Exercise 10: Modelling strategies for inheritance hierarchies

In this exercise we will be looking at the two modelling strategies *bottom-up* and *top-down* as discussed in the lecture.

Task 1: Bottom-up, refactoring

We have the already modelled classes `Bicycle`, `Motorcycle`, `Car` and `Lorry`, as well as the parts `CantileverBrake`, `DiscBrake` and `DrumBrake`.

- Analyse the classes and look for similarities and/or differences.
- Design an appropriate class hierarchy in UML; what are the appropriate base classes and interfaces?
- Implement the modelled components and adapt the existing classes accordingly, thereby paying particular attention to the correct use of constructors. In doing so, the functionality of the given classes becomes smaller and shifts further up in the class hierarchy.

Please note: A suitable approach is to use a multi-layer class hierarchy for the vehicles and an interface for the brakes. You have done a good job if there is no longer any redundant code.

Task 2: Top-down, modelling a class hierarchy

For the programme given in the `DrawingProgramme` class, circles, triangles and rectangles should be modelled. These should be drawn later using an X/Y coordinate system.

- Which attributes are required for all shapes, and which are special? Which attributes can be calculated, and which must be specified?
- Draw a UML diagram of your class hierarchy.
- Implement your class hierarchy. Implement the `toString` method for each class; pay particular attention to the correct use of constructors.
- Complete the `main` method of the `DrawingProgramme` class according to the comments.

Sample output of the finished programme:

```
Circle around (5, 10) with radius 5

Rectangle from (0, 0) to (3, 2) with width 3 and height 2

Triangle at (0, 0), (1, 0) and (0, 1) with side lengths of 1, 1 and
1.4142135623730951.
```