



Alma Mater Studiorum-Università di Bologna

Scuola di Ingegneria

Fondamenti di Informatica T-2

Lab 01

Corso di Laurea in Ingegneria Informatica

Anno accademico 2021/2022

Prof. ROBERTA CALEGARI

Prof. AMBRA MOLESINI

Dipartimento di Informatica – Scienza e Ingegneria (DISI)



Agenda

- **PASSO 1**

Presentazione del tool da linea di comando

- **PASSO 2**

Compilazione ed esecuzione dei primi mini-esempi



Perché da linea di comando?

- In alcune situazioni potrebbe non essere disponibile un ambiente di sviluppo (vedi **Eclipse**)
- Un “I”**ngegnere** deve sempre uscire vittorioso, anche in condizioni avverse 😊



Primo Esempio - Implementazione

1. Creiamo un file con **Blocco Note** nominandolo

Esempio1.java

NB: il nome del file deve essere identico al nome della classe

```
/**
 * Applicazione Java da linea di comando.
 * Stampa la classica frase di benvenuto
 * @author Fondamenti di Informatica T-2
 * @version 1.0 02/2022
 */
public class Esempio1 {

    /**
     * Il main.
     * Stampa l'argomento passato da linea di comando
     * @param args Elenco dei parametri
     */
    public static void main(String[] args) {
        System.out.println(args[0]);
    }
}
```

Per evitare problemi con l'estensione del file, al momento del salvataggio scegliere salva come “Tutti i File” e inserire il nome completo (Esempio1.java): altrimenti, il file potrebbe essere salvato con l'estensione predefinita per i file di testo e dunque con nome **Esempio1.java.txt**



Primo Esempio - Compilazione

2. **Compiliamo** da linea di comando tramite il comando

javac Esempio1.java

javac compila il file **Esempio1.java**
e produce il file **Esempio1.class** (che contiene il **bytecode**)

```
OK Prompt fondamentali
27/01/2022 11:45 <DIR> .
27/01/2022 11:45 <DIR> ..
27/01/2022 11:45 <DIR> docs
27/01/2022 11:45 398 Esempio1.java
27/01/2022 11:45 390 Esempio2.java
27/01/2022 11:45 340 EsempioMath.java
27/01/2022 11:45 <DIR> EsempioMCD
27/01/2022 11:45 887 MyMain.java
27/01/2022 11:45 <DIR> MyMath
27/01/2022 11:45 4 File 2.015 byte
27/01/2022 11:45 5 Directory 300.400.070.656 byte disponibili

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi> javac Esempio1.java

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi> dir
Il volume nell'unità J è Windows
Numero di serie del volume: 1248-6BCC

Directory di J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi
03/02/2022 13:49 <DIR> .
27/01/2022 11:45 <DIR> ..
27/01/2022 11:45 <DIR> docs
03/02/2022 13:49 405 Esempio1.class
27/01/2022 11:45 398 Esempio1.java
27/01/2022 11:45 390 Esempio2.java
27/01/2022 11:45 340 EsempioMath.java
27/01/2022 11:45 <DIR> EsempioMCD
27/01/2022 11:45 887 MyMain.java
27/01/2022 11:45 <DIR> MyMath
27/01/2022 11:45 5 File 2.420 byte
27/01/2022 11:45 5 Directory 300.400.070.656 byte disponibili

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>
```



Sintassi del compilatore javac

javac [options] [sourcefiles]

Opzione	Descrizione
-help	Visualizza l'help
-verbose	Visualizza l'output completo di compilazione
-classpath classpath	Imposta il classpath, se non impostato recupera quello di default del sistema.
-sourcepath sourcepath	Permette di specificare la directory in cui cercare i sorgenti (.java). Se non impostato utilizza la directory corrente.
-d directory	Specifica una directory in cui salvare i bytecode (.class)
-g	Permette di generare le informazioni per il debugging
-deprecation	Avvisa ad ogni utilizzo di metodi deprecati
-nowrite	Controlla solo la sintassi senza compilare
-source [version]	Permette di compilare per una specifica versione di java (Es. 13).



