Geometric Calculator

Basmala Abuzied, Tasbeeh Ismail, Doaa Alaa, Reem Elsayed,
Amira Abdelrahman, Omnia Gamal, Sabah Ahmed

Key words: App, OOP, GUI, Java, javafx, Geometric Shapes.

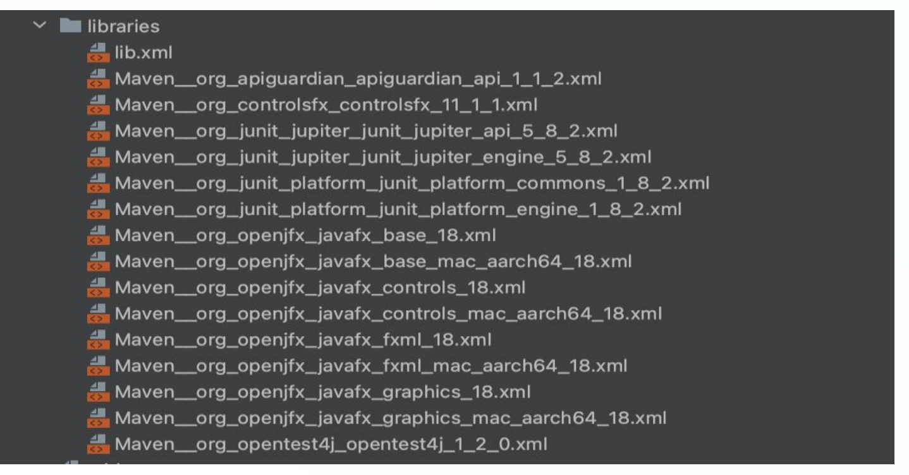
ABSTRACT

It's a GUI-based project used with the Java fx module and Using object-oriented programming concepts to make a Geometric Calculator project to calculate the circumference, area and volume of geometric shapes.

INTRODUCTION

We are going to develop a Geometric calculator Project in Java. We will be creating a GUI interface using Java fx package. The purpose of this poster is to provide a Geometric calculator using the concepts of the Object-Oriented Programming (OOP). It outlines the benefits of using OOP, and the steps involved in developing an OOP project.

LIBRARY&FUNCTIONS



```
libraries
    lib.xml
    Maven__org_apiguardian_apiguardian_api_1_1_2.xml
    Maven__org_controlsfx_controlsfx_11_1_1.xml
    Maven__org_junit_jupiter_junit_jupiter_api_5_8_2.xml
    Maven__org_junit_jupiter_junit_jupiter_engine_5_8_2.xml
    Maven__org_junit_platform_junit_platform_commons_1_8_2.xml
    Maven__org_junit_platform_junit_platform_engine_1_8_2.xml
    Maven__org_openjfx_javafx_base_18.xml
    Maven__org_openjfx_javafx_base_mac_aarch64_18.xml
    Maven__org_openjfx_javafx_controls_18.xml
    Maven__org_openjfx_javafx_controls_mac_aarch64_18.xml
    Maven__org_openjfx_javafx_fxml_18.xml
    Maven__org_openjfx_javafx_fxml_mac_aarch64_18.xml
    Maven__org_openjfx_javafx_graphics_18.xml
    Maven__org_openjfx_javafx_graphics_mac_aarch64_18.xml
    Maven__org_opentest4j_opentest4j_1_2_0.xml
```

java-jdk contains all java libraries (we use Math, IOException and util) and javafx-sdk (we use fxml, scene, stage and application)

