

Programmazione e laboratorio: introduzione al corso

Stefano Mizzaro

Dipartimento di matematica e informatica
Università di Udine
<http://www.dimi.uniud.it/mizzaro/>
mizzaro@uniud.it
Lezione 1, 23 febbraio 2015

Oggi

- Presentazione del corso
 - I docenti
 - L'esame
 - Raccomandazioni
 - Testi e materiale didattico
 - Programma preliminare del corso
- Inizio
 - I primi esempi di programmi Java
 - Come si scrive un programma

S. Mizzaro - Introduzione 2

Docenti

- Teoria
 - Stefano Mizzaro
 - 48 ore in aula (4 a settimana)
 - <http://www.dimi.uniud.it/mizzaro/>
 - mizzaro@uniud.it



S. Mizzaro - Introduzione 3

Docenti

- Laboratorio
 - 72 ore in lab. (6 a settimana)
 - Marco Pavan
 - 40h
 - marco.pavan@uniud.it
 - Eddy Maddalena
 - 20h
 - eddy.maddalena@uniud.it




S. Mizzaro - Introduzione 4

Il corso

- 120 ore, 12 CFU
- 48 ore di teoria, 6 CFU:
 - Mizzaro (48)
- 72 ore di laboratorio, 6 CFU:
 - 60h con assistenza/presenza del docente di lab.
 - 12h senza
- Orario lezioni
- Laboratorio
 - Niente lab. questa settimana, inizia la prossima!
 - Inizio: martedì 3 marzo

S. Mizzaro - Introduzione 5

Orario

		Lun	Mar	Mer	Gio	Ven
I	8:30-9:30	Prog (1)	Prog (5)			
	9:30-10:30					
II	10:30-11:30		Lab. (Lab_1)		Lab. (Lab_1)	
	11:30-12:30					
III	12:30-13:30					
	13:30-14:30			Lab. (Lab_1)		
IV	14:30-15:30					
	15:30-16:30					
V	16:30-17:30					
	17:30-18:30					

S. Mizzaro - Introduzione 6

Laboratorio

- Mappa
- Account
 - Fatelo *subito*!
 - Modulo da compilare??
 - In lab
 - Su Web:
 - http://www.uniud.it/didattica/facolta/scienze/info_dida/laboratorio/
 - Password
 - Foto tessera??
 - Regolamento! (responsabilità anche penali...)
- Altro account
 - <http://goo.gl/forms/ftjlwe4Kvi>

S. Mizzaro - Introduzione

7

Modalità d'esame preliminari (1/4)

- Scritto + autovalutazione + [progetto +] orale
- Scritto
 - Niente appunti, libri, ... allo scritto/provetta
 - Se voto scritto $\leq 10 \Rightarrow -5$ all'appello succ.!!
 - Scritto valido fino all'appello successivo (su richiesta!!)

S. Mizzaro - Introduzione

8

Modalità d'esame preliminari (2/4)

- Progetto (opzionale)
 - Gruppi di 3 persone (eccezioni **da concordare preventivamente**, ad es.: lavoro)
 - Il progetto va consegnato allo scritto e spedito per email (dettagli insieme al progetto)
 - Pubblicato sul web ca. 2 settimane prima dell'appello
 - Voto max. senza progetto: 27
 - Progetto dà incremento di 0-3 punti
 - Solo se voto scritto ≥ 24 !!
 - Se voto < 24 , non consegnato, non presentato \Rightarrow progetto annullato (come non fatto)

S. Mizzaro - Introduzione

9

Modalità d'esame preliminari (3/4)

- Orale
 - Su tutto il programma (e sul progetto)
 - "Consiglio": capite gli errori dello scritto...
 - Va sostenuto nello stesso appello del progetto, altrimenti bisogna rifare il progetto
 - Tutti i componenti del gruppo devono sapere TUTTO sul progetto come se l'avessero fatto da soli
 - Chi dimostra di non sapere abbastanza deve rifare scritto e progetto (chi non ha passato lo scritto...)

