UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Fondamenti di Informatica (117IN)

A.A. 2022 / 2023

Lezione 3 - Introduzione a Java

Sylvio Barbon Junior
sylvio.barbonjunior@units.it

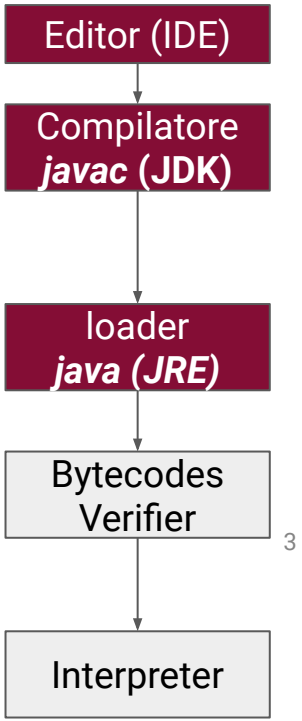
Sommario:

1) Ambienti;
2) Linguaggi di programmazione;
3) La soluzione di problemi per computer;
4) Algoritmo e programma.
5) Esempio Java
6) I paradigmi di Programmazione
7) I linguaggi imperativi
   a) Esempio somma
8) Esempio di errore in Java
9) Pratica di Java

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

1) Ambienti

Java il linguaggio portabile



1 **myCode.java**

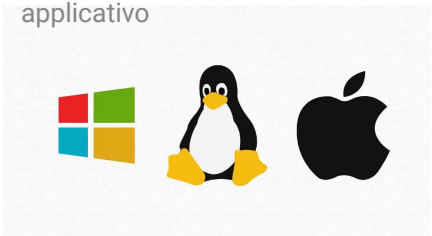
Source Code - codice con il programma

2 **myCode.class**

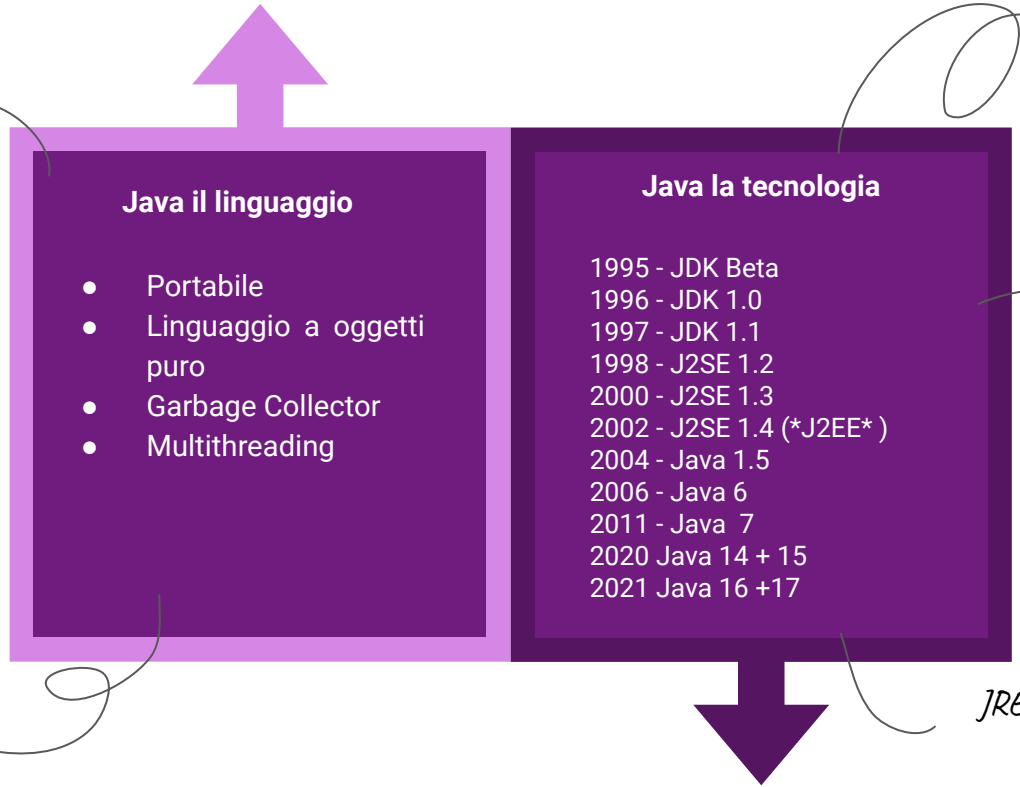
Codice compilato ("pseudo-compilato" bytecode), senza necessità di ricompilazioni, è eseguibile su ogni tipo di piattaforma hardware-software.