METHODS

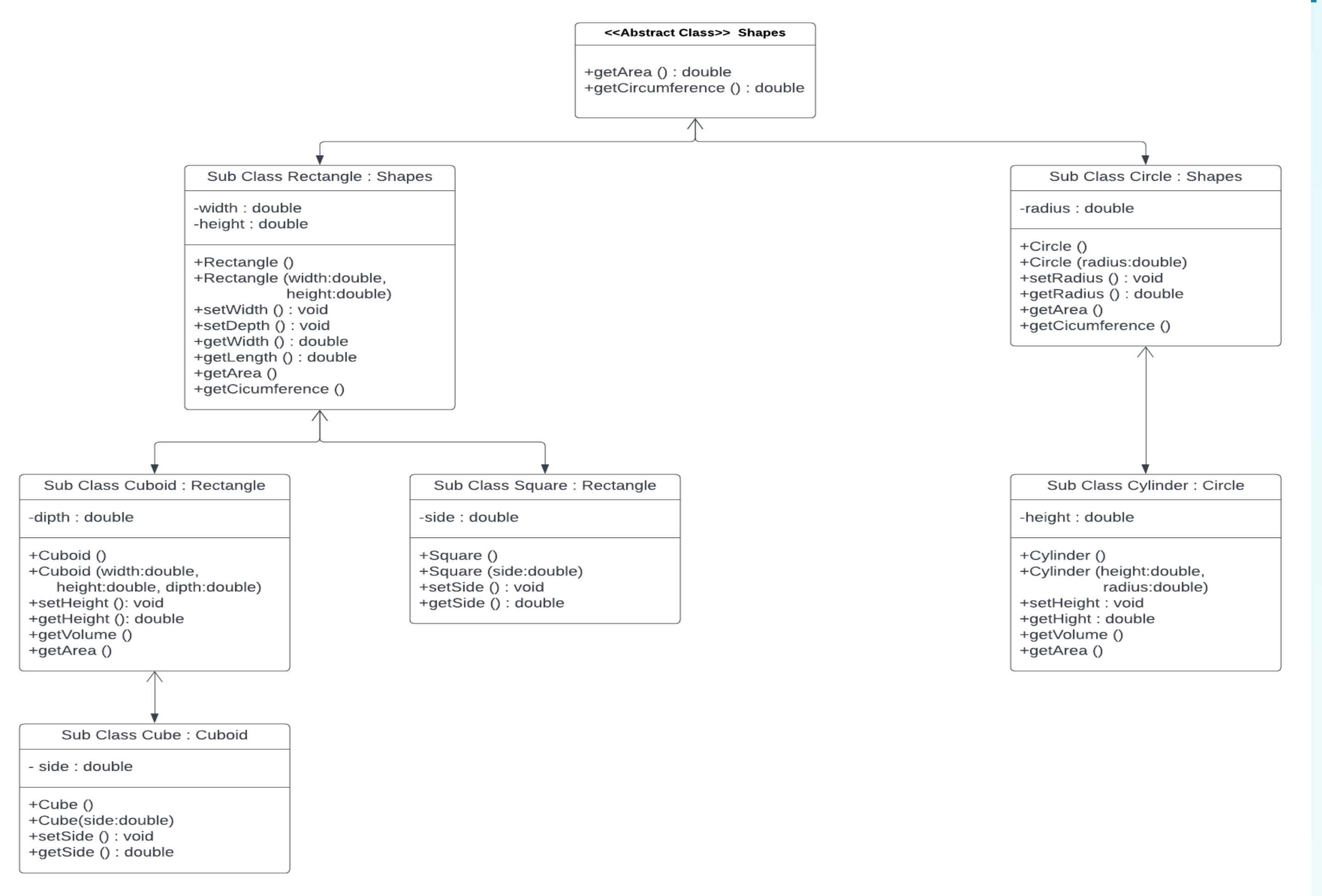
An abstract class named shapes that contains two abstract methods getArea(), and getCircumference().
-Two subclasses named Rectangle and Circle such that each one of the classes extends the superclass shape.
•The subclass Rectangle contains two variables to store width and height of the Rectangle, default constructor, constructor that initialize width and height of the Rectangle, public methods to set and get the two variables, and overriding two methods getArea() that returns rectangle's area, and getCircumference() that returns rectangle's circumference.
-Two subclasses named Cuboid and Square such that each one of the classes extends the superclass Rectangle.
*The subclass Cuboid contains a variable to store the dipth of the cuboid, default constructor, constructor that initialize width, height, and dipth of the Cuboid, public methods to set and get the height of the cuboid, overriding methods getArea() that returns cuboid's area, and method getVolume() that returns cuboid's volume.
*The subclass Square contains a variable to store the side of the Square, default constructor, constructor that initialize side of the Square, and public methods to set and get the side of the Square.
-A subclass named Cube extends the superclass Cuboid. The subclass Cube contains a variable to store the side of the cube, default constructor, constructor that initialize side of the Cube, and public methods to set and get the side of the cube.
•The subclass Circle contains a variable to store the radius of the circle, default constructor, constructor that initialize radius of the Circle, public methods to set and get the radius of the circle, and overriding two methods getArea() that returns circle's area and getCircumference() that returns circle's circumference.
-A subclass named Cylinder extends the superclass Circle.
The subclass Cylinder contains a variable to store the height of the cylinder, default constructor, constructor that initialize radius and height of the Cylinder, public methods to set and get the height of the cylinder, public method getVolume() that returns cylinder's volume, and overriding the method getArea() that returns cylinder's area.

