

Corso di Informatica

Modulo T1

B1– Il concetto di classe

M. Malatesta B1-Il concetto di classe-09

1
24/09/2011

Prerequisiti

- Programmazione elementare
- Conoscenza dell'uso delle funzioni
- Utilizzo della UML

M. Malatesta B1-Il concetto di classe-09

2
24/09/2011

Introduzione

Gli oggetti reali *sono una infinità* e noi non saremmo mai in grado di gestirli, se non procedessimo a *raggruppare tutti gli oggetti con le stesse caratteristiche*. Questo raggruppamento prende il nome di **classificazione**.



Allora ci si chiede:

si possono raggruppare oggetti software simili?

M. Malatesta B1-II concetto di classe-09

3
24/09/2011

Il concetto di classe

Tutti gli oggetti simili, ossia con *stessi attributi* e *stessi metodi*, possiamo immaginarli come scaturiti da un *modello generale* che chiamiamo **classe**.

Una classe:

- è un **modello astratto** che rappresenta un insieme di oggetti aventi caratteristiche comuni (es. schedaTelefonica, studente, televisore,...)
- contiene le **caratteristiche comuni** (attributi e metodi) degli oggetti da essa rappresentati.

Gli oggetti sono tutte le istanze (concrete) di una classe, aventi in comune le proprietà e i metodi di essa.

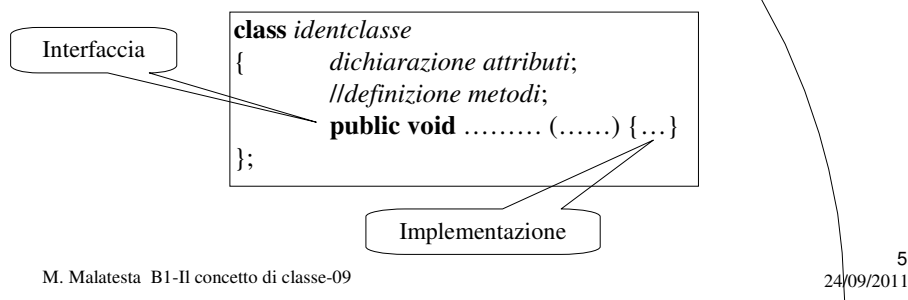
M. Malatesta B1-II concetto di classe-09

4
24/09/2011

Il concetto di classe

La **classe** contiene:

- **cosa** le singole istanze (gli **oggetti**) possono fare dal punto di vista utente (**interfaccia**);
- **come** lo possono fare, attraverso attributi e la definizione dei metodi (**implementazione**).



M. Malatesta B1-II concetto di classe-09

Implementazione di classi

La **dichiarazione** di una **classe** ha le seguenti caratteristiche:

- contiene la dichiarazione degli attributi (le proprietà)
- contiene la definizione dei metodi (le operazioni)
- trattandosi di una dichiarazione la classe **NON** occupa memoria
- è un modello, la cui istanza è l'oggetto che, essendo considerato come una variabile, occupa effettivamente memoria.
- consente di usare l'oggetto creato senza conoscerne l'implementazione, ma solo le firme dei metodi, grazie all'astrazione procedurale

M. Malatesta B1-II concetto di classe-09

6
24/09/2011

Implementazione di classi

Ad esempio, la classe *Autoveicolo* può essere definita come:

public class Autoveicolo

```
{  
  private String marca;  
  String modello;  
  String colore;  
  Intero porte;  
  Reale velocita;  
}
```

Gli attributi in genere si dichiarano **privati** per motivi di sicurezza

I metodi in genere si dichiarano **pubblici** per consentirne l'uso all'esterno della classe

```
public void setMotore (Intero stato) {...}  
public void setVelocita (Intero v) {...}  
public void giraDestra () {...}  
public void giraSinistra () {...}  
public Intero getVelocita () {...}  
}
```

Implementazione dei metodi

Le parole **public** e **private** prendono il nome di **specificatori di accesso**.

M. Malatesta B1-II concetto di classe-09

7
24/09/2011

Diagramma della classe

Mediante la **UML**, si può tracciare il **diagramma della classe** che prevede uno schema del tipo indicato a fianco, in cui il progettista inserisce:

- **nome della classe**
- **dichiarazioni degli attributi**
- **dichiarazione dei metodi**, ossia i **prototipi** (interfacce con eventuali parametri)

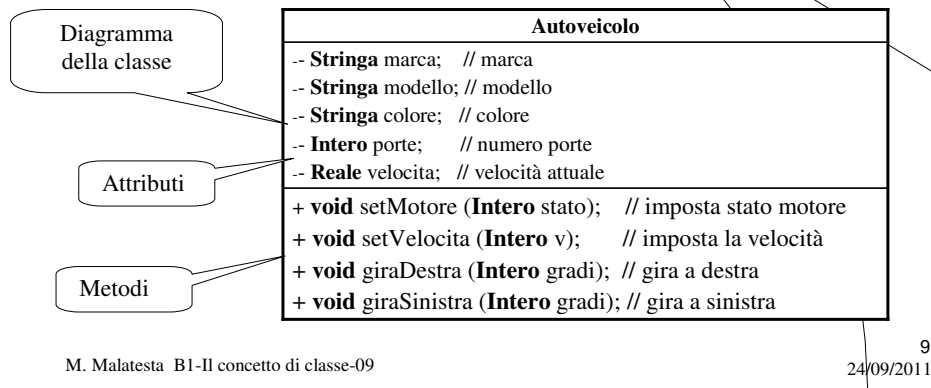
Nome della classe
Nome e tipo degli attributi
Prototipi dei metodi

M. Malatesta B1-II concetto di classe-09

8
24/09/2011

Diagramma della classe

Ad esempio, la classe **Autoveicolo** dell'esempio precedente, può essere appresentata come indicato dal seguente diagramma della classe (+ indica **public**, - indica **private**)



I metodi di default

Sebbene ogni classe possieda metodi specifici in base al problema, esistono tuttavia alcuni metodi, detti metodi di default, che sono praticamente sempre presenti.