3 **My Application**

Varie Sistemi Operativi con il stesso applicativo



1) Ambienti



Garbage Collector fa la gestione della memoria, per motivi di sicurezza e di semplicità di programmazione.

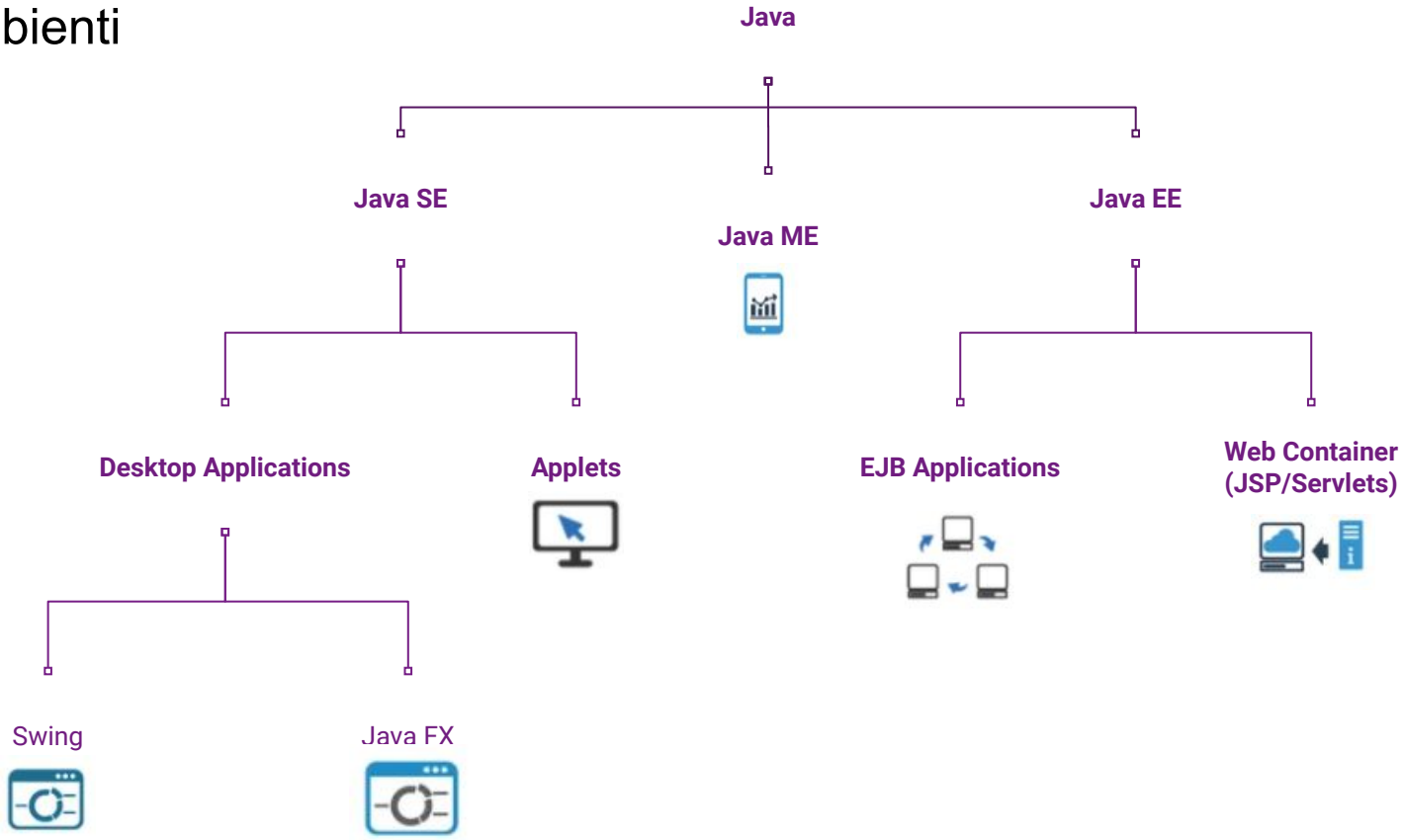
Multithreading che fornisce all'interno della stessa applicazione, eseguire contemporaneamente più task.