ANALYSIS

The analysis for a geometric shapes OOP project should focus on the different shapes that will be used in the project. It should also consider the different properties of each shape, such as area, perimeter, and volume. Additionally, the analysis should consider the different methods that will be used to manipulate the shapes, such as scaling, rotating, and translating. Finally, the analysis should consider the different ways in which the shapes can be used, such as in a game or a drawing program.

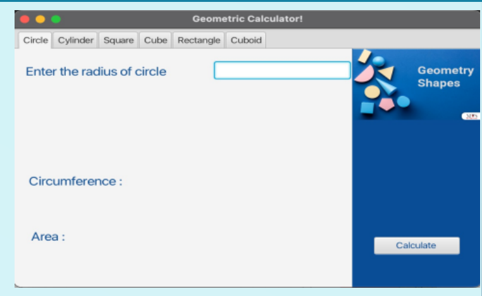
1. Identify the problem to be solved: Determine the properties of various geometric shapes.
2. Design the classes: Create classes for each type of geometric shape, such as Abstract class Shapes and Sub Classes as Rectangle, Circle, Square, Cylinder, Cube and Cupoid.
3. Define the inhertance classes: Rectangle and Circle inhert from shapes, then Cylinder inhert from Circle On the other side Square and Cupoid inhert from Rectangle. finally, Cube inhert from Cupoid. Define the methods: Define the methods for each class, such as calculating the area, valume, circumference , etc.
4. Define the attributes: Define the attributes of each class, such as radius, side, length, width, etc.
5. Define the methods: Define the methods for each class, such as calculating the area, valume, circumference , etc.
6. Test the code: Test the code to ensure that it is working correctly.
7. Document the code: Document the code to make it easier to understand and maintain.

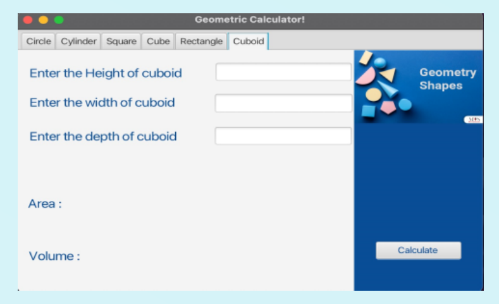
CLASSES DIAGRAM

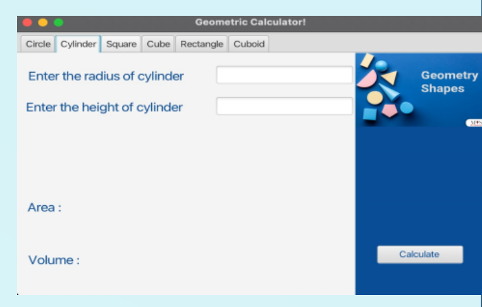


RESULTS

The user will choose the geometric shape (circle, cylinder, cube, square, rectangle or cuboid) whose area and circumference or volume he wants to calculate. After entering the required data, the project (geometric calculator) will calculate the required according to the inputs.







CONCLUSION

The poster for the Geometric Shapes OOP project is designed to be visually appealing and informative. It features a colorful background with a variety of geometric shapes in different colors. The shapes are arranged in a pattern that creates a sense of movement and energy. The poster also includes a brief description of the project, as well as a list of the different shapes that can be created with the project. The poster also includes a link to the project's website, which provides more information about the project and how to get started. Overall, the poster is designed to be eye-catching and informative, and it effectively conveys the purpose of the project. In the end, we were able to create a project that calculates the area and perimeter of two-dimensional geometric shapes and the volume of three-dimensional geometric shapes.

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to:
Dr. Shaymaa Sorour
Eng. Ahmed Saleh

FOR FURTHER INFORMATION

If you need more information you can contact us at:
s30303201800681@cis.dmu.edu.eg
s30309098800781@cis.dmu.edu.eg
s30208251802925@cis.dmu.edu.eg
s30309230200667@cis.dmu.edu.eg
s30301160201201@cis.dmu.edu.eg
s30305251802726@cis.dmu.edu.eg
s30305151802206@cis.dmu.edu.eg