

La programmazione strutturata - II

Stefano Mizzaro

Dipartimento di matematica e informatica
Università di Udine
<http://www.dimi.uniud.it/mizzaro/>
mizzaro@uniud.it
Programmazione, lezione 5
9 marzo 2015

Movie time!

- Code.org
 - <http://code.org/>
- Did you know?
 - <https://www.youtube.com/watch?v=XrJjfDUzD7M>
- Microsoft Hololens
 - <https://www.youtube.com/watch?v=aThCr0PsyUA>

S. Mizzaro - Programmazione strutturata 2

2

Morali

- Importanza della programmazione
 - Dal punto di vista tecnico e culturale
- Importanza dell'informazione
 - Dal punto di vista "sociale" (sviluppo tecnologico ha portato all'aumento ecc.) e culturale
 - Va gestita
- Variazioni rapide e importanti
 - Chissà che lavoro farò
 - Chissà che tecnologia ci sarà
 - Chissà che linguaggio di programmazione

S. Mizzaro - Programmazione strutturata 2

3

Oggi

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

S. Mizzaro - Programmazione strutturata 2

4

Riassunto (1/2)

- | | |
|---|---|
| <ul style="list-style-type: none"> ■ 1. Esempi ■ 2. I mattoni di un programma Java: <ul style="list-style-type: none"> ■ Parole riservate ■ Identificatori ■ Variabili ■ Letterali ■ Costanti ■ Operatori ■ Espressioni ■ Assegnamento ■ Commenti ■ Tipo, dichiarazione di variabile | <ul style="list-style-type: none"> ■ 3. Puntualizzazioni ed esempi <ul style="list-style-type: none"> ■ Costanti ■ Valori booleani o logici ■ Operatore condizionale <code>?:</code> ■ Conversione di tipo ■ Valori temporanei delle espressioni |
|---|---|

S. Mizzaro - Programmazione strutturata 2

5

Riassunto (2/2)

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

S. Mizzaro - Programmazione strutturata 2

6

Sequenza e selezione

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

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

```
if (C)
  I
```

S. Mizzaro - Programmazione strutturata 2 7

if e while

```
if (C)
  I
```

```
while (C)
  I
```

S. Mizzaro - Programmazione strutturata 2 8

Analisi sistematica: dai mattoni più piccoli alle componenti più grandi

S. Mizzaro - Programmazione strutturata 2 9

Scaletta

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

S. Mizzaro - Programmazione strutturata 2 10

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 blog del corso

S. Mizzaro - Programmazione strutturata 2 11

Uso di Leggi.java

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

- **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**

```
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();
```

S. Mizzaro - Programmazione strutturata 2 12

Esempio

- Programma per trovare il massimo fra due numeri interi

```

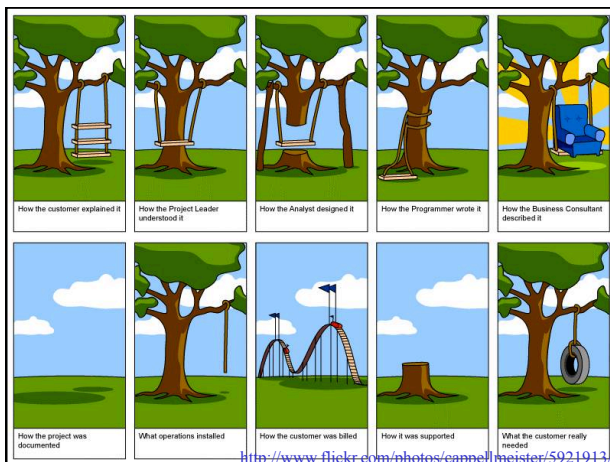
...
int a, b;
a = Leggi.unInt();
b = Leggi.unInt();
if (a>b)
    System.out.println(
        "Il massimo e' il primo");
else
    System.out.println(
        "Il massimo e' il secondo");
...
    
```