JDK = Java Development Kit, un set completo di API e una serie di tool per lo sviluppo di applicazioni Java.

JSE = Java Standard Edition, pacchetto base. Abbiamo anche JEE (Java Enterprise Edition) e JME (Java Mobile Edition)

JRE= Java Runtime Environment, to run Java codes (Java Virtual Machine)

1) Ambienti


Java
Java SE
Java ME
Java EE
Desktop Applications
Applets
EJB Applications
Web Container
(JSP/Servlets)
Swing
Java FX

2) Linguaggi di programmazione

I linguaggi di programmazione si possono suddividere in due grandi categorie:

1. linguaggi a basso livello (linguaggi macchina e Assembler);
2. linguaggi ad alto livello o linguaggi evoluti;

il computer è in grado di comprendere!

permettere l'uso di nomi simbolici che corrispondono a più istruzioni in linguaggio macchina

Abbiamo i software di utilità:

programma sorgente è scritto nel linguaggio di programmazione!

- **L'editor**: è un programma di utilità che consente di inserire e memorizzare un testo e di modificarlo successivamente per effettuare aggiunte o correzioni (esempio: Sublime).

- I **programmi traduttori:** tradurre un programma (programma sorgente) codificato in un linguaggio di programmazione viene tradotto in linguaggio macchina;

compilatori o interpreti fa la traduzione dei linguaggi ad alto livello per basso livello.

2) Linguaggi di programmazione

Il Compilatore:

É un programma che traduce un programma sorgente scritto in linguaggio ad alto livello in un programma oggetto in linguaggio macchina.

Il **compilatore** individua e segnala tutti gli errori formali. Se ci sono errori la traduzione non può avvenire.

Se non ci sono errori il compilatore passa alla fase di sintesi durante la e genera il **codice oggetto**, di solito in formato rilocabile

*1) analisi lessicale, che individua le **parole** del vocabolario e costruisce la tabella dei simboli;*

*2) analisi sintattica, che controlla se **le frasi** sono scritte rispettando la sintassi del linguaggio cioè le regole della grammatica*

*3) analisi semantica, che controlla la validità del **significato** degli elementi in base al contesto (per esempio la presenza delle istruzioni dichiarative necessarie).*

Il Linker:
trasforma il programma oggetto in programma eseguibile.

Portabilità

L'interprete:
tradurre le istruzioni del linguaggio ad alto livello in linguaggio
macchina, una per una, al momento dell'esecuzione.

2) Linguaggi di programmazione

Il Compilatore:

- ↑ maggior velocità di esecuzione;
- ↑ risparmio di memoria;
- ↑ rileva tutti gli errori formali;
- ↑ consente la segretezza del programma sorgente
- ↑ non è necessario codice aggiuntivo
- ↓ tempi di creazione del programma più lunghi
- ↓ minore portabilità

Java è parzialmente compilato e interpretato!

L'interprete:

- ↑ semplicità di messa a punto dei programmi poiché si lavora in ambiente interattivo
- ↑ maggior portabilità dei programmi
- ↓ maggiore occupazione di memoria
- ↓ l'esecuzione risulta più lenta
- ↓ non dà garanzia di correttezza sintattica
- ↓ non consente la segretezza del codice sorgente
- ↓ è necessario avere l'interprete

3) La soluzione di problemi per computer

Problema = qualsiasi proposta per cui è necessario trovare una soluzione.

questione:

** la situazione iniziale (quali sono i dati del problema);
** che cosa si desidera ottenere (risultati)
** le risorse a disposizione;

I dati che riguardano il problema sono tutte le informazioni disponibili all'inizio e quelle che si desiderano ottenere come soluzione del problema.

Le azioni che possono essere eseguite nel processo risolutivo sono le operazioni che il computer è in grado di eseguire. >> Algoritmo

mobile vs grosso computer?

