...
```

Stefano Mizzaro - Prog. strutt. 2

11

Scaletta

- Riassunto
- Input: **Leggi.java**
- Sviluppo incrementale
- Ancora programmazione strutturata
- Iterazione

Stefano Mizzaro - Prog. strutt. 2

12

Sviluppo incrementale

- Come si scrive un programma?
- NON "di getto dalla prima riga"
- Ma per raffinamenti successivi
 - Versioni intermedie
 - In pseudocodice (misto di Java e altro)
 - Ogni versione raffina alcuni aspetti della precedente

Stefano Mizzaro - Prog. strutt. 2 13

Raffinamenti successivi

Stefano Mizzaro - Prog. strutt. 2 14

E POI editing-compilazione-esecuzione

Stefano Mizzaro - Prog. strutt. 2 15

Esempio

- Ultra-semplice, ma...
- Classe di 10 studenti, esame
- Voti:
 - O (ottimo): 10
 - B (buono): 8
 - S (sufficiente): 6
 - I (insufficiente): 4
- Calcolare la media aritmetica

Stefano Mizzaro - Prog. strutt. 2 16

L'idea/L'algoritmo

- Uso una variabile **media**
- A cui aggiungo man mano i voti
- E poi divido per 10

Stefano Mizzaro - Prog. strutt. 2 17

La prima versione (pseudocodice)

```
while ("ci sono ancora voti") {
    "Leggi il prossimo voto";
    "Aggiungi a media";
}
"Stampa media / 10";
```

Stefano Mizzaro - Prog. strutt. 2 18

La seconda versione

```

class Voti {
    public static void main (String[] args)
        throws IOException {
        "azzerata media";
        "inizializza i";
        while (i <= 10) {
            "Leggi voto";
            if (voto == 'O')
                "Aggiungi 10 a media";
            else if (voto == 'B')
                "Aggiungi 8 a media";
            else if (voto == 'S')
                "Aggiungi 6 a media";
            else if (voto == 'I')
                "Aggiungi 4 a media";
            "incrementa i";
        }
        "Stampa media / 10";
    }
}
    
```

La terza versione

```

/* Programma per calcolare la media dei voti*/
import java.io.*;
class Voti {
    public static void main (String[] args)
        throws IOException {
        int voto, media, i;
        media = 0;
        i = 1;
        while (i <= 10) {
            voto = System.in.read();
            if (voto == 'O')
                media = media + 10;
            else if (voto == 'B')
                media = media + 8;
            else if (voto == 'S')
                media = media + 6;
            else if (voto == 'I')
                media = media + 4;
            i = i + 1;
        }
        media = media / 10;
        System.out.print("Media = ");
        System.out.println(media);
    }
}
    
```

Come si scrive un programma

- Prima si riflette su carta (schemi, disegni, riferimenti ad altri programmi già scritti, ...)
- POI ci si mette davanti al computer e si incomincia il ciclo editing-compilazione-esecuzione
 - A volte si deve tornare indietro...
- È più scomodo ragionare a basso livello (linguaggio di programmazione)

Raffinamenti successivi

Mai accontentarsi della prima versione di un programma

- Anche se siete riusciti a scrivere un programma funzionante, lo si può (quasi) sicuramente migliorare...

```

if (voto == 'O')
    media = media + 10;
else if (voto == 'B')
    media = media + 8;
else if (voto == 'S')
    media = media + 6;
else if (voto == 'I')
    media = media + 4;
i = i + 1;

switch (voto) {
    case 'O':
        media = media + 10;
        break;
    case 'B':
        media = media + 8;
        break;
    case 'S':
        media = media + 6;
        break;
    case 'I':
        media = media + 4;
        break;
}
    
```

Ancora su if e incolonnamento

- Meglio:


```

if (voto == 'O')
    media = media + 10;
else if (voto == 'B')
    media = media + 8;
else if (voto == 'S')
    media = media + 6;
else if (voto == 'I')
    media = media + 4;
i = i + 1;
            
```
- Peggio:


```

if (voto == 'O')
    media = media + 10;
else
    if (voto == 'B')
        media = media + 8;
    else
        if (voto == 'S')
            media = media + 6;
        else
            if (voto == 'I')
                media = media + 4;
i = i + 1;
            
```

Esempi ed esercizi

- Migliorare il programma precedente
- Scrivete un programma che visualizza tutti i numeri da 10 a 1 (in ordine decrescente)
- Scrivete un programma che visualizza tutti i numeri da 1 a 10, ma usando i letterali 0 e 9 anziché 1 e 10

Stefano Mizzaro - Prog. strutt. 2 25

Esercizio

- Cosa fa il programma seguente?

