Object-Oriented Programming
Second session, 2023

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

---

1. The problem consists in programming in Java a class `BrokenLine` for representing broken lines in 2D space. A broken line is characterized by a sequence $[(x_1, y_1); (x_2, y_2); \ldots; (x_n, y_n)]$ of 2D coordinates of points, such that for all $i \in [1, n-1]$, $(x_{i+1}, y_{i+1}) \neq (x_i, y_i)$. The number n of points in the sequence satisfies $n \geq 2$, and the coordinates x_i and y_i are integer numbers.

   The class `BrokenLine` should satisfy the following requirements:

   - It should be possible to create an instance of `BrokenLine` corresponding to a sequence $[(x_1, y_1); (x_2, y_2)]$, for given values of x_1, y_1, x_2 and y_2.
   - It must be possible to print a broken line (in the format of your choice) on standard output.
   - It must be possible to compose two given broken lines $[(x_1, y_1); (x_2, y_2); \ldots; (x_n, y_n)]$ and $[(x'_1, y'_1); (x'_2, y'_2); \ldots; (x'_{n'}, y'_{n'})]$ in order to form a new broken line, which corresponds to $[(x_1, y_1); \ldots; (x_n, y_n); (x'_1, y'_1); \ldots; (x'_{n'}, y'_{n'})]$ if $(x_n, y_n) \neq (x'_1, y'_1)$, and to $[(x_1, y_1); \ldots; (x_n, y_n); (x'_2, y'_2); \ldots; (x'_{n'}, y'_{n'})]$ if $(x_n, y_n) = (x'_1, y'_1)$.
   - 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: Two broken lines are considered to be equal if they correspond to identical sequences of coordinates.
   - 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) Define a subclass `Polygon` of the class `BrokenLine`, for representing broken lines whose sequence of coordinates $[(x_1, y_1); (x_2, y_2); \ldots; (x_n, y_n)]$ satisfies $n \geq 3$ and $(x_n, y_n) = (x_1, y_1)$. Instances of `Polygon` are created by specifying their full sequence $[(x_1, y_1); (x_2, y_2); \ldots; (x_n, y_n)]$ of coordinates, and do not support the composition operation defined for broken lines.

(b) Which application of inheritance do you use in your answer to (a)? Is the substitution principle satisfied? (Justify your answer.)

3. (a) In the Java language, when a class is instantiated, which constructors are executed, in which order, and with which arguments?

   (b) In the context of the Java language, what is an interface? How does an interface differ from an abstract class?

   (c) What is type erasure, and how does it restrict generic programming in the Java language? (A brief answer is sufficient; you are not asked to list explicitly all the restrictions imposed by Java.)

4. Consider the following Java class:

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

    public void m()
    {
      System.out.println(++v);
    }
}
```

   (a) Write a Java program that instantiates the class `C`, and then creates two concurrent threads that repeatedly invoke the method `m()` of the resulting object.

   (b) What can be printed during the execution of this program?

   (c) Would adding the attribute `synchronized` to the definition of the method `m()` change your answer to (b)?

   (Your answers to the questions (b) and (c) must be justified.)