OU2 - Mandatory Exercise 2

ID1018

October 22, 2014

Methods and class libraries

Computations on a triangle

Determining the properties of a triangle

In various situations the properties of a triangle need to be determined. You may need to know the lengths of its sides and altitudes, its angles, circumference and area. In other cases the lengths of a triangle's bisectors and medians are needed, or the radius of the incircle or circumcircle or something else.

In order to determine one property one needs to know some other properties of the triangle. For example, if the lengths of the sides are known, the triangle's area and circumference can be calculated. It is also possible to determine the angles, the altitudes and other properties. Often a formula can be used by inserting the appropriate information and compute that which is needed.

One can create a collection of formulas for triangles. As soon as you need to compute something, choose the correct formula and use it. In this way computations on triangles are facilitated. You supply the parameters to the formula and the desired result is returned. You can even create a program unit with several methods. A method corresponds to a formula: the necessary data are given to the method and it returns the computed result. As soon as you need to compute something, you choose the appropriate method, call it, and receive the returned data. The mathematical formulas are embedded in the methods.

Various formula for a triangle can be found here:
`http://en.wikipedia.org/wiki/Triangle`.

Exercises on a triangle

1. Create the class `Triangle`. The class must contain several static methods that perform various computations on the properties of a triangle. Each method accept certain data, computes, and returns some other data. For example, a method may accept the length and altitude of a triangle's side and return the

area of the triangle. Another method may accept the lengths of the triangle's three sides and return the circumference, the area, one of the medians or bisectors, or something else.

2. In the case that the length of two sides and the angle between them are known, the length of the corresponding bisector can be computed with the formula:

$$bis = (2bc\cos(\alpha/2))/(b+c) \tag{1}$$

in which b and c are the lengths of the two sides, α is the angle between them and bis is the length of the bisector that splits the angle in two equal halves.

The class `Triangle` could contain a method that computes the length of a bisector:

```
// The bisector method accepts two sides and the angle (in
// radians) between these sides. The method returns the
// length of the corresponding bisector - the one which
// splits the given angle in two equals halves.
public static double bisector(double b, double c, double alpha)
{
  double p = 2 * b * c * Math.cos (alpha / 2);
  double bis = p / (b + c);

  return bis;
}
```

Assume that all three lengths of the triangle and all three angles are known. Could you then by using method bisector determine the lengths of all three bisectors? Perhaps there should be three separate methods in class `Triangle` — one per bisector?

3. A program called `TriangleAndItsCircles` is to be created. The program shall read the lengths of a triangle's three sides and then determine the radius of the circumcircle and the radius of the incircle. For these computations, suitable methods in class `Triangle` must be used; call the method that computes the radius of the circumcircle and the method that computes the radius of the incircle.

4. Draw a triangle with the side lengths that were used at an execution of the program `TriangleAndItsCircles`. Draw also the two circles; the circumcircle and the incircle. Measure the radii of the circles with a ruler and verify that the lengths correspond to the output of the `TriangleAndItsCircles` program.