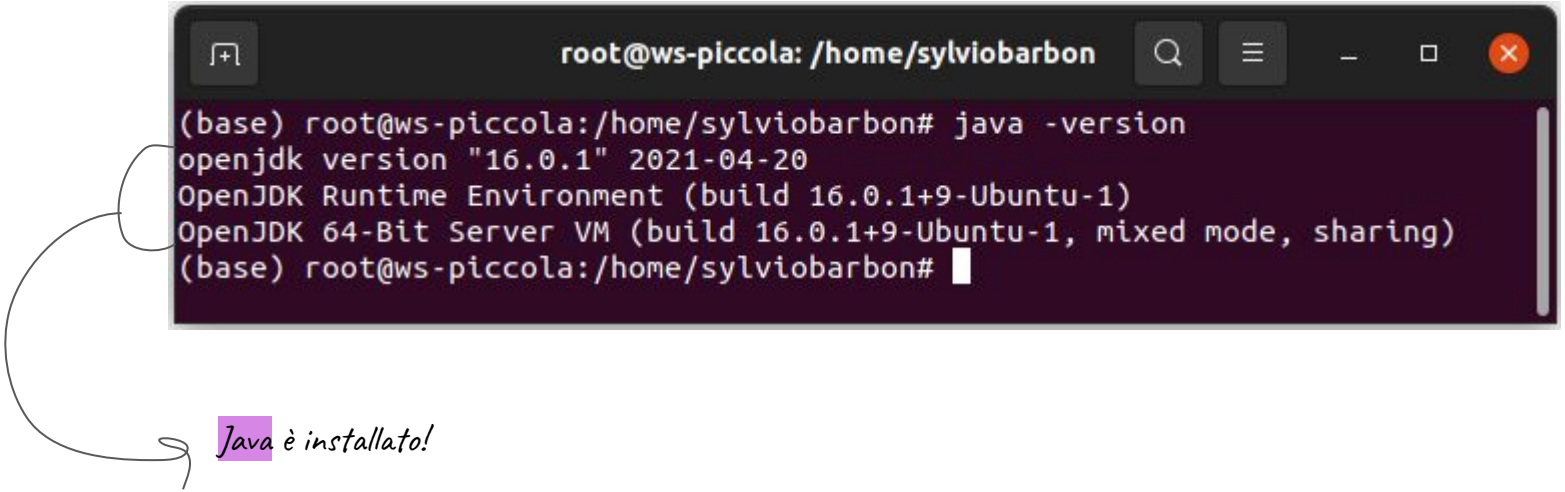
4) Algoritmo e Programma

L'algoritmo è un procedimento che permette di ottenere dei risultati (dati in uscita o di output) partendo da alcuni dati iniziali (dati di ingresso o di input).
L'algoritmo deve:

- essere **generale**: non deve risolvere un singolo caso particolare, ma tutta una serie di problemi dello stesso tipo
- essere **deterministico**: i risultati devono essere sempre gli stessi, partendo dagli stessi dati iniziali, cioè non devono dipendere da qualcosa di casuale;

Il **programma** si ottiene codificando l'algoritmo in un **linguaggio di programmazione**, cioè scrivendo le istruzioni dell'algoritmo secondo la sintassi del linguaggio scelto.

5) Esempio Java



5) Esempio Java

Java1.java

```
1  public class Java1{
2      public static void main(String[] args){
3          System.out.println("My Java programm!");
4      }
5  }
```

Il mio primo programma!

