

# Scopri l'OOP in Java

Secondo incontro, 21/01/2025

Prof.sse A. Schiavon, L. Dalla Montà, gennaio 2025

# Scopri l'OOP in Java

## Lezione 2, 21/01/2025

- La documentazione Java API (Application Programming Interface)
  - Uso della documentazione Java 8.0 (o superiori)
  - <https://docs.oracle.com/javase/8/docs/api>
- Classi dell'API di Java, uso delle classi più comuni
  - Object
  - String
  - Random
  - Rectangle

# Classe Object - uno sguardo alla documentazione API

java.lang

## Class Object

java.lang.Object

public class **Object**

Class Object is the root of the class hierarchy. Every class has Object as a superclass. All objects, including arrays, implement the

Since:  
JDK1.0

Object

+ Object( ) *metodo speciale -> costruttore*

# clone( ): Object

+ equals(obj: Object): boolean

...

+ hashCode( )

...

+ toString( ): String

...

### Constructors

#### Constructor and Description

Object()

### Method Summary

#### All Methods   Instance Methods   Concrete Methods

Modifier and Type	Method and Description
protected <b>Object</b>	<b>clone()</b> Creates and returns a copy of this object.
boolean	<b>equals(Object obj)</b> Indicates whether some other object is "equal to" this one.
protected void	<b>finalize()</b> Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
<b>Class&lt;?&gt;</b>	<b>getClass()</b> Returns the runtime class of this Object.
int	<b>hashCode()</b> Returns a hash code value for the object.
void	<b>notify()</b> Wakes up a single thread that is waiting on this object's monitor.
void	<b>notifyAll()</b> Wakes up all threads that are waiting on this object's monitor.
<b>String</b>	<b>toString()</b> Returns a string representation of the object.
void	<b>wait()</b> Causes the current thread to wait until another thread invokes the <b>notify()</b> method or the <b>notifyAll()</b> method for this object.
void	<b>wait(long timeout)</b> Causes the current thread to wait until either another thread invokes the <b>notify()</b> method or the <b>notifyAll()</b> method for this object, or a specified amount of time has elapsed.
void	<b>wait(long timeout, int nanos)</b> Causes the current thread to wait until another thread invokes the <b>notify()</b> method or the <b>notifyAll()</b> method for this object, or some other thread interrupts the current thread, or a certain amount of real time has elapsed.

# Classe Object

```
public class ObjectTester{  
    public static void main(String[] args){  
        Object oggetto=new Object(); // istanziamento dell'oggetto  
        int codiceHash=oggetto.hashCode(); // invocazione del metodo di istanza, messaggio,  
        interazione con la classe Object  
        System.out.println(codiceHash);  
        System.out.println(oggetto.toString());  
    }  
}
```

## Attività:

- Istanziare un secondo oggetto e utilizzare gli stessi metodi
- Copiare nel secondo oggetto il riferimento del primo e verificare che il codice hash e la rappresentazione “stringa” dell’oggetto cambia.

# Classe String - uno sguardo alla documentazione API

java.lang

## Class String

java.lang.Object

java.lang.String

### Constructor Summary

#### Constructors

##### Constructor and Description

###### **String()**

Initializes a newly created String object so that it represents an empty character sequence.

###### **String(byte[] bytes)**

Constructs a new String by decoding the specified array of bytes using the platform's default charset.

###### **String(byte[] bytes, Charset charset)**

###### **String(char[] value)**

Allocates a new String so that it represents the sequence of characters currently contained in the character array argument.

###### **String(char[] value, int offset, int count)**

Allocates a new String that contains characters from a subarray of the character array argument.

###### **String(int[] codePoints, int offset, int count)**

Allocates a new String that contains characters from a subarray of the **Unicode code point** array argument.

###### **String(String original)**

Initializes a newly created String object so that it represents the same sequence of characters as the argument; in other string.

# Classe String - uno sguardo alla documentazione API

## Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods	Deprecated Methods
Modifier and Type		Method and Description		
char		<b>charAt</b> (int index) Returns the char value at the specified index.		
int		<b>codePointAt</b> (int index) Returns the character (Unicode code point) at the specified index.		
int		<b>codePointBefore</b> (int index)		

.....

int	<b>compareTo</b> (String anotherString) Compares two strings lexicographically.
int	<b>compareToIgnoreCase</b> (String str) Compares two strings lexicographically, ignoring case differences.
<b>String</b>	<b>concat</b> (String str) Concatenates the specified string to the end of this string.
boolean	<b>contains</b> (CharSequence s) Returns true if and only if this string contains the specified sequence of characters.
boolean	<b>contentEquals</b> (CharSequence cs)

# Classe String

```
public class StringTester{  
    public static void main(String[] args){  
        String s1 = new String("Siamo al secondo incontro");  
        // equivalente a String s1="Siamo al secondo incontro";  
        System.out.println(s1.hashCode());  
        System.out.println(s1.toString());  
    }  
}
```

Attività: cercare nella documentazione API i metodi della classe String per:

- Calcolare la lunghezza della stringa
- Estrarre il primo carattere della stringa
- Rimuovere tutti gli spazi contenuti nella stringa
- Sostituire tutti gli spazi con il carattere - (trattino)
- Convertire tutti i caratteri della stringa in caratteri maiuscoli / minuscoli
- Estrarre la sottostringa compresa tra il quinto e il decimo carattere

Attenzione: conservare la stringa originale ed invocare i metodi suddetti su una copia della stringa originale.



