



Proprietà

Definizione

Una classe può contenere anche un altro tipo di metodi, le cosiddette proprietà.

Si tratta di particolari funzioni che permettono di leggere e/o impostare i valori di determinate proprietà di una classe.

Al posto di utilizzare una variabile pubblica, viene proposto l'uso di una variabile privata (scritta in minuscolo con l'aggiunta dell'underscore).

Le proprietà sono degli accessor (getter e/o setter).

Le proprietà garantiscono l'incapsulamento

Definizione d'incapsulamento

L'incapsulamento (information hiding) è la chiave della programmazione orientata agli oggetti.

Tramite esso, una classe riesce ad acquisire caratteristiche di robustezza, indipendenza e riusabilità. Inoltre la sua manutenzione risulterà più semplice al programmatore.

Di seguito sono elencate le caratteristiche dell'incapsulamento:

- Ispirato dalla realtà: tutti gli oggetti del mondo reale sono incapsulati
- Un oggetto deve possedere un **interfaccia pubblica** (che serve per l'interazione dell'oggetto con l'esterno)
- Un oggetto deve possedere **una parte nascosta** (invisibile dall'esterno dell'oggetto).

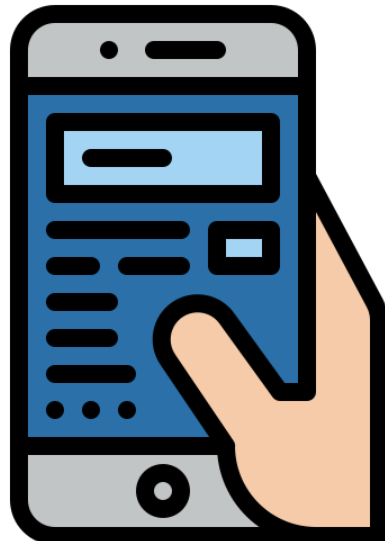
Esempio

Un semplice esempio è costituito da un oggetto frequentemente adoperato: lo smartphone.

La maggior parte degli utenti sa utilizzare uno smartphone; ignorando il funzionamento interno.

Chiunque può scegliere un contatto dalla rubrica, comporre un numero telefonico, chattare con gli amici e così via, ma pochi conoscono in dettaglio la sequenza dei processi scatenati da quei semplici tocchi sul touchscreen.

Per utilizzare uno smartphone, non è necessario essere un programmatore: basta saper usare la sua interfaccia pubblica e non la sua implementazione interna.



Implementazione

Una qualsiasi classe è costituita da **dati** e **metodi**.

La filosofia dell'incapsulamento si basa sull'accesso controllato ai dati mediante metodi che possono prevenire l'utilizzo da classi non autorizzate ed evitare eventuali errori.

Per applicare l'incapsulamento ad una classe occorre:

- Dichiarare privati gli attributi al fine di renderli inaccessibili fuori dalla classe stessa. A tale scopo utilizzando il modificatore ***private***.
- Creare dei metodi pubblici(dichiarati con il modificatore public) che permettono l'accesso agli attributi in maniera controllata. **Tali metodi potrebbero realizzare controlli prima di confermare l'accesso ai dati privati, evitando che essi assumano valori non validi.**

Sintassi

```
#region =02=== membri & proprietà =====
```

```
private double _peso;
```

```
public double Peso
```

```
{
```

```
    get { return _peso; }
```

```
    set { _peso = value; }
```

```
}
```

```
private double _altezza;
```

```
public double Altezza
```

```
{
```

```
    get { return _altezza; }
```

```
    set
```

```
    {
```

```
        if (value < 0)
```

```
            value = 1;
```

```
        _altezza = value;
```

```
    }
```

```
}
```

```
#endregion
```

Esercizio

Crea un applicativo che chiede all'utente di inserire 4 Clown e successivamente stampare a video le loro informazioni.