



# OOP

INCAPSULATION

# Cosa si intende per “incapsulamento”?

- ▶ Per “Incapsulamento” si intende la proprietà che hanno gli oggetti di incorporare al loro interno sia gli attributi che i metodi (cioè caratteristiche e comportamento della classe);
- ▶ Si crea come una barriera concettuale che isola l'oggetto dalle cose esterne, in questo modo tutte le informazioni utili che riguardano l'oggetto sono incapsulate nell'oggetto
- ▶ I metodi **public setXXX()** and **getXXX()** sono punti di accesso alle variabili dell'oggetto.
- ▶ Questi metodi sono anche chiamati getters and setters. Se una classe A vuole accedere ai le variabili private di una classe B utilizzerà i getters and setters della classe B

# Incapsulation: Esempio

```
public class Incapsulation{
    private String nome;
    private String idNum;
    private int age;

    public int getAge(){
        return age;
    } //getAge

    public String getName(){
        return name;
    } //getName

    public String getIdNum(){
        return idNum;
    } //getIdNum

    public void setAge( int newAge){
        age = newAge;
    } //setAge

    public void setName(String newName){
        name = newName;
    } //setName

    public void setIdNum( String newId){
        idNum = newId;
    } //setIdNum
} //class
```

```
/* File name :
Incapsulation.java */
```

```
public String getName(){
    return name;
} //getName

public void setName(String newName){
    name = newName;
} //setName
```

# Incapsulation: Esempio

```
/* File name : Contatto.java */
public class RunIncapsulation{

    public static void main(String args[]){
        Incapsulation contatto = new Incapsulation();

        contatto.setName («Ciccio»);
        contatto.setAge (20);
        contatto.setIdNum ("45698");

        System.out.print("Name : " + contatto.getName() + " Age : " +
            contatto.getAge());
    } //main
} //class
```

## Benefits of Encapsulation:

- ▶ Gli attribute di una classe possono essere read-only or write-only.
- ▶ Una classe puo' avere il controllo totale di cio'che contiene nei suoi attributi.
- ▶ Chi usera' la classe non conosce i dati memorizzati nella classe. Una classe puo'cambiare il tipo di dato e gli utenti della classe non hanno bisogno di cambiare il loro codice.