/\* \*

**\* JAVA Shapes Program**

\* This program implements inheritance and polymorphism for Project 1.

\*

\* This program is a menu-driven program that allows the user to construct

\* various shapes and displays the volume and area of the shape.

\* The program uses the abstract classes Shape, TwoDimensionalShape, and ThreeDimensionalShape

\* and its concrete subclasses to construct the shapes.

\* All shapes are stored in an ArrayList and displayed at the end of the program.

**\* To run the program, build and run the ShapeMenu class.**

\*

\* Author: Christian D Kirk

\* Date: 2023-03-29

\* Class: CMSC 335

\* Professor: Dr. Douglas Mujeye

\*

\* \*/

Documentation:

The code represents a ShapeMenu program that allows the user to choose a shape and input its properties to construct it. It includes classes for each shape, which inherit from a Shape superclass. The TwoDimensionalShape and ThreeDimensionalShape classes are subclasses of the Shape superclass that represent shapes with two and three dimensions, respectively.

The program begins by displaying a menu of shapes, prompting the user to enter a number from 1 to 10 to choose a shape or exit the program. The program uses a switch statement to call the appropriate method to construct the selected shape.

Each construction method prompts the user to input the necessary parameters for the shape, creates an instance of the appropriate shape class, and prints its details.

The Shape superclass has a single instance variable, numberOfDimensions, which is used to differentiate between two-dimensional and three-dimensional shapes.

The TwoDimensionalShape and ThreeDimensionalShape classes each have an instance variable for the area or volume, respectively, which is set in their constructors. They also override the toString() method to include the area or volume in the output.

**To execute the program, ensure all java files are in the same folder. Build and run the program using the ShapeMenu java file. Follow the instructions and use the console to input selections and numbers to the program.**

Lessons learned:

I honestly didn’t even know what a torus was until this program so I learned how to calculate the volume of a torus. This project was a great way to demonstrate inheritance and polymorphism. I felt that I was able to effectively use OOP concepts and good programming practices.

To make this program better, I feel that I should make use of the numberOfDimensions variable. I wrote it thinking I was going to use it but it ended up being unnecessary for my implementation. I think it would still be good for future classification of shapes and creating a table of inputted shapes by dimensions. It could also be a basis for a numberOfSides variable as well.

UML Diagram:

Graphical user interface

Description automatically generated