S. Mizzaro - Introduzione

10

Modalità d'esame preliminari (4/4)

- Autovalutazione (peer assessment)
 - In laboratorio: Esercitazioni e peer assessment
 - Il voto finale dipende dal voto di laboratorio
 - (Da -4 a +4)
 - Risolvere esercizi
 - Valutare le soluzioni degli altri
 - + esercizi risolvete e valutate, meglio è (per voi)
 - ... e meglio lo fate, meglio è...
 - Punteggio per risolutore e punteggio per valutatore

S. Mizzaro - Introduzione

11

Cose importanti (1/4)

- Studiare: Leggere, Capire, Memorizzare
- Organizzatevi: motivazioni, tempi, ...
- Studiate!
 - 198 ... 55, 142 ... 86, 130 ... 63, 130 ... 75
 - Tot.: 600 ... 280
 - Guardatevi... ☺
- Continuate a studiare
 - (140 ... 114; ... 80 ... 47)

S. Mizzaro - Introduzione

12

Cose importanti (2/4)

- Non basta studiare: bisogna anche **fare!**
 - Usate il calcolatore per provare i vs. programmi
 - Seguite le esercitazioni di laboratorio
 - Il laboratorio è *importante*! (6 CFU su 12)
 - Andate in laboratorio per conto vostro
 - Se avete un calcolatore...
- Fate domande! (anche per email)
 - Firmate i messaggi
 - Usate il vostro account universitario ("spes")
- Venite a ricevimento!
 - Orario: Mar. 10:30 – 12:30 (?)
 - Mappa

S. Mizzaro - Introduzione

13

Cose importanti (3/4)

- Bast-Che-Funz: chi se ne frega di come è scritto un programma, basta che funzioni...
- SE LA PENSATE COSÌ, NON PASSATE L'ESAME!
- DOVETE imparare a scrivere programmi di buona qualità
- Anche (soprattutto!) chi pensa di saper già programmare...

S. Mizzaro - Introduzione

14

Cose importanti (4/4)

- "Fidatevi" dei prof.
 - Se noi diciamo che una cosa è importante, lo è.
 - Se noi insistiamo su cose che vi sembrano inutili, turatevi il naso e studiatele
 - Forse abbiamo ragione noi... (ne ripariamo fra 5 anni)
 - E comunque l'esame lo fate con noi...
 - Se pensate che "queste cose qui" non servono a niente, cambiate idea (o corso di laurea...)
 - ...
- Imparare ≠ passare l'esame...

S. Mizzaro - Introduzione

15

Cose importanti (e ovvie?)

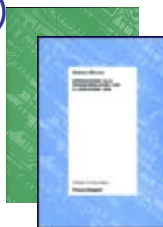
- Non siete più alla scuola dell'obbligo...
- Se siete qui, state pagando...
- Non buttate via i soldi...
- Non disturbate, e...
- ... "arrabbiatevi" con chi vi disturba

S. Mizzaro - Introduzione

16

Testi e materiale (1/2)

- S. Mizzaro, *Introduzione alla programmazione con il linguaggio Java*, Franco Angeli, Milano, 3a ed., 2001, ISBN 88-464-1696-1, <http://www.dimi.uniud.it/mizzaro/LibroJava>
- P. Coppola, S. Mizzaro, *Laboratorio di programmazione in Java*, Apogeo, Milano, 2004, ISBN: 88-503-2145-7



S. Mizzaro - Introduzione

Testi e materiale (2/2)

- Dispensa: V. Della Mea, S. Mizzaro. *L'algoritmo*, ... <http://www.dimi.uniud.it/mizzaro/research/papers/algoritmo.pdf>
- Blog (lucidi + avvisi + link + ...)
 - <http://programmazionetwudine.blogspot.com/>
 - Scorsi anni: <http://programmazionetwudine0910.blogspot.com/>
<http://programmazionetwudine0809.blogspot.com/>
- Gruppo Google (iscrivetevi!!)
 - <http://groups.google.com/group/programmazionetwudine/>
 - Scorsi anni: <http://groups.google.com/group/programmazionetwudine0910/>

