Corso di Laurea Ingegneria Informatica

Fondamenti di Informatica 1

Dispensa E05

Definizione di classi

A. Miola

Novembre 2006

Contenuti

- Classi per istanziare oggetti
- Esempio: la classe Punto
- Variabili d'istanza
- Metodi d'istanza
- Costruttori
- Ulteriori aspetti nella definizione di classi
  - auto-referenziazione
  - i metodi toString e equals
  - sovraccarico del costruttore
  - metodi di classe
  - variabili di classe
  - classi e oggetti
  - occultamento dell'informazione
- Esempi: le classi Quadrato, Triangolo
- Esercizi

Prerequisiti

Questo capitolo presuppone la conoscenza degli argomenti già trattati nelle precedenti lezioni di questo corso, con particolare riferimento al capitolo 12 del libro di testo

Definizione di classi

- In questo capitolo vengono presentati i concetti relativi alla definizione di classi per istanziare oggetti
  - gli aspetti fondamentali del linguaggio Java nella definizione di classi
  - alcuni aspetti metodologici
- Nota Bene – Va ricordato che la definizione di una nuova classe implicitamente definisce un nuovo tipo riferimento
- Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al capitolo 18

Classi per istanziare oggetti

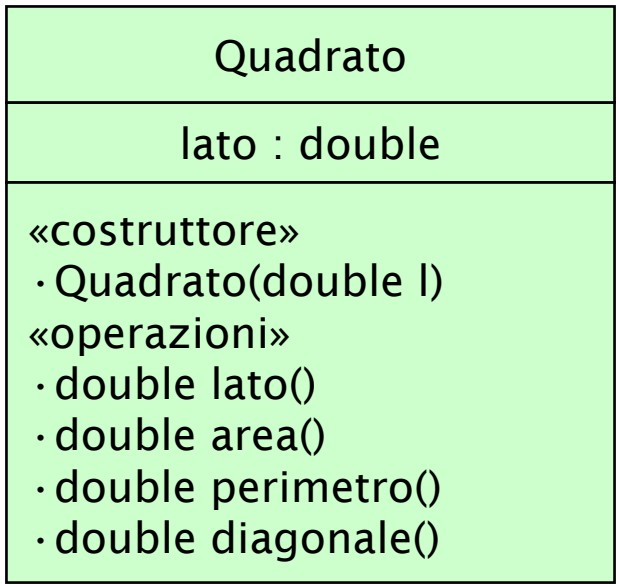
- **Una classe per istanziare oggetti è il progetto per una tipologia di oggetti istanza che sono caratterizzati da**
  - **uno stato, ovvero da un insieme di proprietà**
    - **variabili d'istanza**
  - **un comportamento, ovvero sanno eseguire delle operazioni**
    - **metodi d'istanza**
  - **Un metodo per la loro creazione ovvero per una opportuna inizializzazione del loro stato**
    - **costruttori**

La classe Quadrato . . .

- **Si vuole definire la classe Quadrato per istanziare oggetti**
  - un oggetto Quadrato modella un quadrato
  - lo stato di un oggetto Quadrato consiste nella lunghezza del suo lato
  - per creare un oggetto Quadrato è richiesta la lunghezza del lato come parametro
  - un oggetto Quadrato deve saper eseguire le seguenti operazioni
    - lato() — restituisce il lato del quadrato
    - area() — calcola e restituisce l'area del quadrato
    - perimetro() — calcola e restituisce il perimetro del quadrato
    - diagonale() — calcola e restituisce la diagonale del quadrato

. . . La classe Quadrato

- **Diagramma classi per Quadrato**



Uso della classe Quadrato

```
/* Applicazione di prova per la classe Quadrato. */
class TestQuadrato {
    public static void main(String[] args) {
        Quadrato p, q;        // due quadrati
        double areaP;         // area del quadrato p
        double perimetroQ;    // perimetro del quadrato q
    /* crea i due quadrati */
        p = new Quadrato(10);
        q = new Quadrato(20);
    /* calcola e visualizza l'area di p */
        areaP = p.area();
        System.out.println(areaP);                  // 100
    /* calcola e visualizza il perimetro di q */
        perimetroQ = q.perimetro();
        System.out.println(perimetroQ);             // 80
    }
}
```

La classe Quadrato

- **Ricordiamo come si definisce una classe per istanziare oggetti**
  - **bisogna dichiarare le variabili d'istanza**
    - **in questo caso, c'è bisogno di una variabile che rappresenta il lato del quadrato**
  - **bisogna definire (almeno) un costruttore per la classe**
  - **bisogna definire un metodo per ciascuna operazione che gli oggetti della classe devono saper eseguire**
    - **in questo caso, si devono definire i metodi lato(), area(), quadrato(), perimetro() e diagonale()**

