Classes, inheritance and Makefiles

Homework

teacher: Katja Mankinen, katja.mankinen@hep.lu.se

Handed out: 10 October 2018
Due: 21 October 2018

Description

This week your homework is to extend the `shape` class used in the lecture. Two additional classes should be implemented. The `circle` class should derive directly from `shape`. Its constructor should take the radius of the circle as an argument. The `square` class should derive from `rectangle`, and its constructor should take the side of the square as an argument. Implement getters and setters for the radius of the `circle` and the side of the `square`. You should then implement a new function `circumference` for all the various shapes. It should, as you guessed, return the circumference of the shape.

Figure 1 shows the appearence of the `circle` and `triangle`. Notably, you may assume that the triangle is right-angled when calculating its circumference.

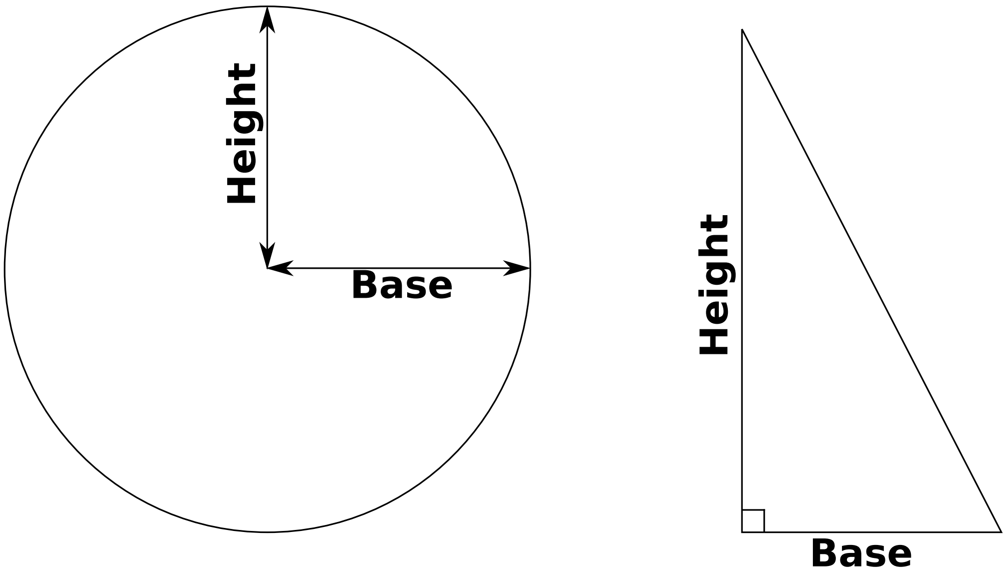

Figure 1: The appearence of the `circle` and `triangle` classes that derive from `shape`. The `base` and `height` of the `shape` class correspond to the radius of the circle. The `triangle` is assumed to be right angled.

The file `defaultTest.cpp` contains the example shown in the lecture. You can use it as a sanity check. The file `newTest.cpp` contains a program that will test the new shapes. Once you have implemented `circle` and `square`, uncomment the corresponding lines in `newTest.cpp` and confirm that the output matches what is shown in Table 1.

Table 1: The properties of the shapes used in `newTest.cpp`.

	Triangle	Rectangle	Square	Circle
Base	10	10	7	5
Height	5	5	7	5
Area	25	50	49	78.5
Circumference	26.2	30	28	31.4
Big enough	X	X	X	✓

Compiling `defaultTest.cpp` and `newTest.cpp` via command line is cumbersome due to the many files involved. In the real world, a `make` tool is used for anything but the most trivial of programs. The `make` tool builds programs using information from a `Makefile`. Read through the provided `Makefile` and try to understand what it does. To use it, you can type `make defaultTest` or `make all` in a terminal to build the `defaultTest` executable. Then you can run it as before: `./defaultTest` When you have implemented the new shapes, edit the `Makefile` so that it builds also the `newTest` executable and run it: `./newTest`.

Hand-in procedure

Submit your solution to Live@Lund by 21 October 2018!

At the beginning of the file, write down the time (approximately) you spent on this homework, and the names of students you may have collaborated with. You need to complete the homework independently but of course you may discuss and think about the problems together.

Note: Homework is mandatory! Don't hesitate to ask help from teachers if you get stuck.