Primo Esempio – Esecuzione

4. Eseguiamo l'applicazione creata con il comando

```
java Esempio1 Hello!!!!
```

Esegue il file `Esempio1.class` (con il main) interpretando il bytecode

Importante: non bisogna indicare l'estensione del file (`.class`)

```
J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi> java Esempio1 Hello!!!  
Hello!!!
```

```
J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>
```



Sintassi dell'interprete java

L'interprete esegue il programma Java sull'infrastruttura JRE, a partire:

- o dalla classe che contiene il main: **java classfile**
- o da un JAR che la contenga: **java -jar jarfile**

In generale, la sintassi dell'interprete è:

java [options] classfile

Opzione	Descrizione
-help	Visualizza l'help
-jar	Per eseguire un file jar (Eseguibile)
-classpath	Imposta il classpath, se non impostato recupera quello di default del sistema.
-verbose	Visualizza l'output completo
-version	Visualizza la versione di java attualmente installata



Primo Esempio - Documentazione

3. Generiamo la **documentazione** tramite comando

```
javadoc -d docs Esempio1.java
```

javadoc genera la documentazione associata al file sorgente **Esempio1.java** *se all'interno del codice vengono inseriti in maniera opportuna dei commenti*

È opportuno specificare la cartella di destinazione (-d *nome*) per evitare di «inquinare» con molti file la cartella di lavoro

Tool di base:

<https://docs.oracle.com/en/java/javase/15/>



Lo Strumento javadoc

- Un commento destinato ad essere interpretato da javadoc deve iniziare con **/**** (e terminare poi normalmente con ***/**)
 - può essere associato a **classi, metodi e campi dati**
 - è composto da una **descrizione** e da una serie di “**tag**” **opzionali** preceduti da **@**

```
/**
```

```
* Applicazione Java da linea di comando.
```

```
* Stampa la classica frase di benvenuto
```

```
* @author Fondamenti di Informatica T-2
```

```
* @version 1.0 02/2022
```

```
*/
```



Sintassi **javadoc**

javadoc [options] [packagenames] [sourcefiles]

Opzione	Descrizione
-help	Visualizza l'help
-d directory	Specifica una directory in cui creare la documentazione, il default è la directory corrente.
-version	Include i paragrafi @version
-author	Include i paragrafi @author

Esistono molte altre opzioni, per i dettagli si rimanda all'help specifico

javadoc -help



Sintassi **javadoc** – Tags

Opzione	Descrizione
@author [author]	Per indicare l'autore (Attenzione: è necessario indicare <code>-author</code> da linea di comando affinché questo tag venga considerato: vedere slide successive)
@deprecated [description]	Indica che il metodo/classe descritto è stato rimpiazzato e ne è sconsigliato l'utilizzo. Può seguire una descrizione su cosa utilizzare in alternativa.
@param [name] [description]	Descrive un parametro in ingresso di un metodo. (Es. <code>@param args</code> Argomenti passati in ingresso)
@returns [description]	Parametro di ritorno
...	

Esempio di generazione della documentazione

```
J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>javadoc -d docs Esempio1.java
Loading source file Esempio1.java...
Constructing Javadoc information...
Building index for all the packages and classes...
Standard Doclet version 17.0.2+8-86
Building tree for all the packages and classes...
Generating docs\Esempio1.html...
Generating docs\package-summary.html...
Generating docs\package-tree.html...
Generating docs\overview-tree.html...
Building index for all classes...
Generating docs\allclasses-index.html...
Generating docs\allpackages-index.html...
Generating docs\index-all.html...
Generating docs\index.html...
Generating docs\help-doc.html...

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>dir
Il volume nell'unità J è Windows
Numero di serie del volume: 1248-6BCC

Directory di J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi
03/02/2022 13:56 <DIR>      .
27/01/2022 11:45 <DIR>      docs
03/02/2022 13:59 <DIR>      .
03/02/2022 13:56      405 Esempio1.class
03/02/2022 13:56      398 Esempio1.java
27/01/2022 11:45      390 Esempio2.java
27/01/2022 11:45      340 EsempioMath.java
27/01/2022 11:45 <DIR>      EsempioMCD
27/01/2022 11:45      887 MyMain.java
27/01/2022 11:45 <DIR>      MyMath
                    5 File      2.420 byte
                    5 Directory 300.400.058.368 byte disponibili

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>
```

