

## Introduzione alla programmazione strutturata

**Stefano Mizzaro**

Dipartimento di matematica e informatica  
Università di Udine  
<http://www.dimi.uniud.it/~mizzaro>  
mizzaro@dimi.uniud.it  
Programmazione, lezione 3  
19 gennaio 2004

## Riassunto

- Capp. 1 e 2 (sorvolando su qualche dettaglio)
- Esempi
- Mattoni di un programma
  - Parole riservate, variabili, costanti, letterali
  - Operatori, costruzione di espressioni
  - Assegnamento
  - Tipi predefiniti
  - ...

Stefano Mizzaro - Prog. strutt.

2

## Oggi

- Puntualizzazioni ed esempi
  - Costanti
  - Valori booleani o logici
  - Operatore condizionale `?:`
  - Il modulo `%`
- Conversione di tipo
- Valori temporanei delle espressioni
- Come mettere insieme i mattoni:
  - Introduzione alle strutture di controllo della programmazione strutturata
    - Sequenza
    - Selezione
    - (Iterazione)

Stefano Mizzaro - Prog. strutt.

3

## Utilità delle costanti

- A cosa servono le costanti?
- Se un programma funziona correttamente e tolgo tutti i `final`, cosa succede?
- Costanti, non letterali!
- Risposte:
  - Disciplina: troppa libertà è pericolosa (tipi, ...)
  - Leggibilità
- Regola (eccessiva!!): in un programma gli unici letterali dovrebbero essere 0, 1 e al massimo 2

Stefano Mizzaro - Prog. strutt.

4

## Valori booleani

- Letterali: `true` e `false`
- Operatori:
  - `&` | `!`
  - `&&` | `||`
- Un'espressione booleana è un'espressione!
- Esempi
  - Numero compreso in un intervallo?
  - Numero compreso in uno di due intervalli?

Stefano Mizzaro - Prog. strutt.

5

## L'operatore di modulo `%`

- Resto della divisione
- Molto utile
  - `x % 2`
  - `x % 10`
  - `x % 100`
  - ...
- Divisibile per 3?

Stefano Mizzaro - Prog. strutt.

6

## Regole

- Rispettare le convenzioni
- Usare i commenti
- Io non lo faccio:
  - per esigenze di spazio
  - perché commento a voce i programmi
- Voi lo fate
  - perché altrimenti non passate l'esame ☺

Stefano Mizzaro - Prog. strutt.

7

## Conversioni di tipo

- In due versi
  - Promozioni (da un tipo più "piccolo" a uno più "grande")  
`byte` → `short` → `int` → `long` → `float` → `double`
  - Cast (da un tipo più "grande" a uno più "piccolo")

Stefano Mizzaro - Prog. strutt.

8

## Promozioni

```
byte b;
int i;
b = 10;
i = b;
```

- Nel secondo assegnamento ci sono 2 tipi diversi...
  - (anche nel primo...)
- C'è una conversione automatica
- Vediamo un esempio più generale

Stefano Mizzaro - Prog. strutt.

9

## Promozioni

```
class Prova {
    public static void main (String[] args) {
        byte b;
        short s;
        int i;
        long l;
        float f;
        double d;
        b = 10;
        s = b;
        i = s;
        l = i;
        f = l;
        d = f;
    }
}
```

Stefano Mizzaro - Prog. strutt.

10

## L'altro verso: il cast

```
byte b;   byte b;   byte b;
int i;   int i;   int i;
b = 10;  i = 128;  i = 10;
i = b;   b = i;   b = i;
```

```
byte b;   byte b;
int i;   int i;
i = 10;  i = 128;
b = (byte) i;  b = (byte) i;
```

Stefano Mizzaro - Prog. strutt.

11

## Cast

- Operatore di cast:
  - si scrive il nome del tipo
  - fra parentesi tonde
  - prima di un'espressione
- È un meccanismo più generale di quello che vediamo qui
  - (tipi definibili dal programmatore...)
  - Ne parleremo
- "Assunzione di responsabilità del programmatore"
- È delicato...

Stefano Mizzaro - Prog. strutt.

