



package

utilizzo classi

- Java contiene molte classi predefinite che sono raggruppate in *categorie di classi* e prendono il nome di *package*
- l'insieme di questi package è chiamato Java *API* (*Java Application Programming Interface*)
- per specificare le classi necessarie per compilare un programma Java viene utilizzata la dichiarazione import
 - **import <package>.<Classe>;**
- *esempio* per utilizzare la classe Scanner del package java.util si usa la seguente dichiarazione
 - **import java.util.Scanner;**
- per utilizzare tutte le classi del package java.util si usa la dichiarazione
 - **import java.util.*;**

- il package è un meccanismo per organizzare classi in *gruppi logici* con l'obiettivo principale di definire namespace distinti per diversi contesti
- il package ha lo scopo di riunire classi *logicamente correlate*
- esempio:
 - nel package `java.awt` fra le altre troviamo le classi `Point`, `Rectangle` ecc

<https://docs.oracle.com/javase/10/docs/api/java/awt/Point.html>

- importare *tutto* il package:

```
import java.awt.*;  
class Prova {  
    Point p;  
}
```

- importare *solo* la classe:

```
import java.awt.Point;  
class Prova {  
    Point p;  
}
```

- usare il nome di package prima della classe:

```
class Prova {  
    java.awt.Point p;  
}
```

- tutte le classi del package devono stare nella stessa cartella
- ogni file inizia con la dichiarazione:
package nomepackage;
- convenzione sui nomi dei package:
 - per evitare di definire package con lo stesso nome si può utilizzare il dominio di primo livello dell'organizzazione che lo realizza, seguito dal dominio e da altri eventuali sottodomini, elencati in ordine inverso
 - i nomi dei package dovrebbero contenere solo lettere minuscole.

```
/**
 * figure geometriche
 */
package it.unipr.ferrari.alberto.geometria;
/**
 * @author alberto.ferrari
 * metodi comuni alle figure geometriche
 */
public interface Figura {
    /**
     * perimetro
     * @return misura del perimetro
     */
    public double perimetro();

    /**
     * area
     * @return misura dell'area
     */
    public double area();
}
```



esempio geometria classe base

A. Ferrari

```
package it.unipr.ferrari.alberto.geometria;

public class Rettangolo implements Figura {

    private double larghezza;
    private double altezza;

    public Rettangolo(double larghezza, double altezza) {
        super();
        this.larghezza = larghezza;
        this.altezza = altezza;
    }

    ...

    public double perimetro() {
        return 2*(larghezza+altezza);
    }

    public double area() {
        return larghezza*altezza;
    }
}
```

```
/**
 * figure geometriche
 */
package it.unipr.ferrari.alberto.geometria;

/**
 * @author alberto.ferrari
 * quadrato
 */
public class Quadrato extends Rettangolo {

    public Quadrato(double lato) {
        super(lato, lato);
    }

}
```



```
/**
 * applicazioni di esempio
 */
package it.univr.ferrari.alberto.applicazioni;

import it.univr.ferrari.alberto.geometria.Figura;
import it.univr.ferrari.alberto.geometria.Quadrato;
import it.univr.ferrari.alberto.geometria.Rettangolo;

/**
 * @author alberto.ferrari
 * esempio utilizzo package
 */
public class EsempioPackage {
    public static void main(String[] args) {
        Figura f = new Rettangolo(10,20);
        System.out.println(f.perimetro());
        Quadrato q = new Quadrato(30);
        System.out.println(q.area());
    }
}
```

esempio

PACKAGE TIME



- il package `java.time` fornisce classi per la gestione delle date e degli orari (a partire dalla versione 1.8)
- `LocalDate`
 - <https://docs.oracle.com/javase/10/docs/api/java/time/LocalDate.html>
- `LocalTime`
 - <https://docs.oracle.com/javase/10/docs/api/java/time/LocalTime.html>
- `LocalDateTime`
 - <https://docs.oracle.com/javase/10/docs/api/java/time/LocalDateTime.html>

```
import java.time.LocalDate;
...
// Data odierna
LocalDate oggi = LocalDate.now();
System.out.println("oggi è " + oggi);
System.out.println("anno " + oggi.getYear());
System.out.println("mese " + oggi.getMonth());
System.out.println("numero mese " + oggi.getMonthValue());
System.out.println("giorno del mese " + oggi.getDayOfMonth());
System.out.println("day of week " + oggi.getDayOfWeek());
// Giorno della settimana
String giornoSettimana = oggi.getDayOfWeek().getDisplayName(
    TextStyle.FULL, Locale.getDefault());
System.out.println("giorno della settimana " + giornoSettimana);
```

```
oggi è 2019-02-03
anno 2019
mese FEBRUARY
numero mese 2
giorno del mese 3
day of week SUNDAY
giorno della settimana domenica
```

```
// Creazione di un oggetto data
LocalDate natale = LocalDate.of(2019, 12, 25);
System.out.println(natale);

// Formattazione data in italiano
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");
String s = natale.format(formatter);
System.out.println(s);

// Differenza fra due date
Period p = Period.between(natale, oggi);
System.out.println("Differenza tra " + oggi + " e " + natale +
    " = " + p.getDays() + " giorni e " + p.getMonths() + " mesi");
```

2019-12-25

25-12-2019

Differenza tra 2019-02-03 e 2019-12-25 = -22 giorni e -10 mesi

```
// ora attuale
LocalTime adesso = LocalTime.now();
System.out.println("adesso sono le " + adesso);
System.out.println("ore " + adesso.getHour());
System.out.println("minuti " + adesso.getMinute());
System.out.println("fra mezz'ora sono le " + adesso.plusMinutes(30));

// mezzogiorno
LocalTime mezzogiorno = LocalTime.of(12, 00);
System.out.println("mezzogiorno " + mezzogiorno);
```

```
adesso sono le 11:30:06.285575400
ore 11
minuti 30
fra mezz'ora sono le 12:00:06.285575400
mezzogiorno 12:00
```

```
// Data ora attuali  
LocalDateTime questoMomento = LocalDateTime.now();  
System.out.println(questoMomento);
```

```
2019-02-03T11:30:06.285575400
```