Perimeter Assignment: Part Three

Introduction

In this assignment you will find the largest perimeter over several shapes by examining several files representing shapes, calculating the size of the largest perimeter and also the name of the file with the largest perimeter. You will add new methods to the PerimeterAssignmentRunner class. You may find it helpful to review the DirectoryResource and FileResource documentation.

The PerimeterAssignmentRunner class already includes the printFileNames method. You should understand the following: The printFileNames method has no parameters and no return value, hence return type **void**. This method first creates a DirectoryResource. When this happens you are prompted to select a file or files. You can select a bunch of files together by clicking on the name of one file, and then hold down the shift key and select a second file. All the files between the first and second file will be highlighted. The code then **iterates over all the files you have selected** using a for loop and the selectedFiles method, printing out the filename for each file.

For this assignment, you will add or modify several methods in the PerimeterAssigmentRunner class. The Goals for this exercise are as follows

1. Complete writing the method getLargestPerimeterMultipleFiles that has no parameters. This method creates a DirectoryResource (so you can select multiple files) and then iterates over these files. For each File f, it converts the file into a FileResource with the line
FileResource fr = new FileResource(f);

Then it should create a Shape from the FileResource and calculate that shape's perimeter, and return the largest perimeter over all the shapes in the files you have selected.

2. Finish writing the void method testPerimeterMultipleFiles to call getLargestPerimeterMultipleFiles and to print out the largest such perimeter. This method has no parameters and no return value. You will select the files when you run this method (hint: see our documentation for the DirectoryResource class).

3. Finish writing the method getFileWithLargestPerimeter that has no parameters. This method should create its own Directory Resource, except that this new method returns the name of the File that has the largest such perimeter, so it has return type String.