

Object-Oriented Programming

June 2023

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 `BoundingBox` with the following features. An instance of this class must be able to represent the *bounding box* of a set of 2-dimensional points that are supplied one by one to this instance. The bounding box of the set of points $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ is defined as the smallest rectangle, aligned with the axes of the coordinate system, that contains all those points. In other words, this bounding box is characterized by two points (x_{min}, y_{min}) and (x_{max}, y_{max}) such that x_{min}, y_{min} are the greatest values and x_{max}, y_{max} the smallest values that satisfy $x_{min} \leq x_i \leq x_{max}$ and $y_{min} \leq y_i \leq y_{max}$ for every $i \in [1, n]$. For instance, the bounding box of $\{(0, 4), (-1.5, 2), (3, 3.5)\}$ corresponds to $(x_{min}, y_{min}) = (-1.5, 2)$ and $(x_{max}, y_{max}) = (3, 4)$, and has an area equal to 9. Note that if the set of points is empty, the values of (x_{min}, y_{min}) and (x_{max}, y_{max}) , and therefore the bounding box of this set, are undefined.

The class `BoundingBox` should satisfy the following requirements:

- Initially, an instance of this class represents the bounding box of an empty set of points. One can then add individual points by invoking a dedicated method (once for each point). The x and y coordinates of points are real numbers.
- It must be possible to print a bounding box (in the format of your choice) on standard output. *Note:* Only the coordinates of the two points that characterize the bounding box need to be printed, and not those of all the points that belong to the set.
- It must be possible to compute the area of a bounding box.
- Instances of this class must be clonable, comparable to each other, and serializable. It must be possible to manipulate them simultaneously from separate threads.

Note: The bounding boxes of two sets of points are considered to be equal if they consist in identical rectangles. This does not require the sets of points to be equal.

- In case of any error, a dedicated exception should be thrown.

Note: You are free to implement any additional classes required by your solution, as well as to choose the interpretation of details that are not specified in this problem statement.

2. (a) How would you define a subclass `CountingBoundingBox` of the class `BoundingBox` that represents, in addition to the bounding box of a set of points, the number of points in this set, and makes this number available by a dedicated method? (You do not need to fully program the class `CountingBoundingBox`; it is sufficient to explain what you would do.)
(b) Which application of inheritance do you use in your answer to (a)? Is the substitution principle satisfied? (Justify your answer.)
3. (a) What is a class variable? What are the two mechanisms offered by Java for initializing such a variable?
(b) Explain the difference between *static link* and *dynamic link*. In Java, for what reason are variables accessed by static rather than dynamic link?
(c) Program a generic class `OneElementStack<T>` that represents a stack of capacity one, for values of type `T`. You are not asked to implement other methods than `push` and `pop`. An instance of this class must initially represent an empty stack. In the case of an error, such as pushing on a stack containing already a value, or popping from an empty stack, a `StackException` must be thrown.
4. Consider the following Java class:

```
public class C
{
    private volatile int v = 0;

    public void m1() throws InterruptedException
    {
        while (true)
        {
            synchronized(this) { wait(); }
            System.out.println(v);
        }
    }

    public void m2() throws InterruptedException
    {
        while (true)
        {
            v++;
            synchronized(this) { notify(); }
        }
    }
}
```

A program instantiates the class `C`, then creates two concurrent threads that invoke (respectively) the methods `m1` and `m2` of the resulting object. *Note:* You are *not* asked to write this program.

Explain what will happen during the execution of this program, by answering the following questions:

- (a) What will be printed by the program?
- (b) Is it possible for non consecutive values (such as 1 followed by 3) to be printed?
- (c) Is it possible to have an execution that does not print the value 1?
- (d) Can an execution end up in a deadlock?

(All of your answers must be justified.)