

Esame di Programmazione II, 9 luglio 2025

(si consegna NoSuchRowException.java, AbstractFigure.java, Triangle.java, Circle.java e Frame.java)

*Si crei un progetto Eclipse e il package `it.univr.figures`. Si copino al suo interno le classi del compito. Non si modifichino le dichiarazioni dei metodi e delle classi. Si possono definire altri campi, metodi o costruttori non richiesti dal compito, ma devono essere *private*. Si possono definire altre classi, che in tal caso vanno consegnate. La soluzione che verrà consegnata dovrà compilare, altrimenti non verrà corretta.*

L'interfaccia `Figure` rappresenta una figura che si può stampare. Essa ha un'altezza (numero di caratteri verticalmente) e una larghezza (numero di caratteri orizzontalmente) e un metodo `getRow(int n)` che restituisce la riga n -esima della figura. Per esempio, la seguente figura:

```
*      questa riga e' getRow(0)
* *    questa riga e' getRow(1)
* *    questa riga e' getRow(2)
*      * questa riga e' getRow(3)
*****questa riga e' getRow(4)
```

rappresenta un triangolo di altezza 5 e larghezza 9. Le sue cinque righe sono restituite dal metodo `getRow`, come indicato sopra. Per esempio, `getRow(2)` restituisce la stringa " `* * * * *`" (inclusi gli spazi all'inizio e alla fine, indicati qui con , ma senza nessuna andata a capo alla fine). Ne consegue che il `toString()` di una figura è la concatenazione dei suoi `getRow()`, con una andata a capo aggiunta alla fine di ciascuno. Se `getRow` viene chiamato fuori dai limiti della figura, esso lancia una `NoSuchRowException`. Per esempio, questo accadrebbe se nella figura mostrata sopra si chiamasse `getRow(5)` o `getRow(6)` o `getRow(-1)`.

Per capire meglio le classi che andranno implementate, si consideri il seguente `main`, già scritto e da non modificare:

```
for (int h = 1; h <= 5; h++) {
    System.out.println("triangolo di altezza " + h + ":");
    System.out.println(new Triangle(h));
}
for (int r = 1; r <= 16; r += 5) {
    System.out.println("cerchio di raggio " + r + ":");
    System.out.println(new Circle(r));
}
for (int h = 1; h <= 5; h++) {
    System.out.println("triangolo di altezza " + h + ", in cornice:");
    System.out.println(new Frame(new Triangle(h)));
}
for (int r = 1; r <= 16; r += 5) {
    System.out.println("cerchio di raggio " + r + ", in cornice doppia:");
    System.out.println(new Frame(new Frame(new Circle(r))));
}
// ok, nessuna eccezione
new Frame(new Circle(2)).getRow(5);
// eccezione, poiche' un cerchio di raggio 2 ha le righe da 0 a 3,
// e aggiungendo una cornice le righe andranno da 0 a 5; quindi 6 e' fuori
new Frame(new Circle(2)).getRow(6);
```

che dovrebbe stampare:

triangolo di altezza 1:

*

triangolo di altezza 2:

*

triangolo di altezza 3:

*
* *

triangolo di altezza 4:

*
* *
* *

triangolo di altezza 5:

*
* *
* *
* *

cerchio di raggio 1:

**
**

cerchio di raggio 6:

** **
* *
* *
* *
* *
* *
* *
* *
* *
* *
* *
* *

cerchio di raggio 11:

*** **
** **
* *
* *
** **

cerchio di raggio 16:

[illegible]

triangolo di altezza 1, in cornice:

```
@@@
@*@
@@@
```

triangolo di altezza 2, in cornice:

```
@@@@@
@ * @
@***@
@@@@@
```

triangolo di altezza 3, in cornice:

```
@@@@@@@
@  *  @
@ * * @
@*****@
@@@@@@@
```

triangolo di altezza 4, in cornice:

```
@@@@@@@@@
@   *   @
@  * *  @
@ *   * @
@*****@
@@@@@@@@@
```

triangolo di altezza 5, in cornice:

```
@@@@@@@@@@@
@    *    @
@   * *   @
@  *   *  @
@ *     * @
@*****@
@@@@@@@@@@@
```

cerchio di raggio 1, in cornice doppia:

```
@@@@@@
@@@@@@
@@**@@
@@**@@
@@@@@@
@@@@@@
```

cerchio di raggio 6, in cornice doppia:

```
@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@
@@      ****      @@
@@  **      **  @@
@@ *           * @@
```

```

@@ *      * @@
@@*        *@@
@@*        *@@
@@*        *@@
@@*        *@@
@@ *      * @@
@@ *      * @@
@@ **     ** @@
@@      **** @@
@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@

```

cerchio di raggio 11, in cornice doppia:

```

@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@
@@      *      @@
@@      ***      @@
@@      **      @@
@@      *      @@
@@      *      @@
@@ **      ** @@
@@ *      * @@
@@ *      * @@
@@ *      * @@
@@*      *@@
@@*      *@@
@@*      *@@
@@*      *@@
@@*      *@@
@@*      *@@
@@ *      * @@
@@ *      * @@
@@ **      ** @@
@@ *      * @@
@@ *      * @@
@@ **      ** @@
@@      ***      @@
@@      *      @@
@@      **      @@
@@      ****      @@
@@      *      @@
@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@

```

cerchio di raggio 16, in cornice doppia:

```

@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@
@@      *      @@
@@      ***      @@
@@      **      @@
@@      **      @@
@@      *      @@
@@      *      @@
@@      *      @@

```

```

@@ **                               ** @@
@@ *                               *  @@
@@ *                               *  @@
@@ *                               *  @@
@@ *                               *  @@
@@ *                               *  @@
@@*                               *@@
@@*                               *@@
@@*                               *@@
@@*                               *@@
@@*                               *@@
@@*                               *@@
@@*                               *@@
@@*                               *@@
@@*                               *@@
@@ *                               *  @@
@@ *                               *  @@
@@ *                               *  @@
@@ *                               *  @@
@@ **                             **  @@
@@ *                               *  @@
@@  *                             *   @@
@@   *                             *    @@
@@    **                           **   @@
@@     **                           **    @@
@@      ***                         ***   @@
@@       *****                    @@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

```

Exception in thread "main" it.univr.figures.NoSuchRowException

Esercizio 1 (2 punti). Si scriva la classe di eccezione **controllata** `NoSuchRowException.java`.

Esercizio 2 (7 punti). Si completi la classe astratta `AbstractFigure.java`, con l'unico metodo `toString()`. Questo `toString()` verrà ereditato da tutte le implementazioni concrete delle figure, senza modifiche, quindi lo definiamo `final`.

Esercizio 3 (7 punti). Si completi la classe concreta `Triangle.java` che implementa un triangolo rettangolo con il vertice posizionato in alto e di altezza `height`. Guardate le figure di esempio stampate sopra per capire qual è la sua larghezza e come deve essere implementato il suo `getRow()`. Per quest'ultimo metodo, il suggerimento è di distinguere i quattro casi: prima riga, ultima riga, righe in mezzo, altre righe. Si suggerisce inoltre di usare il metodo delle stringhe `s.repeat(n)`, che restituisce la stringa `s` ripetuta `n` volte.

Esercizio 4 (7 punti). Si completi la classe concreta `Circle.java` che implementa un cerchio di raggio `radius`. Il suo metodo `getRow()` è già parzialmente scritto. Il suggerimento è di completarlo seguendo i commenti.

Esercizio 5 (8 punti). Si completi la classe concreta `Frame.java` che implementa una figura ottenuta decorando un'altra figura `figure` con una cornice aggiuntiva di caratteri `@` al suo intorno. Per il `getRow()`, il suggerimento è di distinguere il caso della prima e ultima riga della figura risultante (con cornice) dal caso delle altre sue righe.