12

## Valori temporanei delle espressioni

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

        m = 2000000000;
        n = 2000000000;
        i = m * n / m;
        System.out.println(i);
    }
}
```

```
>javac Prova.java
>java Prova
0
>
```

Stefano Mizzaro - Prog. strutt.

13

## Spiegazione

- Il valore di  $m * n$  viene memorizzato in una variabile temporanea gestita dal compilatore
- Di che tipo è?
- Il compilatore si basa sui tipi che compaiono nell'espressione: `int`
- Ma il valore ottenuto non ci sta in un `int`...
- Soluzioni
  - Dichiarare le tre variabili di tipo `long`
  - Dichiarare solo `i` di tipo `long` e usare il cast:
 

```
i = (long)m * n / m;
```
  - Dichiarare solo `m, n` di tipo `long` e usare il cast:
 

```
i = (int)m * n / m;
```

Stefano Mizzaro - Prog. strutt.

14

## Riassunto

- Capp. 1 e 2 (sorvolando su qualche dettaglio)
- Esempi
- Mattoni di un programma
  - Costruzione di espressioni
  - Assegnamento
  - Tipi predefiniti
  - ...
- Passiamo a un livello di "granularità" maggiore

Stefano Mizzaro - Prog. strutt.

15

## Introduzione alla programmazione strutturata

- Un programma è fatto da istruzioni
- Abbiamo visto
  - pezzi di istruzioni
  - (poche) istruzioni complete
- Vediamo come metterli insieme

Stefano Mizzaro - Prog. strutt.

16

## Le 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.

17

## Sequenza

- Ogni istruzione termina con un punto e virgola ;
- Si possono raggruppare più istruzioni "per farle diventare come un'unica istruzione" con le parentesi graffe `{}`
- Uno spazio = più spazi = tabulazione = ritorno carrello (a capo)
- Incolonnare per leggibilità
  - Aggiungere spazi (2, 4) ad ogni `{`
  - Togliere spazi ad ogni `}`
  - `{` a fine riga
  - `}` su riga isolata (qualche eccezione...)

Stefano Mizzaro - Prog. strutt.

18

### Esempio

```

/* Programma per calcolare l'area di un
triangolo */
class AreaTriangolo {
    public static void main (String[] args) {
        int base;
        int altezza;
        int area;

        base = 5;
        altezza = 12;
        area = base * altezza / 2;
        System.out.println(area);
    }
}
    
```

Stefano Mizzaro - Prog. strutt. 19

### Esempio

```

class Stampa1_10 {
    public static void main(String[] args) {
        int contatore;
        contatore = 1;
        while (contatore <= 10) {
            System.out.println(contatore);
            contatore = contatore + 1;
        }
    }
}
    
```

Stefano Mizzaro - Prog. strutt. 20

### Selezione binaria

- **if, if/else**
- Alternativa: fai o questo o quello
- Se una certa condizione è vera fai questo altrimenti fai quello
  - "Condizione": espressione booleana (di tipo **boolean**, quindi con valore **true** o **false**)
  - "Questo" e "quello": istruzioni, eventualmente composte

Stefano Mizzaro - Prog. strutt. 21

### Esempio

```

/* Qual e' il massimo fra 2 cifre? */
import java.io.*;
class Max2 {
    public static void main(String[] args)
        throws IOException {
        int a, b;
        a = System.in.read();
        b = System.in.read();
        if (a>b) {
            System.out.println(
                "Il massimo e' il primo");
        } else {
            System.out.println(
                "Il massimo e' il secondo");
        }
    }
}
    
```

Stefano Mizzaro - Prog. strutt. 22

↳ Eccezione...

### Forma generale

- **if** e **else** sono parole riservate
- **C** è una condizione (espr. booleana)
- **I1** e **I2** sono istruzioni
  - Eventualmente composte (fra { })
- Le graffe sono indispensabili solo per istruzioni composte
  - Ma metterle sempre può essere una buona idea...
- La parte **else** può anche non esserci
- Incolonnamento e graffe
  - Rientrare il ramo **if** e il ramo **else**
  - Incolonnare l'**else** sotto l'**if** corrispondente

