Exercise 2 Deadline: 22 April, 2019.

C++ PROGRAMMING LANGUAGE

Simple geometric figures

University of Wrocław
Institute of Computer Science

Paweł Rzechonek

Prologue.

Euclidean geometry is a mathematical system attributed to Alexandrian Greek mathematician Euclid, which he described in his textbook on geometry: the Elements. Euclid's method consists in assuming a small set of intuitively appealing axioms, and deducing many other propositions (theorems) from these. Although many of Euclid's results had been stated by earlier mathematicians, Euclid was the first to show how these propositions could fit into a comprehensive deductive and logical system. The Elements begins with plane geometry, still taught in secondary school (high school) as the first axiomatic system and the first examples of formal proof. It goes on to the solid geometry of three dimensions. Much of the Elements states results of what are now called algebra and number theory, explained in geometrical language.

In Euclidean geometry, there are primary concepts, accepted without definitions: point, line, plane, and space. Euclid assumed that the point, line, plane and space have the following dimensions: zero, one, two and three.

In Euclid's geometry there are three basic transformations of the plane:

(i) shift (translation), consisting in moving all points of the plane by the same distance in a fixed direction;

(ii) rotation around a fixed point of all points on the plane;

(iii) axial symmetry.

Two figures are defined as equivalent, if one of them can be transformed into another by means of shifts, rotations and symmetrical reflections.

Task.

Define the classes `point`, `segment` and `triangle`, which will represent the point, line segment and triangle respectively on the Euclidean plane with the Cartesian coordinate system. The `point` class should contain two fields `x` and `y` of the texttt double type to store the coordinates. The `segment` class is supposed to represent an line segment on a plane bounded by two different points. The `triangle` class is supposed to represent a triangle on the plane designated by the three non-collinear points. Remember to encapsulation and hide the state of each object.

In the mentioned classes, define constructors (including the copy constructor), copying assignments and methods for moving geometric objects by a given vector and rotating them around a given point by a given angle.

In addition, define a global function that will count the distance between two points.

In the class segment, define the method for calculating the length of the line segment, the method for examining whether the given point lies on the segment and the method that returns the middle of the line segment. In addition, define global functions that will check whether two line segments are parallel and whether they are perpendicular.

In the class triplet, define the method for calculating the triangle's perimeter, the area of the triangle, the method for examining whether the given point lies within the triangle and the method returning the center of the triangle. In addition, define global functions that will check if one triangle contains the other.

At the end, write a program that reliably tests all operations on points, line segments, and triangles. All objects in your program should be created on the stack.

Prompt.

Whenever we encounter any errors, ambiguities or contradictions in the program, this should be signaled with the exceptions. Exceptional situations are reported by the throw instruction. For example, in the constructor of the segment class, you should signal an exception when both ends of the segment have the same coordinates. Let exceptions be invalid_argument objects:

```
if (p.coordx()==q.coordx() && p.coordy()==q.coordy())
    throw invalid_argument("you can not create a segment of length 0");
```

Notice.

Divide the program into header and source files. Put the main() function in a separate source file.