

Programmazione orientata agli oggetti

Luca Iocchi, Massimo Mecella
Daniele Sora

Esercitazione 6 Figure Geometriche (II)



Figure geometriche

Realizzare le classi necessarie per modellare figure geometriche. Nel nostro dominio ci interessano le seguenti figure geometriche: rettangoli, quadrati, cerchi, ellissi, e triangoli.

Definire un insieme di classi e una opportuna gerarchia partendo da una classe base [FiguraGeometrica](#).

Ogni figura geometrica è caratterizzata da una propria descrizione (una stringa) e da campi dati che consentono il calcolo di area e perimetro.

Non deve essere possibile istanziare oggetti della classe [FiguraGeometrica](#)



Figure geometriche

Ogni figura geometrica è caratterizzata da una propria descrizione (una stringa) e da campi dati che consentono il calcolo dell'area.

Ogni figura geometrica deve avere dei costruttori per inizializzare i valori delle proprietà e deve implementare le seguenti operazioni

- double area()** : restituisce l'area della figura geometrica
- double perimetrr()** : restituisce il perimetro della figura geometrica
- String toString()** : restituisce una stringa con le informazioni della figura geometrica (descrizione e dati)



Figure geometriche

Si vuole inoltre poter confrontare le aree tra due istanze di `FiguraGeometrica`.

Si definisca un'interfaccia `Confrontabile` che preveda la definizione di un metodo

-`boolean maggiore(Confrontabile c)` : restituisce *false* se *c* è più grande dell'istanza su cui viene chiamato il metodo, *true* altrimenti.

Si fornisca un'implementazione di `Confrontabile` per le figure geometriche.



Figure geometriche

Scrivere il metodo statico

`Confrontabile massimo(Confrontabile[] c)`

che, dato un array di oggetti confrontabili `c` restituisca l'oggetto più grande contenuto in `c`

Scrivere un programma Java che crei ed inizializzi un array di figure geometriche di diverso tipo (con valori random o costanti) ed individui la figura più grande, stampandone i dettagli