```

if (C)
    I1
else
    I2

if (C) {
    I1;
    I2;
} else {
    I3;
    I4;
}

if (C)
    I1
    
```

Stefano Mizzaro - Prog. strutt. 23

### Diagrammi di flusso

- (flow chart)
- Possono essere utili per capire
- Non per scrivere programmi!
- "Ramo **if**" e "ramo **else**"

Stefano Mizzaro - Prog. strutt. 24

## Esempi/esercizi

- Un frammento di programma che dica se un numero  $x$  è positivo o negativo
- Un frammento di programma che dica se un numero  $x$  è positivo o negativo, e che incrementi di uno il valore assoluto di  $x$
- Un frammento di programma che dica se un numero  $x$  è pari o dispari
- Un frammento di programma che dica se un numero  $x$  è zero, strettamente positivo o strettamente negativo

Stefano Mizzaro - Prog. strutt.

25

## Gli `if` annidati

- `if` dentro a `if` (dentro a `if...`)
- Nel ramo `if` o nel ramo `else`
- Fare attenzione
- Regola: l'`else` si riferisce all'ultimo `if`
- Se la regola non ci dà quello che vogliamo, usiamo le `{ }`
- Annidamenti `if` e annidamenti `else`

Stefano Mizzaro - Prog. strutt.

26


## Preferire gli annidamenti `else`

- Non raggiungono il margine destro:
  - `} else if` tutto su una riga

```

if (C1)
  if (C2) {
    if (C3)
      I1;
    } else
      I2;
  else
    I3;

```



```

if (!C1)
  I3;
else if (!C2)
  I2;
else if (C3)
  I1;

```

Stefano Mizzaro - Prog. strutt.

27

## n-aria: `switch/case`

- Selezione non fra 2 alternative, ma fra più alternative

```

switch (V) {
  case V1: I1; break;
  case V2: I2; break;
  ...
  case Vn: In; break;
  default: I; break;
}

```

Stefano Mizzaro - Prog. strutt.

28

## Esempio

```

switch (c) {
  case 'a':
    System.out.println("Vocale");
    break;
  case 'e':
    System.out.println("Vocale");
    break;
  case 'i':
    System.out.println("Vocale");
    break;
  case 'o':
    System.out.println("Vocale");
    break;
  case 'u':
    System.out.println("Vocale");
    break;
  default: System.out.println("Consonante");
    break;
}

```

Stefano Mizzaro - Prog. strutt.

29

## Esempio (più sintetico)

```

switch (c) {
  case 'a': // Vai al prox.
  case 'e': // Vai al prox.
  case 'i': // Vai al prox.
  case 'o': // Vai al prox.
  case 'u':
    System.out.println("Vocale");
    break;
  default: System.out.println("Consonante");
    break;
}

```

Stefano Mizzaro - Prog. strutt.

30

## Più e meno generale dell'`if`

- Più generale perché selezione n-aria, non solo binaria
- Meno generale perché la condizione deve essere del tipo `V==Vi`
- (mentre nell'`if` può essere qualsiasi)
  - Esempio: `x > 0?`
- `Vi` deve essere un'espressione valutabile dal compilatore
  - (un letterale, ...)
- di tipo `byte`, `short`, `int`, `long` o `char`

Stefano Mizzaro - Prog. strutt.

31

## Altro esempio

```
switch (giorno) {
  case 1:
    System.out.println("Lunedì");
    break;
  case 2:
    System.out.println("Martedì");
    break;
  case 3:
    System.out.println("Mercoledì");
    break;
  case 4:
    System.out.println("Giovedì");
    break;
  case 5:
    System.out.println("Venerdì");
    break;
  case 6:
    System.out.println("Sabato");
    break;
  case 7:
    System.out.println("Domenica");
    break;
  default:
    System.out.println("Valore non lecito");
    break;
}
```

Stefano Mizzaro - Prog. strutt.

32

## Riassunto

- Fine "mattoni"
- Introduzione programmazione strutturata
  - Sequenza
  - Selezione
  - (Iterazione)

Stefano Mizzaro - Prog. strutt.

33

## Prossima lezione

- Continuiamo con la programmazione strutturata
- Iterazione
- Come si scrive un programma
- Esercizio: scrivere un programma che trova il maggiore fra quattro caratteri letti in input, lo assegna a una variabile e visualizza "il maggiore e' il primo/secondo/..."

Stefano Mizzaro - Prog. strutt.

34