La classe Quadrato — variabili d'istanza

```
/* Un oggetto Quadrato rappresenta
 * un quadrato. */

class Quadrato {
    /* lato del quadrato */
    private double lato;

    ... segue ...
```

La classe Quadrato — il costruttore

```
... segue ...

/* Crea un nuovo Quadrato di lato l. */
  public Quadrato(double l) {
      // pre: l>0
      this.lato = l;
  }

... segue ...
```

La classe Quadrato — il metodo lato()

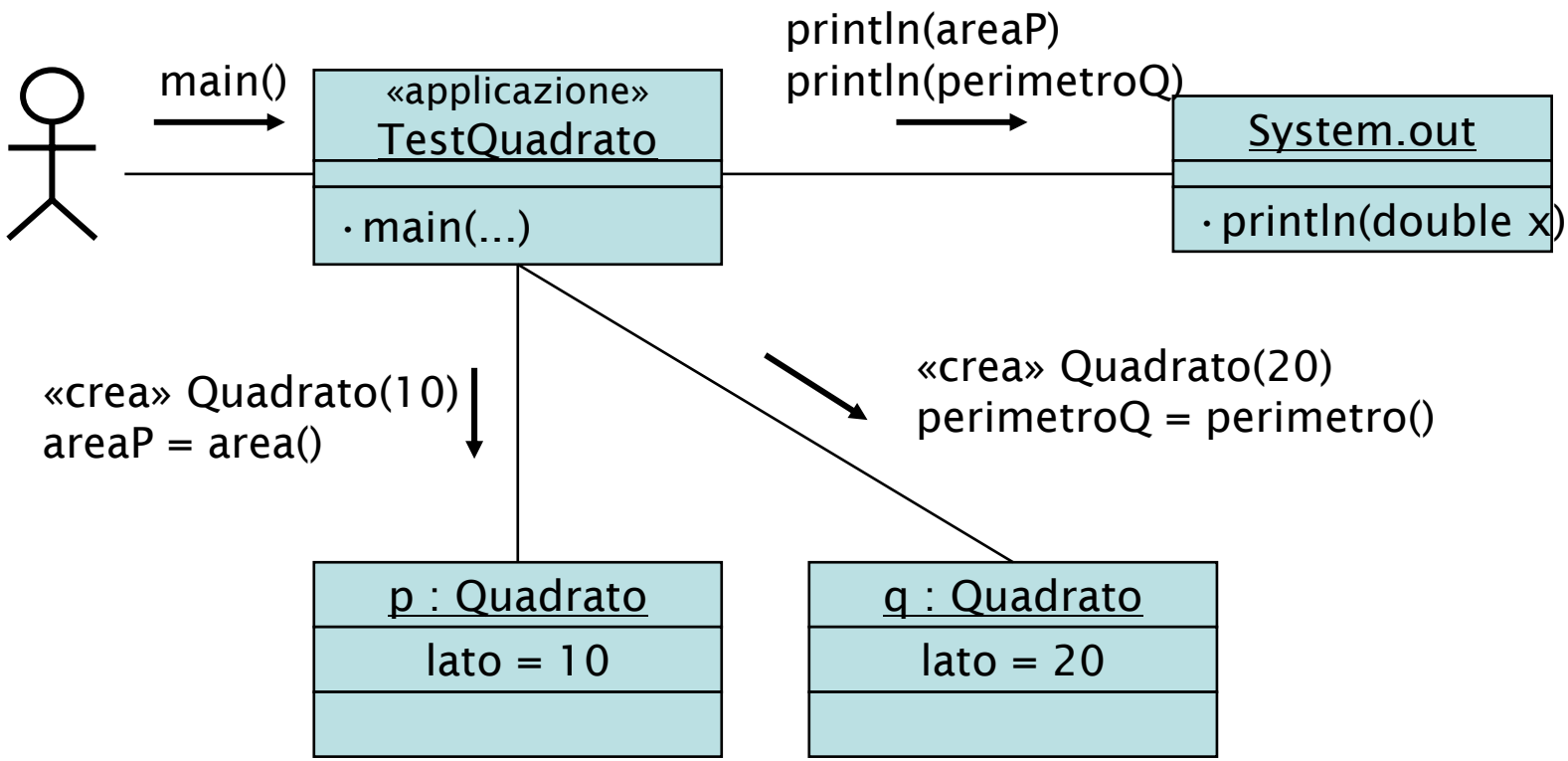
```
. . . segue . . .

    /* Calcola il lato del quadrato. */
    public double lato() {
        /* restituisce il lato */
        return this.lato;
    }
. . . segue . . .
```

