PROJECT REPORT

Course: Object-Oriented Programming
Instructor: Dr. Ezgi ZORARPACI
Project Topic: Shape Drawing / Geometry Calculator
Submission Date: 28.12.2025

Team Members:
Metehan BAYRAK - 5001240044
Harun TURAN - 5001230026
Arda INCEKAR - 5001240015

1. Project Purpose and Features

1.1 Purpose

The primary objective of this project is to design and implement a geometry calculator application using C++. The system demonstrates core Object-Oriented Programming (OOP) principles such as abstraction, inheritance, and polymorphism. It allows users to manage different geometric shapes dynamically and perform calculations like area and perimeter without knowing the internal implementation details of each shape.

1.2 Features

- Dynamic Shape Management: Users can add multiple shapes (Rectangles and Squares) to a centralized list.
- Polymorphic Calculations: The system automatically selects the correct formula for Area and Perimeter based on the shape type.
- Object Tracking: A static counter tracks the total number of active shape objects in memory.
- Memory Management: The project utilizes pointers and vectors (std::vector<Sekil*>) to manage memory dynamically, ensuring proper cleanup.
- Input Validation: The system prevents invalid dimensions by setting default values.

2. Class Design

The project is built upon a hierarchical class structure. Below is the description of the classes designed for this system.

2.1 Class Hierarchy

Sekil (Abstract Base Class):

- Acts as an interface for all shapes.
- Contains pure virtual functions: alanHesapla(), cevreHesapla(), sekilIsim().
- Includes a virtual destructor.

Dikdortgen (Derived from Sekil):

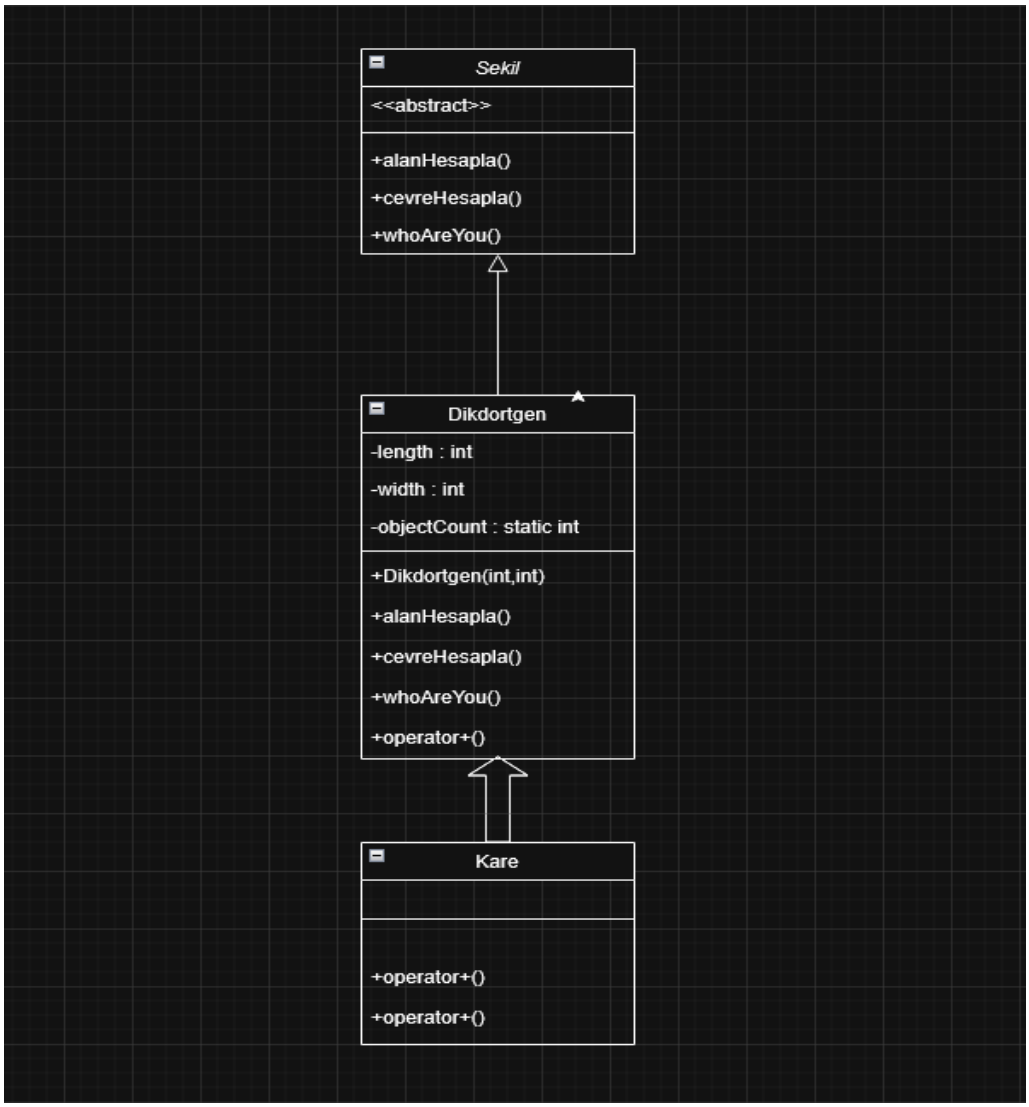
- Represents a rectangle with length and width attributes.
- Implements virtual functions from Sekil.
- Uses encapsulation with private attributes.
- Contains a static object counter.
- Includes operator overloading (operator+).