S. Mizzaro - Introduzione

18

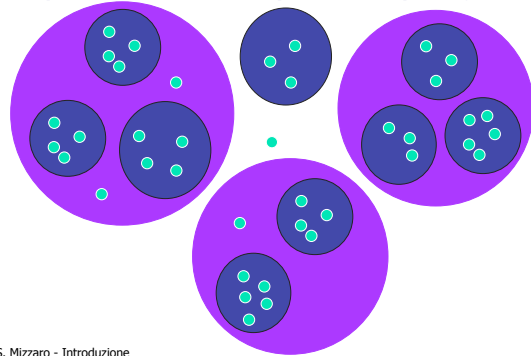
Programma preliminare del corso: obiettivi

- Introdurre gli aspetti fondamentali
 - della programmazione strutturata,
 - dell'occultamento delle informazioni,
 - della programmazione orientata agli oggetti
- Imparare a programmare
 - Usiamo il linguaggio Java
 - Java è il mezzo, non il fine!
 - "Imparare a ragionare"
- Accennare a computabilità e complessità

S. Mizzaro - Introduzione

19

Analisi sistematica: dai mattoni più piccoli alle componenti più grandi

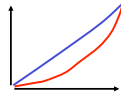


S. Mizzaro - Introduzione

20

Programma preliminare

1. La programmazione strutturata (22h)
2. Tipi di dati astratti e occultamento delle informazioni (4h)
3. I fondamenti della programmazione orientata agli oggetti (10h)
4. Le API del Java (10h)
5. Cenni alle teorie della computabilità e complessità (2h)



S. Mizzaro - Introduzione

21

1. La programmazione strutturata (22h)

- L'ABC della programmazione:
 - Espressioni
 - Strutture di controllo
 - Sviluppo incrementale
 - Array
 - Metodi
 - Ricorsione
- Presenti in ogni linguaggio di programmazione

S. Mizzaro - Introduzione

22

2. Tipi di dati astratti, occultamento delle informazioni (4h)

- Come fare programmi più grandi e complicati...
- ...e comunque modificabili, estendibili, comprensibili, ...
- Tipi di dati astratti, occultamento delle informazioni
- Concetti e realizzazione in Java
- Verso l'OO ("Object Oriented")

S. Mizzaro - Introduzione

23

3. I fondamenti della programmazione OO (10h)

- Evoluzione dai TDA: oggetti, classi, istanze, scambio messaggi
- Eredità
- Polimorfismo
- La programmazione OO in Java
 - Classi astratte
 - Interfacce
 - Package

S. Mizzaro - Introduzione

24

4. Le API del Java (10h)

- Application Programming Interface
- "Librerie"
- Il programmatore di oggi deve conoscere anche le API
- Introduzione, cenni:
 - File, stringhe, utilità, ...
 - Interfacce utente grafiche
 - Applet

S. Mizzaro - Introduzione

25

5. Teorie della computabilità e della complessità (cenni, 2h)

- Teoria matematiche (teoremi!)
- Computabilità
 - Cos'è un programma?
 - "Cosa può fare un programma"?
 - È possibile scrivere un programma che ...?
 - Ci sono linguaggi di programmazione più "potenti" di altri? Cosa vuol dire "più potente"?
- Complessità
 - Dato un problema, ci sono programmi più efficienti di altri per risolverlo? (più veloci, che usano meno risorse, ...)

S. Mizzaro - Introduzione

26

Collegamenti con altri corsi (Un po' con tutti. È IMP.!!)

- | | |
|--|--|
| <ul style="list-style-type: none"> ■ 1o anno <ul style="list-style-type: none"> ■ Architetture ■ Tecnologie Web ■ 2o anno <ul style="list-style-type: none"> ■ Programmazione OO ■ Algoritmi e strutture dati ■ Sistemi operativi ■ Sistemi multimediali ■ Tecnologie Web 2 | <ul style="list-style-type: none"> ■ 3o anno <ul style="list-style-type: none"> ■ Ingegneria del software ■ Interazione uomo macchina ■ Progetto siti Web ■ ... ■ Laurea magistrale <ul style="list-style-type: none"> ■ |
|--|--|