- Esercizio - Completare la definizione della classe con la definizione degli altri metodi d'istanza previsti

Esecuzione di TestQuadrato

- **Lo stato di ogni oggetto istanza è indipendente da quello degli altri oggetti istanza della stessa classe**

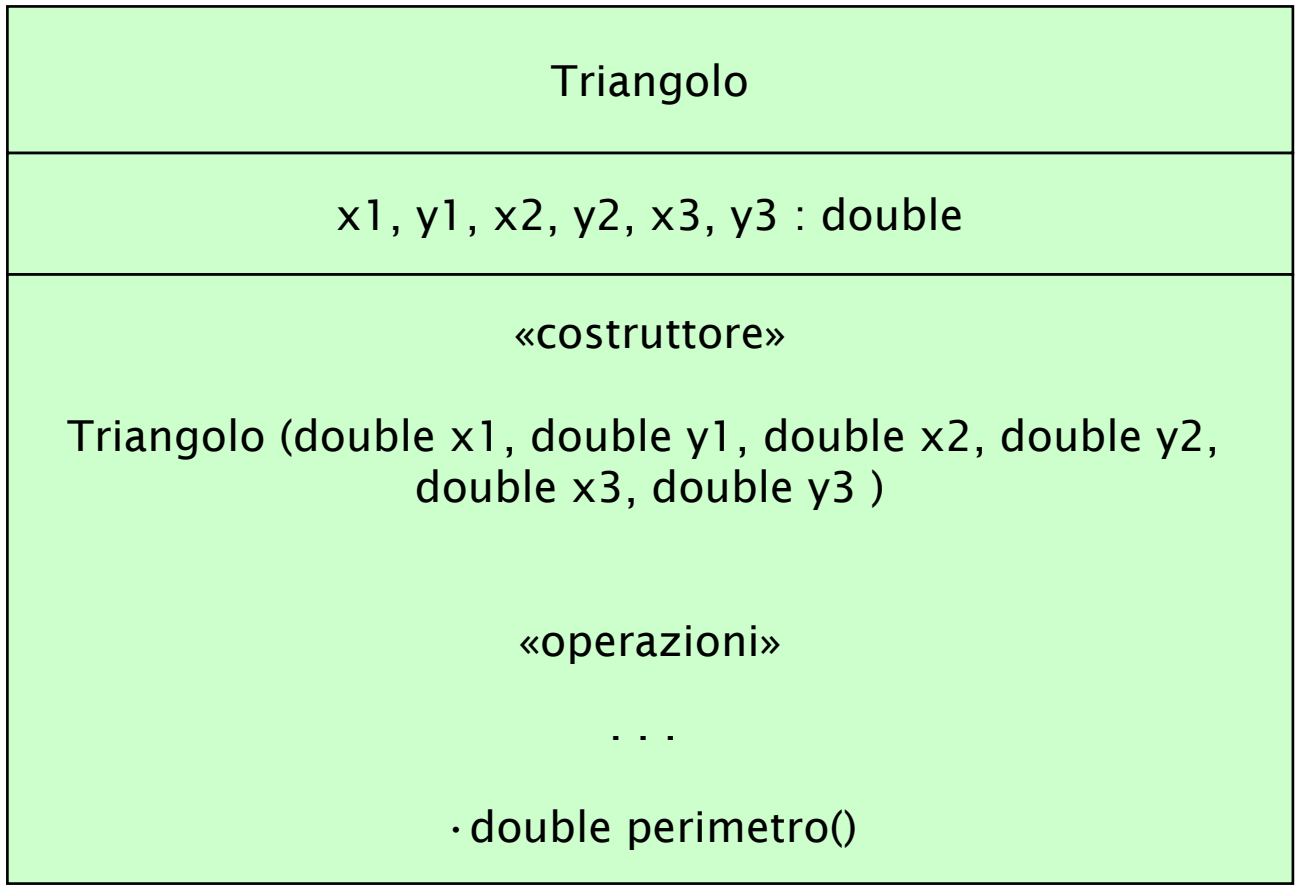


La classe Triangolo . . .