Kare (Derived from Dikdortgen):

- Represents a square.
- Overrides setters to ensure equal sides.

2.2 UML Diagram

The UML diagram illustrates the inheritance hierarchy: Sekil as the base class, Dikdortgen derived from Sekil, and Kare derived from Dikdortgen.

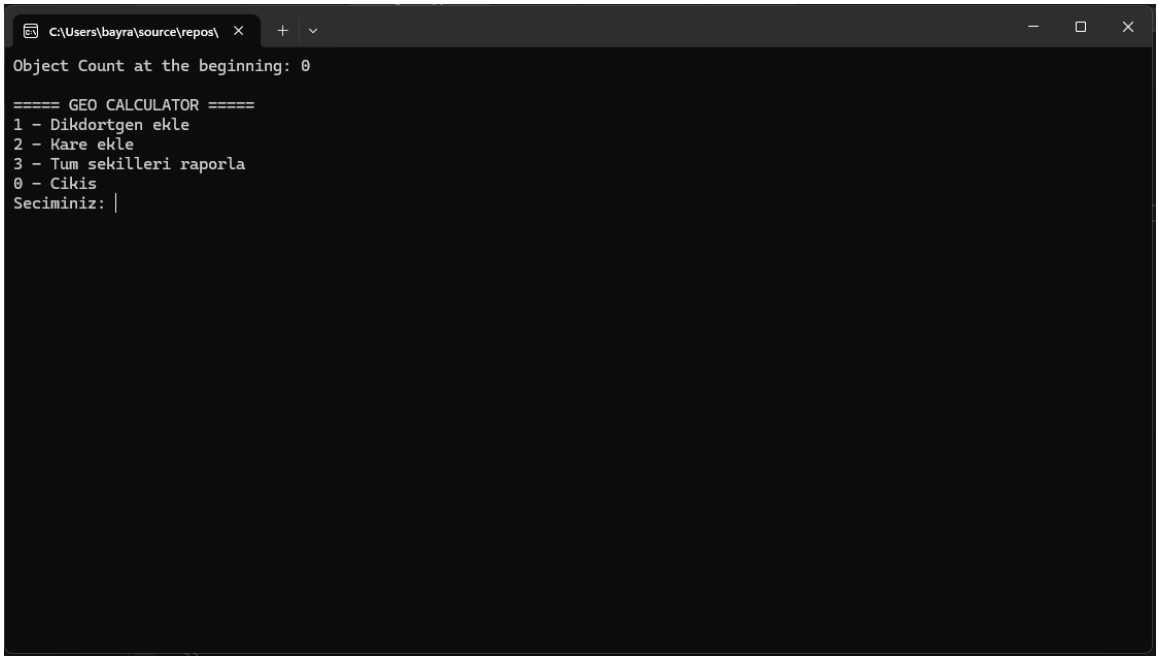


3. Key OOP Concepts Used

1. Abstraction: Implemented using the abstract class Sekil with pure virtual functions.
2. Inheritance: Dikdortgen inherits from Sekil, Kare inherits from Dikdortgen.
3. Polymorphism: Achieved using std::vector<Sekil*> and runtime binding.
4. Encapsulation: Private attributes with public setters and getters.