S. Mizzaro - Introduzione

27

Importanza

- Ci sono collegamenti con un po' tutti i corsi
- Programmerete per molto tempo. Forse per tutta la vostra vita lavorativa
- È un corso importante
 - (ce ne saranno altri)

S. Mizzaro - Introduzione

28

Riassunto

- Il corso
 - Docenti
 - Lezioni
 - Esame
 - Raccomandazioni
 - Programma preliminare del corso e testi
- Domande? Osservazioni?
- I primi esempi di programmi Java
- Come si scrive un programma

S. Mizzaro - Introduzione

29

Il nostro primo programma in Java

```
/* Il nostro primo programma in Java */
class CiaoATutti {
    public static void main (String[] args) {
        System.out.println("Ciao a tutti!");
    }
}
```

```
/* Il nostro primo programma in Java */
class CiaoATutti {
    public static vuoto principale(Stringa[] argomenti) {
        Sistema.uscita.stampalinea("Ciao a tutti!");
    }
}
```

S. Mizzaro - Introduzione

30

Il nostro primo programma in Java

```
/* Il nostro primo programma in Java */
class CiaoATutti {
    public static void main (String[] args) {
        System.out.println("Ciao a tutti!");
    }
}
```

- Prima riga: commento
- Due righe successive: intestazione (saremo più precisi), deve esserci sempre
 - "Nome del programma" (saremo più precisi)
- 4a riga: istruzione di "stampa"
- Poi le graffe chiuse, una per ogni graffa aperta

S. Mizzaro - Introduzione

31

L'attività del programmatore

- Ciclo editing-compilazione-esecuzione
- Editing
 - Usando un editor di testo, scrivere il programma e salvarlo su file
- Compilazione
 - Usando un compilatore, tradurre il programma da Java a un linguaggio più vicino all'HW
- Esecuzione
 - Eseguire il programma

S. Mizzaro - Introduzione

32

Editing-compilazione-esecuzione

- Editing
 - XEmacs, Blocco Note, ... (formato solo testo!!)
- Compilazione
 - Compilatore Java
 - Non produce codice direttamente eseguibile dall'HW (non produce linguaggio macchina)
 - Produce un "codice a byte" (bytecode)
 - `>javac NomeFile.java`
- Esecuzione
 - Interprete Java
 - Sa eseguire il bytecode
 - `>java NomeProgramma`

S. Mizzaro - Introduzione

33

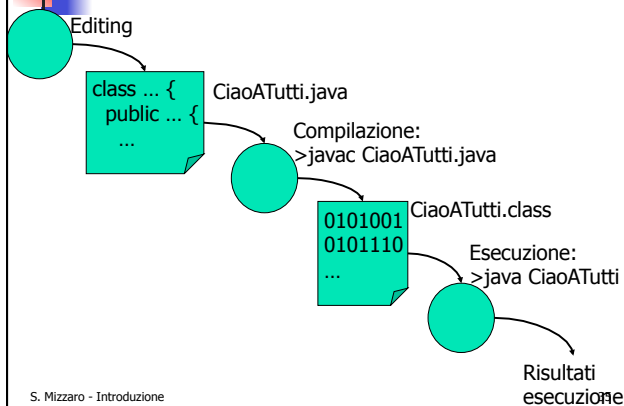
Il JDK

- JDK/SDK (Java/Standard Development Kit)
 - <http://www.java.com>
- La "cassetta degli attrezzi" del programmatore Java
- Compilatore: **javac**
- Interprete: **java**
 - JVM (Java Virtual Machine, Macchina virtuale Java)
 - JRE (Java Runtime Environment) = JVM + ...
- Installatelo sul vostro calcolatore
 - installare = scaricare + leggere le istruzioni + ...

S. Mizzaro - Introduzione

34

Schema



S. Mizzaro - Introduzione

Editing-compilazione-esecuzione in pratica

- Scrivo il programma **CiaoATutti** (uso l'editor XEmacs)
- Lo compilo
- Lo eseguo

```
>javac CiaoATutti.java
>java CiaoATutti
Ciao a tutti!
>
```

S. Mizzaro - Introduzione

36

Un programma più "serio"

```

/* 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);
    }
}
    
```

S. Mizzaro - Introduzione 37

Notare

- Parte "dichiarativa" e parte "eseguibile"
- Sequenza di istruzioni (;)
- Assegnamento (=)
- Variabili: dichiarazione e uso
 - area: variabile inutile
- Espressioni
- **System.out.println**
- Eseguiamo in pratica anche questo
 - E poi proviamolo su altri dati...

