

# Programmazione ad Oggetti

## Modificatori e Incapsulamento (Information Hiding)

---

A.A. 2022/2023

---

*Docente: Prof. Salvatore D'Angelo*  
*Email: [salvatore.dangelo@unicampania.it](mailto:salvatore.dangelo@unicampania.it)*



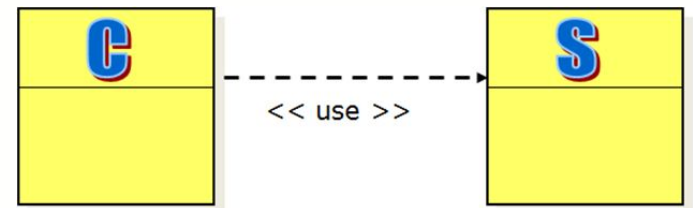
Università  
degli Studi  
della Campania  
*Luigi Vanvitelli*

*Dipartimento di Ingegneria*

# Modificatori

Una Classe può giocare un duplice ruolo:

- Ruolo "**Cliente**" : cioè può utilizzare le risorse messe a disposizione da altre Classi.
- Ruolo "**Servente**" : cioè essere usata da altre Classi o da un Programma Utente.



Esempio: La classe Automobile

- Interfaccia pubblica:
  - dà accesso a “ciò che l'auto può fare”
    - volante
    - blocchetto di accensione
    - pedale dell'acceleratore
    - ...
- Implementazione privata:
  - specifica “come l'auto fa ciò che può fare”
    - meccanica dello sterzo
    - elettromeccanica dell'avviamento
    - sistema di alimentazione e accensione
    - ...

# Modificatori

## La specifica di una classe

- Rappresenta un'interfaccia per la classe stessa in cui sono descritte:
  - le risorse messe a disposizione ai suoi potenziali utenti
  - le regole sintattiche per il loro utilizzo.
- E' separata dalla implementazione, permette l'utilizzo senza che l'utente conosca i dettagli di implementazione.
- E a cura dello sviluppatore della classe.

# Modificatori

SPECIFICA DELLA CLASSE

```
Public class C {  
    //public:  
    //prototipi delle funzioni membro  
        public T1 f1 (...);  
        public T2 f2 (...);  
        .....  
    //private:  
    //struttura dati  
        private int i;  
        private char c;  
        .....  
}; //fine specifica della classe C
```

# Modificatori: public

```
public class Customer {  
    public String name;  
    public String phone;  
    public String address;  
  
    public void setName (String customerName) {  
        name = customerName.toUpperCase ( );  
    }  
}
```

```
//...  
Customer cust = new Customer ( );  
//...  
cust.setName ("John Phillips");  
//...  
//...  
cust.name = "Jack Phillips";
```

# Modificatori: private

```
public class Customer {  
    private String name;  
    private String phone;  
    private String address;  
  
    public void setName (String customerName) {  
        name = customerName.toUpperCase ( );  
    }  
}
```

```
//...  
Customer cust = new Customer ( );  
//...  
cust.setName ("John Phillips");  
//...  
//...  
cust.name = "Jack Phillips";
```



# I Metodi Get e Set

```
public class PrivateShirt1Test {  
    public static void main (String args[ ]) {  
        PrivateShirt1 privShirt = new PrivateShirt1( );  
        char colorCode;  
  
        privShirt.setColorCode('R');  
        colorCode = privShirt.getColorCode( );  
  
        System.out.println("Color Code: " + colorCode);  
  
        privShirt.setColorCode('Z');  
        colorCode = privShirt.getColorCode( );  
        System.out.println("Color Code: " + colorCode);  
    }  
}
```

```
public class PrivateShirt1 {  
    private int shirtID = 0;  
    //...  
    private int quantityInStock = 0;  
    public char getColorCode( ) {  
        return colorCode;  
    }  
    public void setColorCode(char newCode) {  
        switch (newCode) {  
            case 'R':  
            case 'G':  
            case 'B':  
            case 'U':  
                colorCode = newCode;  
                break;  
            default:  
                System.out.println("Invalid Color Code");  
        }  
    }  
}
```