Documentazione
nella cartella docs

Esempio di generazione della documentazione

Documentazione
prodotta

```
03/02/2022 13:59 <DIR> .
03/02/2022 13:56 <DIR> ..
27/01/2022 11:45 1.233 allclasses-frame.html
03/02/2022 13:59 2.874 allclasses-index.html
27/01/2022 11:45 1.213 allclasses-noframe.html
27/01/2022 11:45 1.224 allclasses.html
03/02/2022 13:59 2.661 allpackages-index.html
27/01/2022 11:45 3.124 constant-values.html
27/01/2022 11:45 3.037 deprecated-list.html
03/02/2022 13:59 17 element-list
03/02/2022 13:59 10.195 Esempio1.html
03/02/2022 13:59 8.359 help-doc.html
03/02/2022 13:59 3.464 index-all.html
03/02/2022 13:59 962 index.html
27/01/2022 11:45 <DIR> jquery
03/02/2022 13:59 1.495 jquery-ui.overrides.css
03/02/2022 13:59 <DIR> legal
03/02/2022 13:59 198 member-search-index.js
27/01/2022 11:45 253 member-search-index.zip
03/02/2022 13:59 45 module-search-index.js
03/02/2022 13:59 2.652 overview-tree.html
27/01/2022 11:45 1.341 package-frame.html
27/01/2022 11:45 16 package-list
03/02/2022 13:59 95 package-search-index.js
27/01/2022 11:45 211 package-search-index.zip
03/02/2022 13:59 3.134 package-summary.html
```

PACKAGE CLASS TREE DEPRECATED INDEX HELP

ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

SEARCH:

Class Esempio1

java.lang.Object
Esempio1

```
public class Esempio1
extends java.lang.Object
```

Applicazione Java da linea di comando. Stampa la classica frase di benvenuto

Constructor Summary

Constructors	Description
Esempio1()	

Method Summary

All Methods	Static Methods	Concrete Methods
-------------	----------------	------------------



Secondo Esempio (1/2)

EsempioMath.java

NB: il nome del file deve essere identico al nome della classe

```
/**
 * Applicazione Java da linea di comando.
 * Stampa  $\sin(\pi/3)$ 
 * @author Fondamenti di Informatica T-2
 * @version 1.0 02/2022
 */
public class EsempioMath {
    /**
     * Il main.
     * Calcola e stampa  $\sin(\pi/3)$ 
     */
    public static void main(String args[]) {
        double x = Math.sin(Math.PI/3);
        System.out.println(x);
    }
}
```



Secondo Esempio (2/2)


Compilazione

```
javac EsempioMath.java
```

Viene generato il file `EsempioMath.class`

Esecuzione

```
java EsempioMath
```



Creare anche la
documentazione

```
D:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>javac EsempioMath.java
D:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>java EsempioMath
0.8660254037844386
D:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>
```




Terzo Esempio - Fattoriale (1/4)

Applicazione strutturata su **due funzioni** contenute in una classe

- Più avanti vedremo come distribuire le funzioni
 - su più classi
 - su più file
- in modo che ogni classe abbia le proprie *responsabilità*

Architettura

- La funzione **fact** effettua il calcolo
- Il **main** usa la funzione **fact** e ne verifica il funzionamento



Terzo Esempio - Fattoriale (2/4)

MyMain.java

```
/**
 * Applicazione Java da linea di comando.
 * Realizza un fattoriale ricorsivo
 * @author Fondamenti di Informatica T-2
 * @version 1.0 02/2022
 */
public class MyMain {
    /**
     * Calcola il fattoriale in modo ricorsivo
     */
    public static int fact(int n) {
        if(n==0)
            return 1;
        else
            return n * fact(n-1);
    }
}
// segue...
```

NB: il nome del file deve essere
identico al nome della classe

Oppure

```
return n==0 ? 1 : n*fact(n-1)
```



Terzo Esempio - Fattoriale (3/4)

```
// continua...
```

```
/**
```

```
* Il main - calcola e stampa fattoriali
```

```
*/
```

```
public static void main(String[] args) {
```

```
    int fact1 = fact(3); // RISULTATO ATTESO: 6
```

```
    System.out.println(fact1 == 6); // success or failure?
```

```
    int fact2 = fact(6); // RISULTATO ATTESO: 720
```

```
    System.out.println(fact2 == 720); // success or failure?
```

```
    int fact3 = fact(8); // RISULTATO ATTESO: 40320
```

```
    System.out.println(fact3 == 40320); // success or failure?
```

```
}
```

```
}
```




Terzo Esempio - Fattoriale (4/4)

Compilazione

```
javac MyMain.java
```

Viene generato il file *di bytecode* **MyMain.class**



Creare anche la documentazione

```
OK Selezione Prompt fondamentali
J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi> javac MyMain.java

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi> dir
Il volume nell'unità J è Windows
Numero di serie del volume: 1248-6BCC

Directory di J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi
03/02/2022  14:12    <DIR>          .
27/01/2022  11:45    <DIR>          ..
03/02/2022  13:59    <DIR>          docs
03/02/2022  13:56             405 Esempio1.class
03/02/2022  13:56             398 Esempio1.java
03/02/2022  14:08             390 Esempio2.java
03/02/2022  14:10             455 EsempioMath.class
03/02/2022  14:09             340 EsempioMath.java
27/01/2022  11:45    <DIR>          EsempioMCD
03/02/2022  14:12             705 MyMain.class
03/02/2022  14:09             887 MyMain.java
27/01/2022  11:45    <DIR>          MyMath
              7 File             3.580 byte
              5 Directory  300.399.947.776 byte disponibili

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>
```