S. Mizzaro - Introduzione 38

La "leggibilità" (comprensibilità)

```

/* Programma per calcolare l'area di
   un triangolo */
class AreaTriangoloDisordinata { public
    static void main
    (
    String[] args) { int
    base; int altezza; int area; base =
    5; altezza = 12; area = base * altezza / 2;
    System.out.println(area);}}
    
```

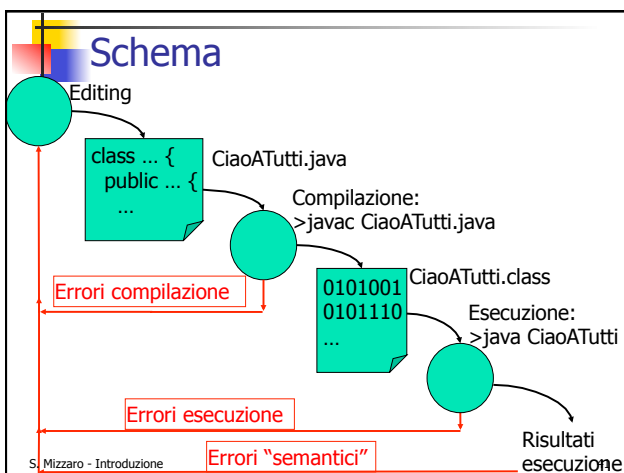
- È lo stesso programma? Fa le stesse cose?
- Morale: la leggibilità è importante!

S. Mizzaro - Introduzione 39

Errori

- Si fanno **sempre** errori quando si scrive un programma!
- Bisogna imparare a gestirli:
 - Capire i messaggi di errore del compilatore
 - Capire i messaggi di errore al tempo di esecuzione
 - Capire perché il programma non fa quello che vogliamo
- "Ciclo" editing-compilazione-esecuzione

S. Mizzaro - Introduzione 40



Un altro programma

```

/* 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");
        }
    }
}
    
```

S. Mizzaro - Introduzione 42

Osserviamo

- `System.in.read()`
 - throws `IOException`
 - `import java.io.*`
- `System.in.read()`
 - Legge un carattere (! **ACHTUNG!!!**)
 - ... **verrete a dirmi: "Prof., non funziona!"...**
- `if ... else ...`
 - Alternative
 - Dopo l'`if` ci va un'espressione logica (condizione)
 - "else" significa "altrimenti"

S. Mizzaro - Introduzione

43

Stampare i numeri da 1 a 10 (brutto!!)

```
class Stampal_10 {
    public static void main(String[] args) {
        System.out.println(1);
        System.out.println(2);
        System.out.println(3);
        System.out.println(4);
        System.out.println(5);
        System.out.println(6);
        System.out.println(7);
        System.out.println(8);
        System.out.println(9);
        System.out.println(10);
    }
}
```

S. Mizzaro - Introduzione

44

Stampare i numeri da 1 a 10 (meglio)

```
class Stampal_10 {
    public static void main(String[] args) {
        int contatore;
        contatore = 1;
        while (contatore <= 10) {
            System.out.println(contatore);
            contatore = contatore + 1;
        }
    }
}
```

- Questo è molto meglio. Perché:
 - Altrimenti non passate l'esame... ☺
 - ...stampare i numeri da 1 a 1000...

S. Mizzaro - Introduzione

45

Fine! Riassunto

- Prima parte
 - Il corso, i docenti, l'esame, raccomandazioni, ...
 - Programma preliminare del corso e testi
- Seconda parte
 - I primi esempi di programmi Java
 - Editing-compilazione-esecuzione
- Prossima lezione
 - Altri esempi di programmi
 - Analisi più sistematica
 - "Dai mattoni più piccoli alle componenti più grandi"

S. Mizzaro - Introduzione

46