Object-Oriented Programming
May 2018

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 `LineSegment` suited for representing a line segment in a 3D coordinate system. An instance of this class is characterized by two points $P_1 = (x_1, y_1, z_1)$ and $P_2 = (x_2, y_2, z_2)$ with integer coordinates x_1, x_2, y_1, y_2, z_1 and z_2, representing the endpoints of the segment $[P_1 P_2]$. These endpoints must be different, i.e., $(x_1, y_1, z_1) \neq (x_2, y_2, z_2)$, in order for the segment to be valid. The order of the endpoints of a segment is irrelevant, i.e., the segments $[P_1 P_2]$ and $[P_2 P_1]$ are considered to be identical.

   The class `LineSegment` should satisfy the following requirements:

   - The coordinates of the endpoints of a segment are provided when this segment is instantiated.

   - It must be possible to translate a segment by a given vector (dx, dy, dz), i.e., to turn its endpoints (x_1, y_1, z_1) and (x_2, y_2, z_2) into respectively $(x_1 + dx, y_1 + dy, z_1 + dz)$ and $(x_2 + dx, y_2 + dy, z_2 + dz)$.

   - It must be possible to split a segment $[P_1 P_2]$ into its two halves $[P_1 M]$ and $[P_2 M]$, where M is the middle point of the segment, computed as

     $$M = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}, \frac{z_1 + z_2}{2} \right),$$

     where $P_1 = (x_1, y_1, z_1)$ and $P_2 = (x_2, y_2, z_2)$. This operation should return an array containing the two halves, represented as newly created segments.

     Notes:

     - The order of the elements in the returned array does not matter.
     - Since this operation will be carried out in integer arithmetic, it might produce an invalid (zero-length) segment. This special case must be detected and dealt with appropriately.

   - Instances of this 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. (All answers should be thoroughly justified.)

   (a) In object-oriented programming, what is the purpose of limiting the visibility of instance and class variables?

   (b) A programmer defines a subclass `FixedLineSegment` of the class `Line-Segment` considered in Problem 1, in which the possibility of translating a segment is disabled by raising an exception whenever the corresponding method is invoked. What is the particular application of inheritance used in this case? Is the substitution principle satisfied by this application?

   (c) Explain how to use the Java keywords `this` and `super` in order to influence the execution of constructors.

   (d) In a few words, what is the advantage of defining generic classes?

   (e) In Java, what are the two techniques for creating concurrent threads? What are their respective advantages?