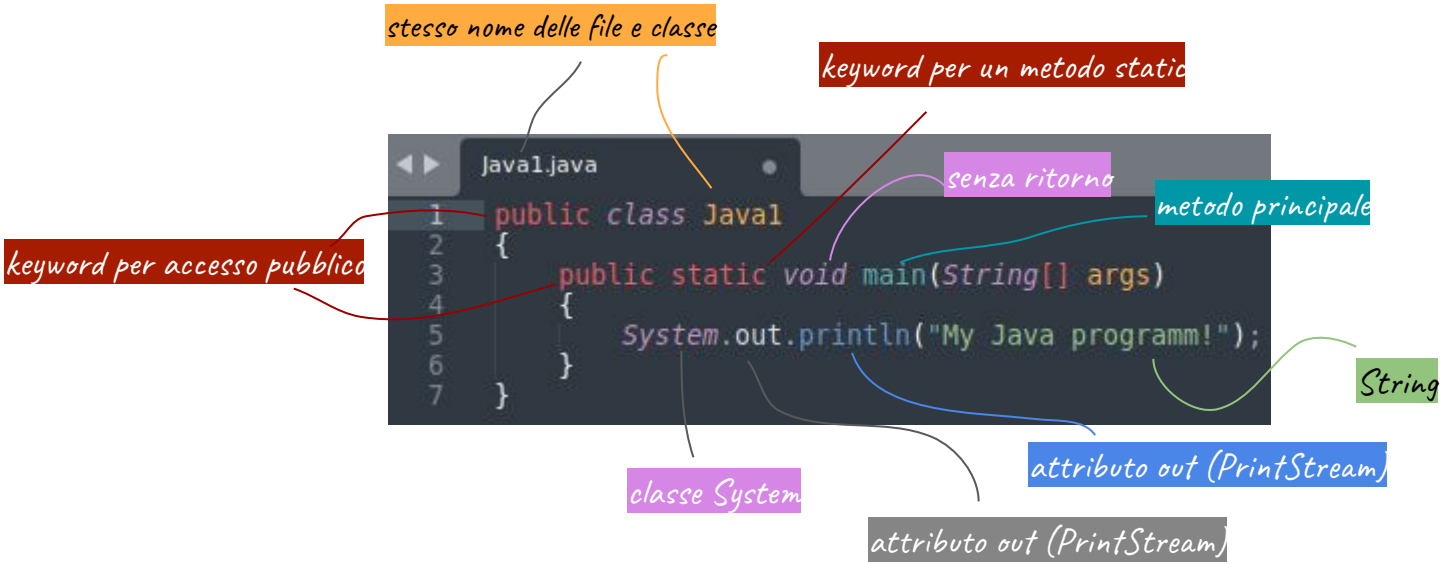
```
root@ws-piccola: /home/sylviobarbon/Scaricati/Fi/CodiciSorgenti
(base) root@ws-piccola:/home/sylviobarbon/Scaricati/Fi/CodiciSorgenti# javac Java1.java
(base) root@ws-piccola:/home/sylviobarbon/Scaricati/Fi/CodiciSorgenti# ls
Java1.class  Java1.java
(base) root@ws-piccola:/home/sylviobarbon/Scaricati/Fi/CodiciSorgenti# java Java1
My Java programm!
(base) root@ws-piccola:/home/sylviobarbon/Scaricati/Fi/CodiciSorgenti#
```

compilazione!

creazione dell .class

esecuzione

5) Esempio Java



5) Esempio Java

SalveMondo.java

```java
public class SalveMondo {

    public SalveMondo (){
        this.print();
    }

    public void print(){
        System.out.println("Ciao mondo");
    }

    public static void main(String[] args){
        SalveMondo sm = new SalveMondo();
    }
}
```

```
sylviobarbon@ws-piccola: ~/Scaricati/Fi/CodiciSorgenti
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$ javac SalveMondo.java
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$ java SalveMondo
Ciao mondo
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

6) I paradigmi di Programmazione

Computabilità

- Un problema si dice computabile se la sua soluzione può essere descritta mediante un algoritmo.
- I problemi che non sono risolvibili con un procedimento algoritmico si dicono non computabili.

La descrizione dei dati e delle azioni per risolvere il problema dipende **dal paradigma di programmazione** utilizzato e per linguaggi che usano lo stesso paradigma, almeno parzialmente dal linguaggio scelto, infatti potrebbero essere disponibili per esempio tipi o strutture di dati diverse.

Un **paradigma di programmazione** è un modello che permette di descrivere astrattamente l'algoritmo (cioè il metodo di soluzione di un problema).

I principali paradigmi di programmazione sono:

- il paradigma dichiarativo;
- il paradigma procedurale;

descrive come deve essere risolto un problema.

*descrive **che cosa** si vuole ottenere*

6) I paradigmi di Programmazione

La programmazione con il **paradigma dichiarativo** si può suddividere in:

1. **programmazione logica:** descrive i fatti e le relazioni tra i fatti e permette di ricavarne delle conseguenze;[PROLOG]

2. **programmazione funzionale**: il programma è costituito da un insieme di funzioni che elaborano liste di simboli; [LISP]

3. linguaggi **di markup:** usano dei codici (tag) per stabilire il ruolo degli elementi; [HTML]

4. linguaggi di **interrogazione di database**. [SQL]

7) I paradigmi di Programmazione

La programmazione con il **paradigma procedurale** si può suddividere in:

1. **programmazione imperativa**: si basa su esplicite richieste fatte all'esecutore del programma; la soluzione del problema è descritta come una sequenza di azioni che producono dei cambiamenti di stato nell'ambiente (come la modifica del valore delle variabili);

   Assembler, pascal, C, Cobol

2. **programmazione orientata agli oggetti:** parte dal concetto di oggetto che descrive proprietà e azioni che un oggetto può compiere; la soluzione del problema è descritta dalle interazioni tra gli oggetti.

   Java, C++, C#, Python

8) I linguaggi imperativi

Il vocabolario usato da ciascun linguaggio comprende delle **parole riservate** e delle parole che possono **essere definite dal programmatore** seguendo alcune semplici regole.

- Le parole **riservate** permettono di definire i dati e di scrivere le istruzioni;
- Le parole **definite** dal programmatore (o identificatori) sono in genere i nomi delle variabili e i nomi delle procedure da richiamare.

SalveMondo.java

