

# Fondamenti di Java



Fac.Scienze - Università di Trento

Static  
I/O

# Fondamenti di Java



Fac.Scienze - Università di Trento

Static



# Modificatori: static

**Variabili e metodi associati ad una Classe anziché ad un Oggetto sono definiti “static”.**

Le variabili statiche servono come singola variabile **condivisa** tra le varie istanze

I metodi possono essere richiamati **senza creare una istanza**.



## Variabili “static”: esempio 1

```
class S {  
    static int instanceCount = 0; //variabile "di classe"  
    S() {instanceCount++;}  
}  
  
public class A {  
    public static void main(String a[]) {  
        new A();  
    }  
    A() {  
        for (int i = 0; i < 10; ++i) {  
            S instance=new S();  
        }  
        System.out.println("# of instances: "+S.instanceCount);  
    }  
}
```

Output:  
# of instances: 10



## Variabili "static": esempio 2

```
class S {
    static int instanceCount = 0; //variabile "di classe"
    S() {instanceCount++;}
    public void finalize() {instanceCount--;}
}

public class A {
    public static void main(String a[]) {
        new A();
    }
    A() {
        for (int i = 0; i < 10; ++i) {
            S instance=new S();
        }
        System.out.println("# of instances: "+S.instanceCount);
        System.gc();
        System.out.println("# of instances: "+S.instanceCount);
    }
}
```

Output:

# of instances: 10

# of instances: 0



## Metodi "static": esempio 1

```
class S {
    static int instanceCount = 0; //variabile "di classe"
    S() {instanceCount++;}
    static void azzerContatore() {instanceCount=0;}
}

public class A {
    public static void main(String a[]) {
        new A();
    }
    A() {
        for (int i = 0; i < 10; ++i) {
            if (i%4==0) S.azzerContatore();
            S instance=new S();
        }
        System.out.println("instanceCount: "+S.instanceCount);
    }
}
```

Può agire solo su  
variabili statiche!

Output:

instanceCount: 2

Ruolo:  
Metodi che agiscono su  
variabili statiche



## metodi "static": esempio 2

Notare la  
maiuscola!  
(per convenzione)

```
Math.sqrt();
System.gc();
System.arraycopy(...);
System.exit();
Integer.parseInt(String s);
Float.parseFloat(String s);
```

Ruolo:  
analogo alle  
librerie del C

Che cos'è:  
`System.out.println()` ?



## Perchè il main è "static"?

```
public class A {
    String s="hello";
    public static void main(String a[]) {
        System.out.println(s);
    }
}
```

Non static variable s cannot be referenced from static context

```
public class A {
    String s="hello";
    public static void main(String a[]) {
        new A();
    }
    A() {
        System.out.println(s);
    }
}
```

hello



## Sezione: I/O

Fac.Scienze - Università di Trento

### Input/Output



## Lettura di stringhe

```
import java.io.*;
public class A {
    public A() {
        BufferedReader s = new BufferedReader(
            new InputStreamReader(System.in));
        try {
            System.out.println("Dammi una stringa");
            String str=s.readLine();
            System.out.println("Hai scritto "+str);
        } catch (IOException e) {e.printStackTrace();}
    }
    public static void main(String [] ar) {
        A a=new A();
    }
}
```

Dammi una stringa  
abracadabra  
Hai scritto abracadabra



## Lettura di int

```
public A() {  
    int i=0;  
    BufferedReader s = new BufferedReader(  
        new InputStreamReader(System.in));  
    try {  
        System.out.println("Dammi un intero");  
        i=Integer.parseInt(s.readLine());  
        System.out.println("Hai scritto "+i);  
    }catch (Exception e) {e.printStackTrace();}  
}
```

```
Dammi un intero  
2  
Hai scritto 2
```



```
public A() {  
    int i=0;  
    BufferedReader s = new BufferedReader(  
        new InputStreamReader(System.in));  
    try {  
        System.out.println("Dammi un intero");  
        i=Integer.parseInt(s.readLine());  
        System.out.println("Hai scritto "+i);  
    }catch (IOException e) {e.printStackTrace();}  
}
```

```
Dammi un intero  
pippo  
java.lang.NumberFormatException: For input string: "gh"  
    at  
    java.lang.NumberFormatException.forInputString(NumberFormatException.java:48)  
    at java.lang.Integer.parseInt(Integer.java:426)  
    at java.lang.Integer.valueOf(Integer.java:532)  
    at pila.A.<init>(A.java:11)  
    at pila.A.main(A.java:19)  
Exception in thread "main"
```

```

1 public A() {
2     float f=0; boolean error;
3     BufferedReader s = new BufferedReader(
4         new InputStreamReader(System.in));
5     try {
6         do {
7             System.out.println("Dammi un float");
8             try{
9                 error=false;
10                f=Float.parseFloat(s.readLine());
11            } catch (NumberFormatException e) {
12                error=true;
13                System.out.println("Input non valido");
14            }
15        } while (error);
16        System.out.println("Hai scritto "+f);
17    }catch (IOException e) {e.printStackTrace();}
18 }

```

```


Dammi un float
pippo
Input non valido
Dammi un float
3
Hai scritto 3.0

```

14

Programmazione 2 - Marco Ronchetti

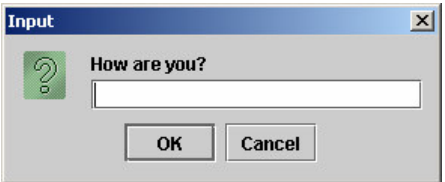
## Lettura di stringhe con GUI



```

import javax.swing.JOptionPane;
public A() {
    ...
    String input = JOptionPane.showInputDialog(
        "How are you?");
    System.out.println(input);
    System.exit(1);
}

```



**Essenziale!**  
 Altrimenti la thread che gestisce la GUI rimane viva, e il processo non termina

Fac.Scienze - Università di Trento