I metodi di default sono:

- **Metodo costruttore**
- **Metodi accessori**
- **Metodi modificatori**

Vediamo in dettaglio a cosa serve ciascuno di essi.

Il metodo costruttore

Per istanziare una classe è opportuno che sia presente un particolare metodo detto **metodo costruttore**.

```
public Autoveicolo()
{
    marca = "";
    modello = "";
    colore = "";
    porte = 0;
    velocita = 0;
}
```

Il costruttore **inizializza** gli attributi con valori prestabiliti dal programmatore.

M. Malatesta B1-II concetto di classe-09

11
24/09/2011

Il metodo costruttore

Il costruttore:

- è un **metodo di default** che non ha valore di ritorno
- serve ad **inizializzare** gli oggetti della classe con valori prestabiliti ed evita il rischio di utilizzare oggetti i cui attributi non siano stati inizializzati.
- è l'unico metodo che può avere lo stesso nome della classe
- può essere senza parametri (gli attributi dell'oggetto creato avranno i valori che il programmatore assegna all'interno del costruttore)
- può essere con parametri (i valori degli attributi possono essere letti da input ed assegnati all'oggetto)

M. Malatesta B1-II concetto di classe-09

12
24/09/2011

Il metodo costruttore

La creazione di un oggetto (**istanza della classe**) si indica con

Autoveicolo *automobile* = **new** Autoveicolo();

Nome della classe

Istanza del costruttore

Viene dichiarato e allocato l'oggetto di nome *automobile*.

- L'accesso agli attributi segue la sintassi
automobile.nomeattributo
- L'invocazione dei metodi segue la sintassi
automobile.nomemetodo (parametri);

M. Malatesta B1-II concetto di classe-09

13
24/09/2011

Il metodo costruttore

Un esempio di **costruttore con parametri** può essere

public Autoveicolo (**Stringa** ma, **Stringa** mo, **Stringa** c, **Intero** p, **Reale** v)

```
{  marca = ma;  
  modello = mo;  
  colore = c;  
  porte = p;  
  velocita = v;  
}
```

Elenco parametri formali

In questo caso la creazione di un oggetto (**istanza della classe**) si indica con
Autoveicolo *automobile* = **new** Autoveicolo("Fiat", "Punto", "Rosso", 3, 40);

Elenco parametri attuali

M. Malatesta B1-II concetto di classe-09

14
24/09/2011

Il metodo costruttore

```
public Autoveicolo();  
public Autoveicolo (Stringa ma, Stringa mo, Stringa c, Intero p, Reale v)
```

I due metodi **hanno lo stesso nome**:

- in alcuni paradigmi di programmazione non è ammesso
- nel paradigma ad oggetti prende il nome di **sovraccarico dei metodi** (*overloading*),
- *hanno firma diversa*.
- è un esempio di **polimorfismo**, una delle proprietà della OOP che vedremo in seguito.

M. Malatesta B1-II concetto di classe-09

15
24/09/2011

I metodi accessori

I **metodi accessori** sono metodi che consentono di accedere in lettura agli attributi di una classe.

```
public Intero getVelocita () { Ritorna velocita; }
```

In generale i metodi accessori:

- non hanno parametri;
- hanno un valore di ritorno diverso da void
- hanno, per convenzione, un nome con prefisso **get** (sono anche detti **metodi get**)

M. Malatesta B1-II concetto di classe-09

16
24/09/2011

I metodi modificatori

I **metodi modificatori** sono metodi che consentono di accedere in scrittura agli attributi di una classe.

```
public void setVelocita (Intero v) { velocita = v; }
```

In generale i metodi modificatori:

- hanno come parametro il valore da assegnare all'attributo;
- hanno un valore di ritorno **void**
- hanno, per convenzione, un nome con prefisso **set** (sono anche detti **metodi set**)

M. Malatesta B1-II concetto di classe-09

17
24/09/2011

La classe Punto

ATTIVITA': disegnare il diagramma della classe *Punto* che individui un generico punto del piano cartesiano e munirla di costruttore con parametri e senza e di metodi accessori e modificatori per l'ascissa e per l'ordinata.

Punto	
- Reale x;	// ascissa
- Reale y;	// ordinata
+ Punto();	// costruttore senza parametri
+ Punto (Reale x, Reale y);	// costruttore con parametri
+ Reale getx ();	// restituisce l'ascissa
+ Reale gety();	// restituisce l'ordinata
+ void setx (Reale x);	// imposta l'ascissa
+ void sety (Reale y);	// imposta l'ordinata

M. Malatesta B1-II concetto di classe-09

18
24/09/2011

Argomenti

- Il concetto di classe
- Implementazione di classi
- Diagramma della classe
- I metodi di default
- Il metodo costruttore
- I metodi accessori
- I metodi modificatori
- La classe Punto

M. Malatesta B1-II concetto di classe-09

19
24/09/2011

Altre fonti di informazione

- P.Gallo, F.Salerno – Informatica Generale 1, ed. Minerva Italica
- M.Romagnoli, P.Ventura – Linguaggio C/C++, ed. Petrini
- A. Garavaglia, F.Petracchi, S.Forte
Strutture dati e programmazione per oggetti, ed. Masson Scuola

M. Malatesta B1-II concetto di classe-09

20
24/09/2011