- **Si vuole definire la classe Triangolo per istanziare oggetti**
  - **un oggetto Triangolo modella un triangolo sul piano cartesiano**
  - **lo stato di un oggetto Triangolo consiste nella conoscenza delle coordinate dei suoi vertici x1, y1 , x2 , y2 , x3 , y3**
  - **per creare un oggetto Triangolo sono richiesti, come parametri, i valori delle coordinate dei suoi vertici**
  - **un oggetto Triangolo deve saper eseguire, tra altre possibili, la seguente operazione:**
    - **perimetro() — calcola e restituisce il perimetro del triangolo**

. . . La classe Triangolo

- **Diagramma classi per Triangolo**



Uso della classe Triangolo

```
/* Applicazione di prova per la classe Triangolo. */
class TestTriangolo {
    public static void main(String[] args) {
      Triangolo t;          // un triangolo
       double perimetroT;   // perimetro del triangolo t

     /* crea un triangolo */
      t = new Triangolo (1, 2, 4, 6, 8, 1);

    /* calcola e visualizza il perimetro di t */
      perimetroT = t.perimetro();
     System.out.print("Il perimetro del triangolo è ");
     System.out.println(perimetroT);    // 80 ???
    }
}
```

La classe Triangolo — variabili d'istanza

```
/* Un oggetto Triangolo rappresenta un
  triangolo sul piano cartesiano. */
class Triangolo {
    /* coordinate dei vertici */
    private double x1, y1, x2, y2, x3, y3;

    ... segue ...
```

La classe Triangolo — il costruttore Triangolo

```
... segue ...
    /* Crea un nuovo Triangolo con vertici le cui
     * coordinate x1, y1, x2, y2, x3, y3
     * sono rispettivamente ax, ay, bx, by, cx, cy. */
    public Triangolo (double ax, double ay, double bx,
                      double by, double cx, double cy)
  {
        // pre: ax>0, ay>0, bx>0, by>0, cx>0, cy>0
    this.x1 = ax;
    this.y1 = ay;
    this.x2 = bx;
    this.y2 = by;
    this.x3 = cx;
    this.y3 = cy;
    }                ... segue ...
```

La classe Triangolo — il metodo double perimetro()

- Il metodo perimetro() è un metodo d'istanza che deve semplicemente restituire il valore del perimetro del triangolo
- In pratica, quindi, il metodo deve calcolare il perimetro del triangolo (a partire dai valori che il costruttore Triangolo avrà assegnato alle variabili d'istanza x1, y1, x2, y2, x3, y3 del triangolo) ripercorrendo i passi dell'algoritmo già visto in precedenza per l'applicazione PerimetroTriangolo

La classe Triangolo — il metodo double perimetro()

```
    ... segue ...
/* Calcola il perimetro del triangolo. */
public double perimetro() {
    double d12, d13, d23;  // distanze tra i vertici
    double perimetro;      // perimetro del triangolo
/* calcola le distanze tra i vertici */
    d12 = Misuratore.distanza(this.x1, this.y1, this.x2,
  this.y2);
    d13 = Misuratore.distanza(this.x1, this.y1, this.x3,
  this.y3);
    d23 = Misuratore.distanza(this.x2, this.y2, this.x3,
  this.y3);
/* calcola il perimetro del triangolo */
    perimetro = d12 + d13 + d23;
/* restituisce il perimetro */
    return perimetro;   } }
```

?

La classe Triangolo — osservazioni ed esercizi

❑ Osservazioni

- Nella precedente definizione del metodo d'istanza perimetro() siamo proprio sicuri che sia tutto a posto ?
  - La classe Misuratore chi è in questo contesto ?
  - Dove sta? Dove sta il suo metodo distanza ?
  - E' tutto corretto quello che abbiamo scritto ?

❑ Esercizi

- Dare risposte alle precedenti domande intervenendo con le eventuali modifiche necessarie
- Costruire il diagramma di collaborazione per l'applicazione TestTriangolo