# Classe Random - uno sguardo alla documentazione API

java.util

## Class Random

java.lang.Object

java.util.Random

**All Implemented Interfaces:**

Serializable

**Direct Known Subclasses:**

SecureRandom, ThreadLocalRandom

### Constructor Summary

#### Constructors

##### Constructor and Description

###### Random()

Creates a new random number generator.

###### Random(long seed)

Creates a new random number generator using a single long seed.

boolean	<b>nextBoolean()</b> Returns the next pseudorandom, uniformly distributed boolean value from this random number generator's sequence.
void	<b>nextBytes(byte[] bytes)</b> Generates random bytes and places them into a user-supplied byte array.
double	<b>nextDouble()</b> Returns the next pseudorandom, uniformly distributed double value between 0.0 and 1.0 from this random number generator's sequence.
float	<b>nextFloat()</b> Returns the next pseudorandom, uniformly distributed float value between 0.0 and 1.0 from this random number generator's sequence.
double	<b>nextGaussian()</b> Returns the next pseudorandom, Gaussian ("normally") distributed double value with mean 0.0 and standard deviation 1.0 from this random number generator's sequence.
int	<b>nextInt()</b> Returns the next pseudorandom, uniformly distributed int value from this random number generator's sequence.
int	<b>nextInt(int bound)</b> Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.
long	<b>nextLong()</b> Returns the next pseudorandom, uniformly distributed long value from this random number generator's sequence.



# Classe Random

```
import java.util.Random;
public class RandomTester{
    public static void main(String[] args){
        Random r1 = new Random();
        System.out.println(r1.hashCode());
        System.out.println(r1.toString());

        // estrazione di un valore intero (reale, booleano, ...) casuale
        int numeroCasuale=r1.nextInt();
        System.out.println(numeroCasuale);

        // estrazione di un secondo nuemro casuale
        System.out.println(r1.nextInt());
    }
}
```

Considera l'esempio in figura: a tuo parere, quali sono i valori possibili per il numero casuale estratto? Perché?

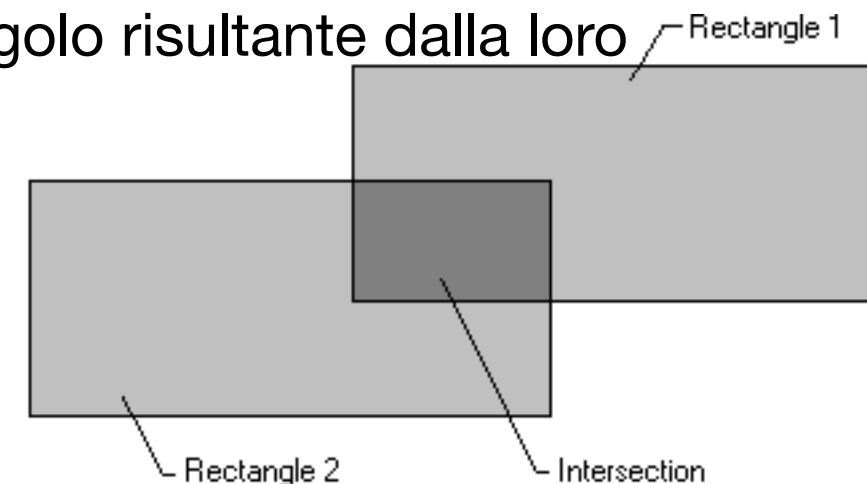
Attività: cercare nella documentazione API i metodi della classe Random per:

- Estrarre un numero casuale intero compreso tra -5 e +5
- Estrarre 5 valori booleani
- Estrarre 10 numeri reali compresi tra 0.0 e 1.0
- Simulare il gioco del lancio di un dado:
  - la vittoria c'è se al lancio si ottiene il numero 6
  - se dopo 5 lanci non è mai uscito il 6, la partita è persa.

# Scopri le classi Rectangle e Point

**Consultare la documentazione API (package java.awt) e, scegliendo i metodi corretti, svolgere le seguenti attività.**

- Realizzare l'UML della classe (sintesi)
- Realizzare una classe tester, per:
  - ✓ Istanziare un oggetto Rectangle
  - ✓ Istanziare tre oggetti Rectangle, i cui vertici in alto a sinistra corrispondano a:
    - le coordinate (0,0), con base e altezza 0
    - un oggetto Point(10,20), con base e altezza 0
    - il punto di coordinate x=5 y=5, con base 15 e altezza 5
  - ✓ Scegliere uno degli oggetti Rectangle con dimensione 0 e aumentarne base e altezza, con valori a scelta
  - ✓ Scegliere due degli oggetti Rectangle e restituire il rettangolo risultante dalla loro intersezione



- ✓ Dopo ciascuna azione eseguita, per ogni oggetto sopracitato, visualizzarne lo stato.