Object-Oriented Programming
June 2022

Notes or documents of any kind forbidden. Duration: 3 1/2h. Please answer the questions on separate sheets labeled with your name, section, and student ID.

---

1. The problem consists in programming in Java a class `Rectangle` suited for representing rectangles that have their sides parallel to the axes of the (x, y) coordinate system. Such rectangles satisfy the following properties.

   - The x and y coordinates of each corner of a rectangle are (positive or negative) integer numbers.
   - A rectangle cannot have a zero area, in other words, opposite corners of a rectangle must have different x and different y coordinates.

   The class `Rectangle` should satisfy the following requirements:

   - It must be possible to create a rectangle specified by the coordinates of two opposite corners.

     Note: Each rectangle admits two pairs of opposite corners, and the order between them is not taken into account. For instance, the pairs of corners $((2, 2), (4, 1))$ and $((4, 2), (2, 1))$ define the same rectangle.
   - It must be possible to check whether a given point (x, y) belongs or not to a rectangle. A point belongs to a rectangle if it is either located inside this rectangle, on one of its sides, or at one of its corners.
   - It must be possible to compute the *bounding box* of two given rectangles R_1 and R_2, defined as the smallest rectangle that contains all the points that belong to R_1 or R_2.
   - It must be possible to check whether a rectangle R_1 is included into a rectangle R_2, i.e., whether all the points of R_1 belong to R_2.
   - Instances of the class must be clonable, comparable to each other, and serializable. It must be possible to manipulate them simultaneously from separate threads.
   - In case of any error, a dedicated exception should be thrown.

   Note: You are free to implement any additional class required by your solution.

2. (a) Define a subclass `Square` of the class `Rectangle` obtained in (1), that is only able to represent squares, i.e., rectangles in which all sides share the same length.

   Note: The bounding box of two squares is a rectangle, and not necessarily a square.

   (b) What is the particular application of inheritance used in this case? Is the substitution principle satisfied?

3. (All answers should be thoroughly justified.)

   (a) Explain as simply as possible the operation performed by the following Java method

```
public boolean m()
{
  return a() && b();
}
```

   (b) The following fragment of Java code contains an error.

```
class C
{
  public final int x;

  public C(int x)
  {
    this.x = x;
  }
}

class D extends C
{
  public int y;
}
```

   Explain the nature of this error, whether it will be reported at compile time or at runtime, and how it can be corrected.

   (c) What is an abstract method? What is the purpose of defining such a method? Can an abstract method be defined in any class?

   (d) In Java, why is it forbidden to define a generic exception class?

   (e) In Java, what is a lock? How is a lock created? Give an example of a (complete) Java program that manipulates (in any way that you choose) the lock of an object.