```
class CicloWhile {
    public static void main (String[] args) {
        int i;

        i = 1;
        while (i <= 10)
            System.out.println(i);
        i = i + 1;
        System.out.println("Fine stampa");
    }
}
```

Stefano Mizzaro - Prog. strutt. 2 26

Esempi ed esercizi

- Scrivete un programma che visualizza tutti i numeri fra 1 e 100 che siano multipli di 2 o di 3

```
i = 1;
while (i <= 100) {
    if (i % 2 == 0 || i % 3 == 0)
        System.out.println(i);
    i = i + 1;
}
```

Stefano Mizzaro - Prog. strutt. 2 27

Esempi ed esercizi

- Scrivete un programma che visualizza tutti i numeri fra 1 e 100 che siano multipli di 2 e di 3

```
i = 1;
while (i <= 100) {
    if (i % 2 == 0 && i % 3 == 0)
        System.out.println(i);
    i = i + 1;
}
```

- MEGLIO! Multipli di 2 e di 3?? Ah, multipli di 6!!

```
i = 6;
while (i <= 100) {
    System.out.println(i);
    i = i + 6;
}
```

Stefano Mizzaro - Prog. strutt. 2 28

Il do/while

- Un'altra istruzione di iterazione
- **do I while (C);**
 - "Fai I mentre c è vera"
- Differenza dal **while**:
 - Il controllo del valore di c viene effettuato alla fine dell'iterazione, anziché all'inizio
 - Con **while (C) I**, potrebbe darsi che I non sia eseguita neanche una volta
 - Con **do I while (C)**, sicuramente I viene eseguita almeno una volta

Stefano Mizzaro - Prog. strutt. 2 29

Diagramma di flusso

- Differenze fra **while** e **do/while**

```
while (C)
    I
```

```
do {
    I
} while (C);
```

Stefano Mizzaro - Prog. strutt. 2 30

Esempio

- Stampare i numeri da 1 a 10 con un **do/while**

```

i = 1;
do {
    System.out.println(i);
    i = i + 1;
} while (i <= 10);

i = 0;
do {
    i = i + 1;
    System.out.println(i);
} while (i < 10);
    
```

Stefano Mizzaro - Prog. strutt. 2 31

Puntualizzazioni

- Le parentesi graffe sono superflue ("do" e "while" fanno già da parentesi)
 - Miglior metterle per leggibilità
 - " } while (C)" non è senz'altro un inizio di ciclo while (sarebbe "while (C)")
- do/while è superfluo, ma fa comodo
- Si usa meno del while perché "non uniforme"

Stefano Mizzaro - Prog. strutt. 2 32

Simulare il do/while con il while

- Bisogna essere sicuri che I sia eseguita almeno una volta...

```

do {
    I
} while (C);

I;
while (C) {
    I
}

boolean primo;
primo = true;
while (primo || C) {
    primo = false;
    I;
}
    
```

Stefano Mizzaro - Prog. strutt. 2 33

Simulare il while con il do/while

- Più difficile
- Non è detto che I sia eseguita almeno una volta...
- ... bisogna innanzitutto controllare c...

```

while (C) {
    I
}

if (C)
do {
    I
} while (C);
    
```

Stefano Mizzaro - Prog. strutt. 2 34

Esempio: controllo input

- Se volete leggere un carattere che DEVE essere 's' o 'n': **Leggi.unChar()**

```

do {
    c = System.in.read();
} while ((c != 'n') && (c != 's'));
    
```

- ...meglio del while perché la prima iterazione deve essere "diversa", non uniforme

Stefano Mizzaro - Prog. strutt. 2 35

Riassunto

- Strutture di controllo della programmazione strutturata
 - Sequenza
 - ; e {}
 - Selezione
 - if, if/else, switch/case
 - Iterazione (ripetizione)
 - while, do/while, for
 - (break, continue)
- Input più agevole (**Leggi.java**)
- Sviluppo incrementale

Stefano Mizzaro - Prog. strutt. 2 36

Prossima lezione

- Continuiamo con la programmazione strutturata
 - Iterazione (ripetizione)
 - `while`, `do/while`, `for`
 - `(break, continue)`

Stefano Mizzaro - Prog. strutt. 2 37

