

# *Classi e Oggetti*

cetic 



# Oggetti

---

- *Un oggetto rappresenta un entità reale o astratta ed è caratterizzato da:*
  - **stato** : *definisce le proprietà*
  - **Comportamento**: *definisce le azioni*

# Classi

---

*Una classe qualifica la natura (stato e metodi) degli oggetti dello stesso genere*

*Le **classi**  
sono utilizzate per modellare  
gli **oggetti***

# Classi

---

- *E' possibile definire una classe per modellare oggetti specifici*
- *Ad esempio: è possibile scrivere la classe **Automobile** per creare il modello di una **Automobile***

# I metodi

---

- **I metodi** stabiliscono i comportamenti dell'oggetto e definiscono il codice che verrà eseguito quando vengono invocati
- Quando un metodo viene invocato, il flusso di esecuzione passa al metodo e viene eseguito il relativo codice
- Una volta che l'esecuzione del metodo è completata, il flusso ritorna al punto da cui è partita la chiamata
- In relazione alla progettazione della classe, alcuni metodi possono restituire un (solo) valore, che viene restituito al codice chiamante

# La progettazione di una classe

- **STATO**
  - proprietà
- **METODI**
  - **Costruttore:** per costruire le istanze
  - **metodi:** per compiere delle azioni

# Una classe in Java

---

```
class Automobile{  
    // proprietà  
  
    // costruttore  
  
    // metodi  
  
}
```

# Una classe in Java

---

*in pseudo codice*

```
class Automobile{  
  // proprietà  
  marca  
  modello  
  litriCarburante;  
  kmPercorsi;  
  
  // costruttore  
  Automobile (marca, modello, litri, km)  
  
  // metodi  
  percorri(km)  
}
```



# Una classe in Java: i tipi delle variabili

```
class Automobile{  
    // proprietà  
    String      marca;  
    String      modello;  
    int         litriCarburante;  
    double      kmPercorsi;  
  
    // costruttore  
    Automobile (String marca, String modello, int litri, double  
    km) {}  
  
    // metodi  
    percorri(int km) {}  
}
```

# Una classe in Java: il costruttore

---

```
class Automobile{
    // proprietà
    String      marca;
    String      modello;
    int         litriCarburante;
    double      kmPercorsi;

    // costruttore
    Automobile (String marca, String modello, int litri, double km)    {
        this.marca = marca;
        this.modello = modello;
        this.litriCarburante = litri;
        this.kmPercorsi = km;
    }

    Automobile (double km)      {
        this.kmPercorsi = km;
    }
}
```

# Una classe in Java: i metodi

---

```
class Automobile{  
    // proprietà  
    String      marca;  
    String      modello;  
    int         litriCarburante;  
    double      kmPercorsi;  
  
    void percorri(double km) {  
        this.kmPercorsi = this.kmPercorsi + km;  
    }  
  
    double getKm() {  
        return this.kmPercorsi;  
    }  
}
```

# Le istanze

---

- *Ogni oggetto istanziato dalla classe **Automobile** contiene una "copia" delle variabili che definiscono lo stato dell'oggetto.*
- *Tutte le istanze hanno gli stessi metodi ma variabili di istanza diverse.*

# Le istanze

---

*Dichiarazione ed assegnazione di una variabile di tipo primitivo:*

***int t = 11 ;***

*Dichiarazione ed assegnazione di un oggetto di tipo Automobile:*

***Automobile nomevar = new Automobile (213);***

↑  
*tipo*

↑  
*nome della  
variabile*

↑  
*costruttore*

# Le istanze

---

```
class esempio{  
  
    public static void main(String pippo[]){  
  
        Automobile autoDaniele = new Automobile (54);  
        Automobile autoRobert = new Automobile (100);  
  
        autoDaniele.marca = "Ferrari";  
        autoDaniele.percorri(12);  
        System.out.println( autoDaniele.getKm() );  
  
    }  
}
```

# Modificatori di visibilità

---

- *3 modificatori di visibilità:*
  - **public**
  - **private**
  - **Protected**
- *I membri di una classe dichiarati **public** sono accessibili da chiunque*
- *I membri di una classe dichiarati **private** sono accessibili solo dall'interno della classe stessa*
- *I membri di una classe dichiarati **protected** sono accessibili solo dall'interno della classe e delle sotto-classi*

# Modificatori di visibilità

---

*Alcune best-practice:*

- *Dichiarare le variabili di stato come **private***
- *Dichiarare i metodi che dovranno essere accessibili come **public**, tutti gli altri metodi (metodi di supporto) come **private***



# Una classe in Java: il costruttore

---

```
public class Automobile{
    // proprietà
    private String      marca;
    private String      modello;
    private int         litriCarburante;
    private double      kmPercorsi;

    public Automobile (double km)      {
        this.kmPercorsi = km;
    }

    public void percorri(double km) {
        this.kmPercorsi = this.kmPercorsi + km;
    }
}
```

# L'API di J2SE

---

*L'ambiente di sviluppo Java si chiama (attualmente) Java 2 Software Development Kit Standard Edition (J2SE SDK, o anche JDK)*

*Il JDK include un ampio insieme di classi predefinite (attualmente sono oltre 2700) organizzate in librerie (chiamate package: attualmente sono più di 130)*

*I package organizzano applicativamente le classi: per esempio, ci sono i package per la gestione dell'interfaccia utente grafica, della comunicazione in rete tra applicazioni, dell'accesso a database, ...*

*Per usare nelle proprie applicazioni le classi contenute in un package è sufficiente "importare" nell'applicazione la classe oppure l'intero package*

# API JDK

*Supponiamo di voler creare una semplice applicazione che visualizzi un numero casuale ogni volta che viene eseguita.*

*Esplorando la documentazione dell'API scopriamo che il package **util** contiene una classe **Random***

```
import java.util.Random;
```

```
class NumeroCasuale{
```

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

```
        Random casuale = new Random();
```

```
        System.out.print( "numero casuale: " + casuale.nextDouble() );
```

```
    }
```

```
}
```