GEOMETRY CALCULATOR

Object-Oriented Programming Project

Course Name: Object-Oriented Programming
Student Name: Furkan Akdağ
Student Number: 5001230010
Date: 29/12/2025

1. Introduction

The Geometry Calculator project is a console-based application developed in C++ using Object-Oriented Programming (OOP) principles. The main purpose of this project is to calculate geometric properties such as area, perimeter, surface area, and volume for various two-dimensional (2D) and three-dimensional (3D) shapes.

This project aims to demonstrate the effective use of abstraction, inheritance, and polymorphism by designing a modular and extensible class hierarchy. The application allows users to select different geometric shapes through a menu-driven interface and enter the required parameters to perform calculations.

2. Object-Oriented Design

The project is designed following core OOP principles to ensure code reusability, maintainability, and scalability.

2.1 Abstraction

Abstraction is achieved by defining abstract base classes that represent general geometric concepts.

- The **Shape** class is an abstract base class that defines a common interface for all shapes.
- The **Shape2D** class represents two-dimensional shapes and declares abstract methods for area and perimeter calculations.

- The **Shape3D** class represents three-dimensional shapes and declares abstract methods for surface area and volume calculations.

These abstract classes prevent direct instantiation and ensure that derived classes implement their specific calculation logic.

2.2 Inheritance

Inheritance is used to create a hierarchical relationship between classes.

- 2D shapes such as **Rectangle**, **Square**, **Circle**, **Triangle**, **Ellipse**, and **Parallelogram** inherit from the **Shape2D** class.
- 3D shapes such as **Cube** and **Sphere** inherit from the **Shape3D** class.

This structure allows common behaviors to be shared while enabling specialized implementations for each shape.

2.3 Polymorphism

Polymorphism is implemented through method overriding. Each concrete shape class provides its own implementation of the abstract methods defined in the base classes.

For example:

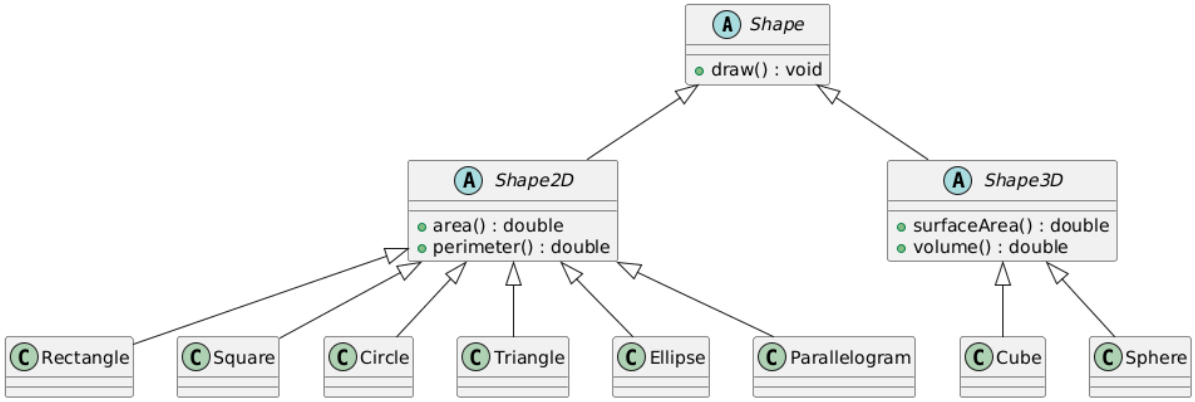
- The `area()` and `perimeter()` methods are implemented differently for each 2D shape.
- The `surfaceArea()` and `volume()` methods are implemented differently for each 3D shape.

This enables the program to treat all shapes uniformly while executing shape-specific behavior at runtime.

3. UML Class Diagram

The UML class diagram illustrates the structure of the project and the relationships between classes. It shows the abstract base classes, derived classes, and the inheritance hierarchy used in the application.

Figure 1: UML Class Diagram of the Geometry Calculator Project.



4. Program Flow

The program follows a menu-driven structure:

1. The user is presented with a menu containing 2D and 3D shape options.
2. The user selects a shape using a numeric choice.
3. The program requests the required input values for the selected shape.
4. The corresponding calculation methods are executed.
5. Results are displayed on the screen.
6. A do–while loop allows the user to continue using the program until exit is selected.

A `switch-case` structure is used to handle user selections efficiently.

5. Technologies Used

- **Programming Language:** C++
- **Development Environment:** Code::Blocks
- **Design Tool:** PlantUML
- **Version Control:** GitHub

6. Conclusion

In conclusion, the Geometry Calculator project successfully demonstrates the application of Object-Oriented Programming principles in C++. By using abstraction,

inheritance, and polymorphism, the project achieves a clean and extensible design. The modular structure allows new shapes to be added easily, making the application scalable for future improvements.

This project provides a solid example of how OOP concepts can be applied to real-world programming problems in an organized and effective manner.

7. GitHub Repository

The source code and related project files are available on GitHub:

GitHub Repository Link:

https://github.com/furkanakdag01/Geometry_Calculator