Esecuzione

```
java MyMain
```

```
J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi> java MyMain
true
true
true

J:\Users\Administrator\Documents\CORSI\Fondamenti\Codice-Esami-Lab-svn\Lab\Lab01-MiniEsempi>
```



PRIMO STEP ESERCITAZIONE

Tempo a disposizione: 30 minuti



Quarto Esempio: applicazione su tre classi con nostra libreria

Innanzitutto, vogliamo sviluppare:

- una mini-libreria matematica che esponga una *funzione pubblica per il calcolo del Massimo Comun Divisore*
 - classe pubblica `MyMath`
 - funzione statica pubblica `int mcd(int a, int b)`
- un *cliente (main)* che la usi
 - classe pubblica `EsempioMath`
- un *mini-cliente ausiliario di test* che la collaudi
 - classe pubblica `MyMathTest`

Poi, vorremo anche:

- produrre un JAR eseguibile per *distribuire l'applicazione* così ottenuta (cliente + libreria) in modo unitario → `MyApp`
- con la relativa *documentazione*



Specifica del main

Requisiti

- il **cliente (main)** deve
 - recuperare due valori interi dalla linea di comando
 - calcolare l'MCD fra essi
 - stamparlo a video
- Come recuperare valori interi dalla linea di comando?
 - dalla linea di comando si recuperano stringhe
 - per convertire una stringa in `int` esiste una funzione (statica) di libreria: **`int Integer.parseInt(String s)`**



Implementazione del main

```
/**
 * Cliente che calcola mcd fra due numeri passati da riga di comando
 * utilizzando la libreria MyMath
 * @author Fondamenti di Informatica T-2
 * @version 1.0 02/2022
 */
public class EsempioMath {
    /** Calcola mcd dei due numeri passati da linea di comando
     */
    public static void main(String args[]) {
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);
        int mcd = MyMath.mcd(a,b);
        System.out.println(mcd);
    }
}
```

..e se uno degli
argomenti non
fosse un numero..?



La classe MyMath

```
public class MyMath {  
    public static int mcd(int a, int b) {  
        int resto;  
        if (b > a){ // swap a,b  
            int tmp = a;  
            a = b;  
            b = tmp;  
        }  
        do {  
            resto = a % b;  
            a = b;  
            b = resto;  
        } while (resto != 0);  
        return a;  
    }  
}
```

Banale algoritmo, identico a quello che avreste fatto in C..

..ma come collaudarlo?



Il mini-main di test

```
public class MyMathTest {  
    public static void main(String args[]) {  
        testOK();  
        // testKO();  
    }  
    private static void testOK() {  
        assert(MyMath.mcd(10, 5) == 5);  
        assert(MyMath.mcd(7, 3) == 1);  
        assert(MyMath.mcd(21, 49) == 7);  
    }  
    private static void testKO() {  
        assert(MyMath.mcd(10, 5) == 4);  
    }  
}
```

PIANO DI COLLAUDO!

Caso corretto

Controprova
(farà esplodere il main!)

Ricorda: sarà operativo solo se invocato con **-ea**



Quarto Esempio – Jar

Applicazione **MyApp** costituita da:

- `MyMath.class`
- `EsempioMath.class` (*con il main*)

File .class già compilati
(solo da zippare)

CREAZIONE FILE JAR ESEGUIBILE

```
jar cmf info.txt MyApp.jar MyMath.class EsempioMath.class
```

oppure, se si trovano tutti e soli nella stessa cartella:

```
jar cmf info.txt MyApp.jar *.class
```

Contiene la riga:

Main-Class: EsempioMath

Ricordatevi di lasciare una riga vuota
dopo altrimenti non funziona ☺

TEST

```
java -jar MyApp.jar
```



MCM

- Aggiungere alla libreria matematica **MyMath** una *funzione pubblica* per il calcolo del Minimo Comune Multiplo
 - classe pubblica **MyMath**
 - funzione statica pubblica `int mcm(int a, int b)`
- un *cliente (main)* che la usi
 - classe pubblica **EsempioMathMCM**
- un *mini-cliente ausiliario di test* che la collaudi
 - classe pubblica **MyMathTestMCM**



SECONDO STEP ESERCITAZIONE

Tempo a disposizione: 45 minuti