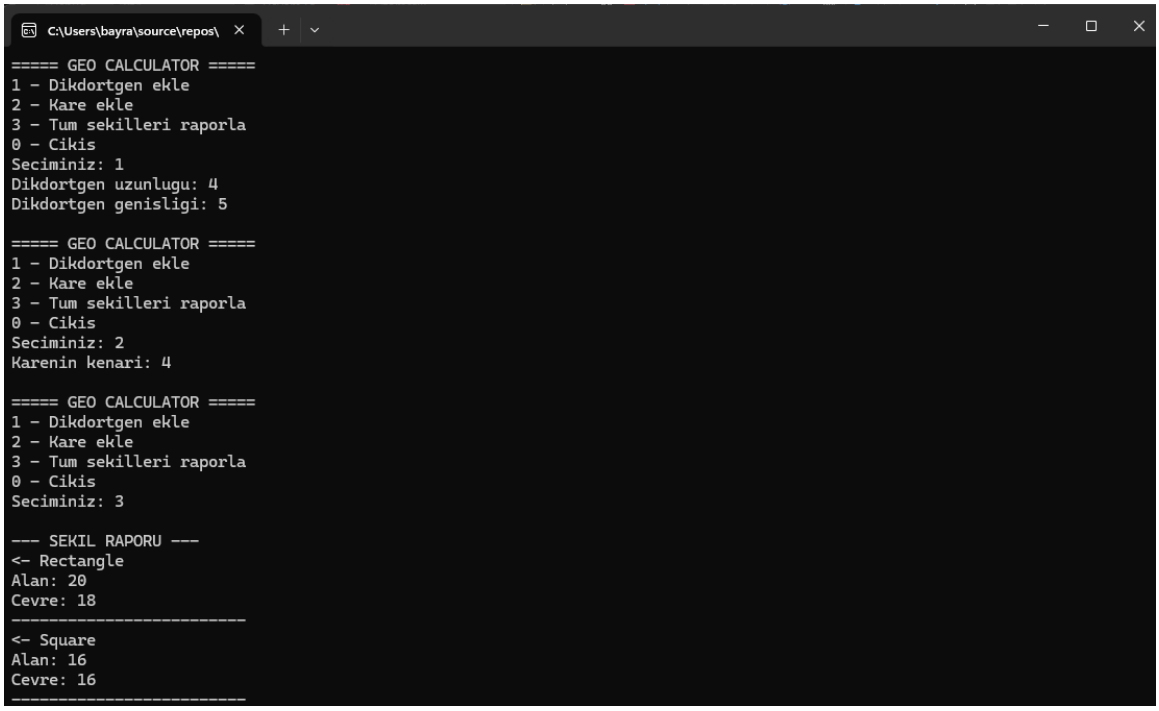
4. Execution Demo

Step 1: Application Menu and Object Counting.



Step 2: Adding Shapes (Rectangle and Square).

Step 3: Polymorphic Reporting.



5. Conclusion

In this project, we successfully developed a modular geometry calculator using C++. We gained practical experience in class design, inheritance, polymorphism, dynamic memory management, and encapsulation. The project meets all assignment requirements.

6.GitHub Link

https://github.com/mbayrak29/geo-calculator-oop.git

7.Referances

This project is fueled by excessive coffee and a little bit of GPT assistance. 😊