Object Oriented Programming

—

Exercise Series 5

March 2024

Exercise 1 Design a class `Rectangle` with the following features. An instance of this class must be able to represent a rectangle positioned on a 2D-grid, with its sides parallel to the axes of the coordinate system. The coordinates x and y of each corner of a `Rectangle` are integer numbers. The area of a `Rectangle` has to be strictly positive. Furthermore, a color among `RED`, `BLUE`, and `GREEN` is associated to each rectangle (cf. the file `MyColor.java`).

Make sure the way you store the information about the rectangle is as simple as possible, and does not contain redundant information.

Guide: You can use a class defined in a previous exercise session to deal with the coordinates, but make sure your implementation respects the encapsulation principle.

Exercise 2 Design a class `Grid`, representing a 2D plane containing a set of `Rectangles`. It should implement the following methods:

- `void rotate(MyColor c)` that computes the effect of a 90° (clockwise) rotation for each rectangle of color `c` in the plane, with $(0, 0)$ as its center of rotation.
- `void rotate(int cx, int cy, MyColor c)` that has a similar behavior as `rotate()`, but with $(\texttt{cx}, \texttt{cy})$ as its center of rotation.
- `boolean overlaps()` which returns true, if at least two rectangles have their inner area (excluding the sides and the corners) overlapping.

Is it an issue that two methods share the same name?

Bonus: In another dedicated class, design a method `solve()` that, given a `Grid`, tries to rotate the rectangles (by group of colours, using only the rotation centered on $(0, 0)$) until no rectangle overlaps.

Exercise 3 The Java archive `GUI.jar` has a class `GUI` that allows you to display all the rectangles in a given grid. Its constructor takes a `Grid` as parameter. After being instanciated, it can be initialised by calling the method `start()`.

This graphic interface allows the following interactions:

- When pressing the key 'r' of your keyboard, the `GUI` should call the method `rotate()` of the associated `Grid`, and update the display.
- When pressing the key 'c', it should change the current color following the pattern `RED` $\rightarrow$ `BLUE` $\rightarrow$ `GREEN` $\rightarrow$ `RED` $\rightarrow \ldots$

NB: For the display to work properly, each corner (x, y) of a rectangle should satisfy $-25 \leq x \leq 25$ and $-25 \leq y \leq 25$.