Corso di Programmazione
Anno Accademico 2015–2016
Prof. Giovanni Pighizzini

Note ed esercizi aggiuntivi

Gli esercizi proposti sono utili per rivedere gli esempi riportati, che sono stati sviluppati e discussi in dettaglio a lezione.

10. Ereditarietà

Esempio. Lettura di una sequenza di figure, con scelta guidata in base a un menu. Comunicazione della figura di area più grande (o di una delle figure di area più grande, se vi sono più figure con la stessa area), con l'indicazione del suo tipo.

```
import prog.io.*;
import prog.utili.Figura;
import prog.utili.Rettangolo;
import prog.utili.Quadrato;
import prog.utili.Cerchio;

class FiguraAreaMax {

  public static void main(String[] args) {
    //predisposizione dei canali di comunicazione
    ConsoleInputManager in = new ConsoleInputManager();
    ConsoleOutputManager out = new ConsoleOutputManager();

    Figura rAreaMax = null;
    boolean continuare;

    do {
      //leggi i dati di una figura
      Figura r = leggiFigura(in, out);

      //confronta la figura con quella di area maggiore
      if (rAreaMax == null || r.haAreaMaggiore(rAreaMax))
        rAreaMax = r;

      continuare = in.readSiNo("Vuoi inserire i dati di un'altra " +
                               "figura? (s/n) ");
    } while (continuare);
```

```
  //comunica le caratteristiche della figura di area maggiore
  out.print("La figura di area maggiore è un ");
  if (rAreaMax instanceof Cerchio)
    out.print("cerchio: ");
  else if (rAreaMax instanceof Quadrato)
    out.print("quadrato: ");
  else
    out.print("rettangolo: ");
  out.println();
  out.println("  " + rAreaMax.toString());
  out.println("  area = " + rAreaMax.getArea() +
              ", perimetro = " + rAreaMax.getPerimetro());
}

private static Figura leggiFigura(ConsoleInputManager in,
                                  ConsoleOutputManager out) {
  Figura r = null;

  do {
    //scelta della figura
    out.println("    Scegli la figura da inserire: ");
    out.println();
    out.println(" r  Inserimento di un rettangolo");
    out.println(" q  Inserimento di un quadrato");
    out.println(" c  Inserimento di un cerchio");
    out.println();
    char scelta = in.readChar("    Scelta? ");
    out.println("");

    //leggi i dati di una figura del tipo selezionato
    switch (scelta) {
    case 'r':
      out.println("Inserimento di un rettangolo:");
      double x = in.readDouble(" - base? ");
      double y = in.readDouble(" - altezza? ");
      if (x < 0 || y < 0)
        out.println("I dati inseriti non rappresentano " +
                    "un rettangolo");
      else
        r = x == y ? new Quadrato(x) : new Rettangolo(x, y);
      break;

    case 'q':
      out.println("Inserimento di un quadrato:");
      x = in.readDouble(" - lato? ");
      if (x < 0)
        out.println("Attenzione: il lato di un quadrato " +
                    "non può essere negativo!");
      else
        r = new Quadrato(x);
      break;
```

```
      case 'c':
        out.println("Inserimento di un cerchio");
        x = in.readDouble(" - raggio? ");
        if (x < 0)
          out.println("Attenzione: il raggio di un cerchio " +
                      "non può essere negativo!");
        else
          r = new Cerchio(x);
        break;

      default:
        out.println("    Scelta non valida\n\n");
        break;
      }

    } while (r == null);

    //visualizza le caratteristiche della figura letta
    out.println("  " + r.toString());
    out.println("  area = " + r.getArea() +
                ", perimetro = " + r.getPerimetro());
    out.println();

    return r;
  }
}
```

Note

- Il metodo statico `leggiFigura` funge da sottoprogramma a servizio del `main`, ed evita di scrivere direttamente nel metodo `main` la lunga parte relativa alla lettura e alla costruzione dell'oggetto.

- Si osservi l'uso del supertipo comune `Figura` per la variabile `rAreaMax`, destinata a riferirsi alla figura più grande. Il ciclo `do-while` nel metodo `main`, utilizzato per determinare la figura più grande, è del tutto indipendente dai tipi `Cerchio`, `Rettangolo`, `Quadrato`, ma usa solo le caratteristiche del tipo `Figura`, in particolare il metodo `haAreaMaggiore`.

Esercizio 10.1
Modificate il codice presentato nell'esempio precedente in modo che oltre a comunicare il tipo della figura con area più grande ne indichi anche:

- il lato, se la figura è un quadrato,

- la base e l'altezza, se la figura è un rettangolo (ma non un quadrato),

- il raggio, se la figura è un cerchio.

Per ricavare queste informazioni utilizzate gli specifici metodi che trovate nelle classi che rappresentano i vari tipi di figure.

Esercizio 10.2
Modificate il codice presentato nell'esempio precedente in modo che, oltre a determinare la figura con area maggiore, determini anche quella con perimetro maggiore. La classe `Figura` dispone di un metodo

```
boolean haPerimetroMaggiore(Figura)
```

che restituisce `true` quando la figura che esegue il metodo ha perimetro maggiore di quella il cui riferimento viene fornito tramite l'argomento.

Esercizio 10.3
Scrivete un'applicazione che richieda all'utente di inserire i dati di un elenco di figure (rettangoli, quadrati, cerchi). L'inserimento può essere guidato mediante un menu, come nell'esempio precedente. L'applicazione dovrà determinare l'area media delle figure inserite e produrre:

- Un elenco di tutte le figure la cui area è superiore alla media, nello stesso ordine in cui sono state inserite.

- Tre elenchi come il precedente, ma suddivisi per tipo di figura (cioè uno per i rettangoli, uno per i quadrati, uno per i cerchi).

Per svolgere l'esercizio sfruttate il meccanismo dell'ereditarietà (potete ispirarvi all'esempio precedente). Come cambierebbe la soluzione senza l'uso del tipo `Figura` e dell'ereditarietà?

Esercizio 10.4
Si consideri una classe `Sopra` che, oltre a un costruttore privo di argomenti, offre i seguenti metodi:

- `public int doppio(int x)`
  Restituisce il doppio del valore ricevuto tramite l'argomento.

- `public int uno()`
  Restituisce `1`.

- `public int m(int x)`
  Restituisce un valore uguale a quello ricevuto tramite l'argomento.

Si consideri poi una classe `Sotto`, che estende `Sopra`. Anche questa classe fornisce un costruttore privo di argomenti. Inoltre la classe prevede i seguenti metodi, oltre a quelli ereditati da `Sopra`:

- `public int zero()`
  Restituisce `0`.

- `public int m(int x)`
  Restituisce il triplo del valore ricevuto tramite l'argomento.

Si consideri, infine, la seguente classe di prova:

```
import prog.io.*;

class Prova {
  public static void main(String[] args) {
    ConsoleInputManager in = new ConsoleInputManager();
    ConsoleOutputManager out = new ConsoleOutputManager();

    int v = in.readInt();
    Sopra s;
    int x = v;

    if (v > 10)
      s = new Sopra();
    else
      s = new Sotto();
    out.println(s.doppio(v));
    out.println(s.m(v));
    if (s instanceof Sotto) {
      Sotto st = (Sotto) s;
      x = x + st.zero();
    } else
      x = x + s.uno();
    out.println(x);
  }
}
```

1. Scrivete l'output prodotto dal programma nell'ipotesi che l'utente inserisca il valore `8`.
2. Scrivete l'output prodotto dal programma nell'ipotesi che l'utente inserisca il valore `11`.