```
1  public class SalveMondo {
2
3      public SalveMondo (){
4          this.print();
5      }
6
7      public void print(){
8          System.out.println("Ciao mondo");
9      }
10
11     public static void main(String[] args){
12         SalveMondo sm = new SalveMondo();
13     }
14 }
```

8) I linguaggi imperativi

Esempio di Somma

Andiamo a creare un codice Java per fare una somma di due numeri costanti (3+4).

Somma.java

```
1  public class Somma{
2
3      public static void main(String args[]){
4          int firstNum;
5          int secondNum;
6          int sum;
7
8          firstNum = 3;
9          secondNum = 4;
10
11         sum = firstNum + secondNum;
12         System.out.println(Integer.toString(sum));
13     }
14 }
```

```
sylviobarbon@ws-piccola:~/Scaricati/FI/CodiciSorgenti$ java Somma
7
```

Sintassi di dichiarazione di una variabile

[modificatore] tipo_di_dato nome_della_variabile **[=inizializzazione];**

molto importante per che riguarda l'uso de memoria.

deve far senso e aiutare la lettura del codice. Utilizzare il standard camelCase

9) Esempio di errore in Java

cannot find symbol

Il compilatore non sai che cos'è Ciao:

- *Variabile*
- *String*

Errore.java

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

```
sylvtoberton@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$ javac Errore.java
Errore.java:4: error: cannot find symbol
            System.out.println(Ciao);
                               ^
  symbol:   variable Ciao
  location: class Errore
1 error
```

9) Esempio di errore in Java

; expected

- *Manca il punto e virgola!*

Errore.java

```
1  public class Errore{
2
3      public static void main(String[] args){
4          System.out.println("Ciao")
5      }
6
7  }
```

```
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$ javac Errore.java
Errore.java:4: error: ';' expected
            System.out.println("Ciao")
                                      ^
1 error
```

9) Esempio di errore in Java

exception in thread “main”

- *ArithmeticException, dovuta ad una **divisione per zero***

Errore.java

```
1  public class Errore{
2
3      public static void main(String[] args){
4
5          int number = 0;
6
7          System.out.println(number/number);
8      }
9
10 }
```

```
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$ javac Errore.java
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$ java Errore
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Errore.main(Errore.java:7)
sylviobarbon@ws-piccola:~/Scaricati/Fi/CodiciSorgenti$
```

10) Tipi di errore

Non ti preoccupare, è normale avere errore:

1. **Durante la compilazione:** errore di sintassi del linguaggio;
2. **Runtime errore:** dividendo an integer by zero, accessing an element that is out of range of the array, trying to store a value into an array that is not compatible type, ecc;
3. **Errore logiche:** difficile da trovare!

11) Pratica di Java

1. Crea un programma che stampa sul display a 'Hello World';
2. Crea un programma che stampa sul display somma di quattro numeri diversi;
3. Crea un programma che stampa sul display sottrazione di un numero per la sua metà;
4. Crea un programma che stampa sul display un diamante come nell'immagine: