

## Esercizio 1.

In Java, l'operatore `instanceof` ha la seguente definizione:

### Sintassi

```
espressione_riferimento instanceof tipo_riferimento
```

► È un'espressione:

**tipo:** `boolean`

**valore:** `true` se e solo se *espressione\_riferimento* si riferisce a un oggetto che è un'istanza di *tipo\_riferimento*

Ricordando che un'istanza di una classe è istanza anche di tutte le sue superclassi

Utilizzando le classi `Rettangolo` e `Quadrato` (estensione di `Rettangolo`) del package `prog.utili`, scrivere un'applicazione che legge una sequenza di figure e indica quella di area massima specificandone anche il tipo (quadrato o rettangolo e il perimetro). Suggerimenti: 1) in un ciclo iterativo, usare un metodo `leggiRettangolo` che gestisce l'input e restituisce un riferimento ad un oggetto `Rettangolo`; 2) in `leggiRettangolo`, ad ogni lettura verificare se è necessario istanziare un oggetto di classe `Rettangolo` o `Quadrato`; 3) nel corso dell'iterazione tenere traccia della figura di area massima usando il metodo:

```
public boolean haAreaMaggiore(Rettangolo r)
```

della classe `Rettangolo`, che restituisce `true` se il rettangolo ha area maggiore di `r`.

## Esercizio 2.

Le classi `Rettangolo` e `Quadrato` del package `prog.utili` sono in realtà delle sottoclassi della classe `Figura`. Il package `prog.utili` comprende anche la classe `Triangolo` che estende la classe `Figura` e che ha il seguente costruttore:

```
Triangolo(double base, double latoSinistro, double angoloCompreso)  
Costruisce il triangolo dati la lunghezza della base, la lunghezza del lato a adiacente alla sinistra della base e l'angolo fra essi compreso.
```

Si estenda l'applicazione dell'Esercizio 1 in modo che tratti anche triangoli.

## Esercizio 3.

Oltre alle classi `Rettangolo` e `Triangolo`, la classe `Figura` è estesa anche dalla sottoclasse `Cerchio` con il seguente costruttore:

```
Cerchio(double r)  
Costruisce un oggetto che rappresenta un cerchio di raggio uguale a quello fornito come argomento.
```

Si estenda l'applicazione dell'Esercizio 2 in modo che tratti anche cerchi. In questo caso sarà opportuno che, in fase di input, l'utente indichi in un semplice menu il tipo di figura geometrica che intende inserire tra rettangolo, quadrato, triangolo e cerchio.

#### Esercizio 4.

Scrivere un'applicazione che legge una sequenza indefinita di figure geometriche (classe `Figura`) e quindi:

- 1) le elenca in output organizzate per tipo (ad esempio prima tutti i quadrati, poi tutti i rettangoli etc.);
- 2) indica la figura con area più vicina a quella media, specificandone l'area, il tipo e le caratteristiche geometriche (es. il raggio se cerchio, base ed altezza se rettangolo etc.).

Si suggerisce di utilizzare una classe collezione (ad esempio `ArrayList<Figura>`) sfruttando il fatto che un oggetto di superclasse può sempre essere usato per riferirsi ad oggetti di sue sottoclassi.