Le classi astratte

Nella programmazione ad oggetti, la realizzazione delle applicazioni inizia con la dichiarazione delle classi, elencando gli attributi e i metodi, e, successivamente, prosegue con la creazione degli oggetti come istanze delle classi.
In Java esistono delle classi particolari che non possono essere utilizzate per creare gli oggetti; si dice che non possono essere istanziate. Queste classi sono chiamate **classi astratte**.
Le classi astratte si distinguono dalle altre classi perché hanno, di solito, almeno un metodo che non è stato implementato, ma è stato soltanto definito. In pratica, nella definizione della classe, viene indicata l'intestazione del metodo ma non viene descritto il codice all'interno delle parentesi graffe. Non essendo definito il codice del metodo, un eventuale oggetto non saprebbe comportarsi qualora venisse richiesta l'esecuzione di quel metodo specifico.
Una classe astratta, al cui interno è definito un metodo astratto, assume la seguente struttura:

```
abstract class NomeClasse
{
   abstract livelloDiVisibilità tipoRestituito nomeMetodo(parametri);
}
```

La parola chiave **abstract** nell'intestazione della classe indica una classe astratta.
Il metodo, definito con la parola chiave *abstract*, non deve avere il blocco del codice, solitamente identificato dalle parentesi graffe, ma l'intestazione deve terminare con il punto e virgola.
Le classi astratte, prese singolarmente, non sono molto utili, mentre assumono un ruolo importante quando vengono inserite in una gerarchia di classi. La classe astratta descrive solitamente la sopraclasse di un insieme di classi che devono condividere un'interfaccia comune. L'interfaccia che la classe impone alle sue sottoclassi è rappresentata dall'elenco di tutti i metodi astratti dichiarati nella classe astratta.
Le sottoclassi, per poter essere utilizzate e istanziate in un'applicazione, saranno obbligate a definire un'implementazione di tutti i metodi astratti ereditati dalla sopraclasse. Quando tutti i metodi sono stati implementati, la classe non è più una classe astratta e può essere usata per generare gli oggetti.
Si supponga di voler realizzare un'applicazione per calcolare l'area di diverse figure geometriche. Si può pensare di organizzare la gerarchia delle classi in modo da creare una superclasse astratta *Figura* che imponga alle sue sottoclassi di avere un'implementazione per il metodo *calcolaArea*. Dichiarando le figure geometriche *Quadrato* e *Cerchio* come sottoclassi di *Figura*, si obbligano queste due sottoclassi a ridefinire il metodo *calcolaArea* al loro interno.
La definizione della classe astratta *Figura* è la seguente:

```
abstract class Figura
{
   abstract public double calcolaArea();
}
```

Una classe di questo tipo non può essere istanziata.
Infatti, se si prova ad inserire in un programma il seguente comando,

```
Figura f = new Figura();
```

il compilatore Java restituisce il messaggio di errore:

```
error: Figura is abstract; cannot be instantiated
```

A partire dalla classe *Figura*, si può costruire la classe concreta *Quadrato* nel seguente modo.

```
class Quadrato extends Figura
{
   private double lato;

   public Quadrato(double lato)
   {
      this.lato = lato;
   }

   public double calcolaArea()
   {
      return lato*lato;
   }
}
```

La classe *Cerchio* è riportata di seguito.

```
class Cerchio extends Figura
{
   private double raggio;

   public Cerchio(double raggio)
   {
      this.raggio = raggio;
   }

   public double calcolaArea()
   {
      return raggio*raggio*Math.PI;
   }
}
```

Si noti che se la sottoclasse *Cerchio* non contenesse un'implementazione del metodo astratto *calcolaArea*, il compilatore Java restituirebbe il messaggio di errore:

```
error: Cerchio is not abstract and does not override abstract method
calcolaArea() in Figura
```

Il programma che calcola l'area di un quadrato e quella di un cerchio utilizza un oggetto di classe *Figura* a cui vengono successivamente assegnate le istanze della classe *Quadrato* e della classe *Cerchio*. Nel momento in cui viene richiesta l'esecuzione del metodo *calcolaArea*, il sistema di runtime di Java stabilisce qual è l'opportuno metodo da eseguire, a seconda che la figura sia un quadrato oppure un cerchio (*polimorfismo*).
Il codice sorgente che calcola l'area di un quadrato e di un cerchio è il seguente:

```
class ProgFigura
{
   public static void main(String args[])
   {
      Figura f;

      f = new Quadrato(4.5);
      System.out.println("Area = " + f.calcolaArea());

      f = new Cerchio(2.76);
      System.out.println("Area = " + f.calcolaArea());
   }
}
```