S. Mizzaro - Programmazione strutturata 2 13

Scaletta

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

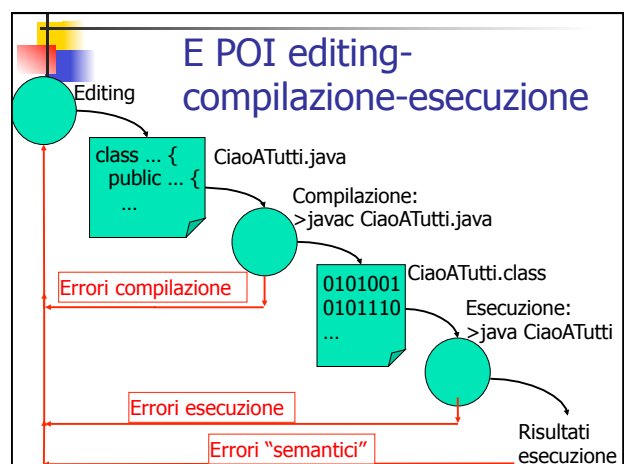
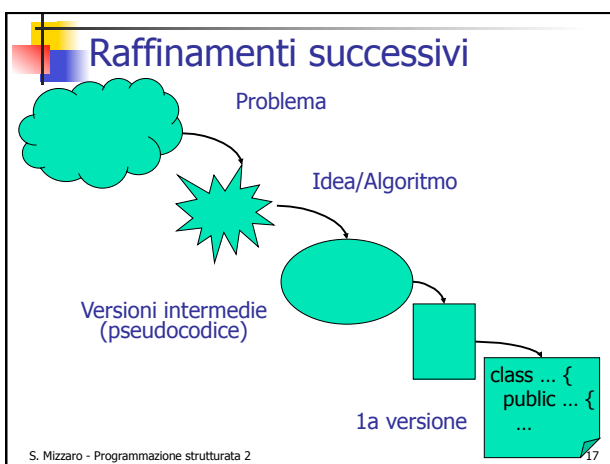
S. Mizzaro - Programmazione strutturata 2 14



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

S. Mizzaro - Programmazione strutturata 2 16



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



S. Mizzaro - Programmazione strutturata 2

19

L'idea/L'algoritmo

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



S. Mizzaro - Programmazione strutturata 2

20

La prima versione (pseudocodice)

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



S. Mizzaro - Programmazione strutturata 2

21

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);
  }
}
```

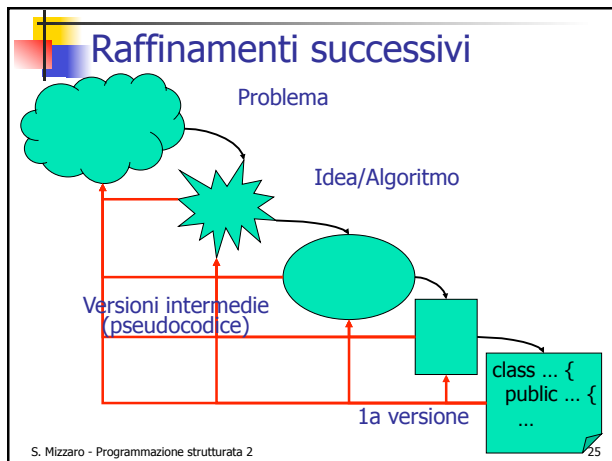
```
class ... {
  public ... {
    ...
  }
}
```

Come si scrive un programma

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

S. Mizzaro - Programmazione strutturata 2

24



L'esperienza

- Ovviamente l'esperienza aiuta
 - Sapere che una certa parte del programma si può fare perché lo si è già fatto
 - (o si è fatto qualcosa di simile)
 - ci lascia più tranquilli,
 - ci consente di astrarre,
 - ci consente di fare meno fatica
- E anche le conoscenze tecniche aiutano
- ==> fare esercizi!!

S. Mizzaro - Programmazione strutturata 2 26

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;
}
```

S. Mizzaro - Programmazione strutturata 2 27

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;
```

S. Mizzaro - Programmazione strutturata 2 28

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

S. Mizzaro - Programmazione strutturata 2 29

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");
}
}
```

S. Mizzaro - Programmazione strutturata 2 30

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;
}
```

S. Mizzaro - Programmazione strutturata 2

31

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;
}
```

S. Mizzaro - Programmazione strutturata 2

32

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

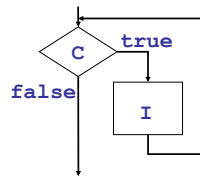
S. Mizzaro - Programmazione strutturata 2

33

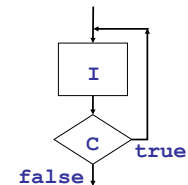
Diagramma di flusso

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

```
while (C)
    I
```



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



S. Mizzaro - Programmazione strutturata 2

34

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);
```

S. Mizzaro - Programmazione strutturata 2

35

Puntualizzazioni

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

S. Mizzaro - Programmazione strutturata 2

36

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;
}
```

S. Mizzaro - Programmazione strutturata 2

37

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);
```

S. Mizzaro - Programmazione strutturata 2

38

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

S. Mizzaro - Programmazione strutturata 2

39

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

S. Mizzaro - Programmazione strutturata 2

40

Prossima lezione

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

S. Mizzaro - Programmazione strutturata 2

41

Esercizio

- Stampare le cifre di un numero intero

S. Mizzaro - Programmazione strutturata 2

42