Esercizi — Altri classi per i triangoli

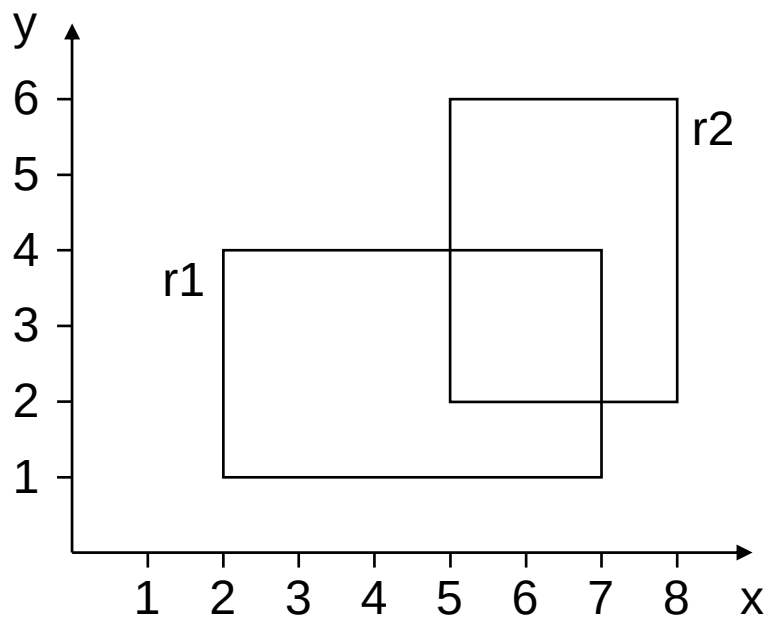
- Definire la classe TriangoloEquilatero il cui stato consiste nella lunghezza del suo lato ed è capace di calcolare il suo lato, il suo perimetro e la sua area
  - Suggerimento: Ricordarsi le specifiche (simili) della definizione della classe Quadrato
- Definire la classe TriangoloRettangolo il cui stato consiste nella lunghezza dei suoi cateti ed è capace di calcolare la sua ipotenusa, oltre a saper eseguire altre operazioni da scegliere
  - Suggerimento: Ricordarsi le specifiche (simili) della definizione della classe TriangoloEquilatero e la classe Ipotenusa già vista in precedenza
- Definire classi per altre tipologie di triangolo a piacere

Esercizi — Classe Quadrato

- **Definire una classe QuadratoSulPiano per definire quadrati sul piano cartesiano con i lati paralleli agli assi**
  - **lo stato di un oggetto QuadratoSulPiano consiste nella conoscenza delle coordinate dei suoi vertici vbs in basso a sinistra e vad in alto a destra, mentre le operazioni sono le stesse della classe Quadrato**
- **Nella classe QuadratoSulPiano aggiungere metodi di classe per verificare se**
  - **due quadrati hanno lo stesso lato**
  - **due quadrati hanno la stessa area**
  - **due quadrati sono uno interno all'altro**

Esercizio — Classe Rettangolo

- **Definire una classe Rettangolo**
  - **ciascun oggetto Rettangolo rappresenta un rettangolo in un piano bidimensionale, ortogonale rispetto agli assi cartesiani**



Caratterizzazione della classe Rettangolo

- **Comportamento**
  - un rettangolo deve saper calcolare
    - la sua base
    - la sua altezza
    - la sua area
  - un rettangolo deve sapersi traslare di (DX,DY)
  - bisogna poter calcolare l'intersezione tra due rettangoli
- **Stato**
  - lo stato di un rettangolo deve rappresentare
    - le sue dimensioni
    - la sua posizione nel piano
    - le posizioni dei suoi vertici

Esercizi — Altri Numeri

- Definire una classe NumeroReale per rappresentare numeri reali (di tipo double)
- Definire una classe NumeroRazionale per rappresentare numeri razionali come coppia di numeri interi (di tipo int)
- Definire una classe NumeroComplesso per rappresentare numeri complessi come coppia di numeri reali (di tipo double)

Esercizi — Altri oggetti

- Definire una classe **Libro** per rappresentare oggetti libro con il nome dell'autore, il titolo e il numero di pagine e con opportuni metodi d'istanza tra cui un metodo del tipo **String ToString()** per la sua descrizione
- Definire una classe **LineaBus** per rappresentare oggetti linea di autobus urbano con il numero identificativo, il nome dei due capolinea e con opportuni metodi d'istanza tra cui un metodo del tipo **String ToString()** per la descrizione del suo percorso

Esercizi — Altri oggetti

- Definire una classe **Auto** per rappresentare oggetti automobile con il nome della marca, il nome del modello, la targa e l'anno di immatricolazione e con opportuni metodi d'istanza tra cui un metodo del tipo **String ToString()** per la sua descrizione
- Definire una classe **Studente** per rappresentare oggetti studente con il cognome, il nome, il codice fiscale, il numero di matricola e con opportuni metodi d'istanza tra cui un metodo del tipo **String ToString()** per la sua descrizione

Cosa abbiamo visto finora

- Tipi riferimento e classi
- Esempi ed esercizi di definizione di classi per istanziare oggetti

Riferimenti al libro di testo

- **Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al capitolo 18**