**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ТЕЛЕКОММУНИКАЦИЙ КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ**

**Лабораторная работа по дисциплине «ООП»**

Выполнил:

Хасензода Муборакшох Латиф

Студент 2 курса группы ПИН-б-о-22-1

Направления подготовки

09.03.03 Прикладная информатика

очной формы обучения

Ставрополь, 2023 г.

**Цель работы:** изучить базовые понятия (классы, подклассы и методы)

Реализовать фундаментальные принципы объектно-ориентированного

программирования.  
  
**Вариант 3.**

Построить систему классов для описания плоских геометрических

фигур: круг, квадрат, прямоугольник. Предусмотреть методы для создания

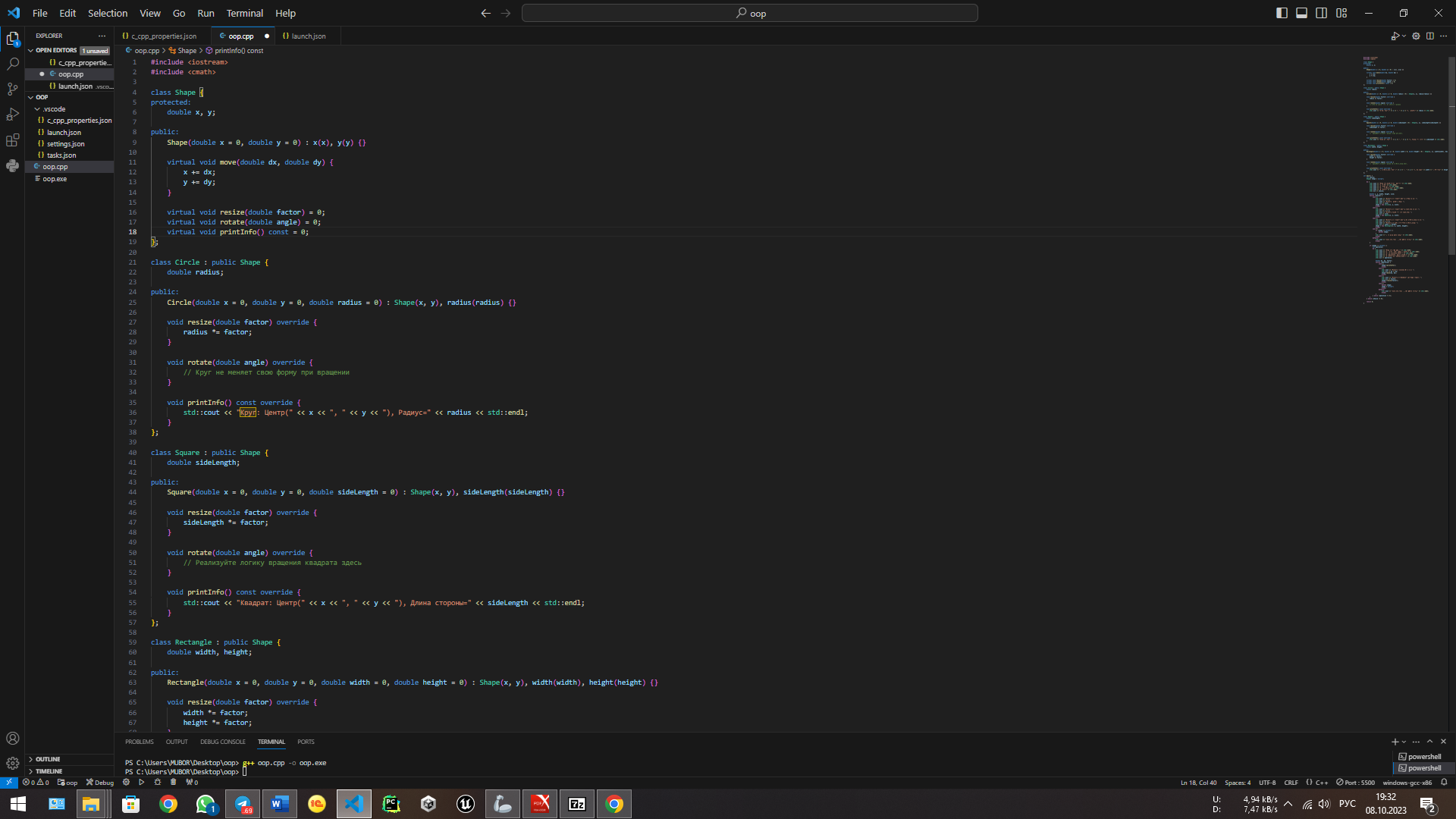
объектов, перемещения на плоскости, изменения размеров и вращения на

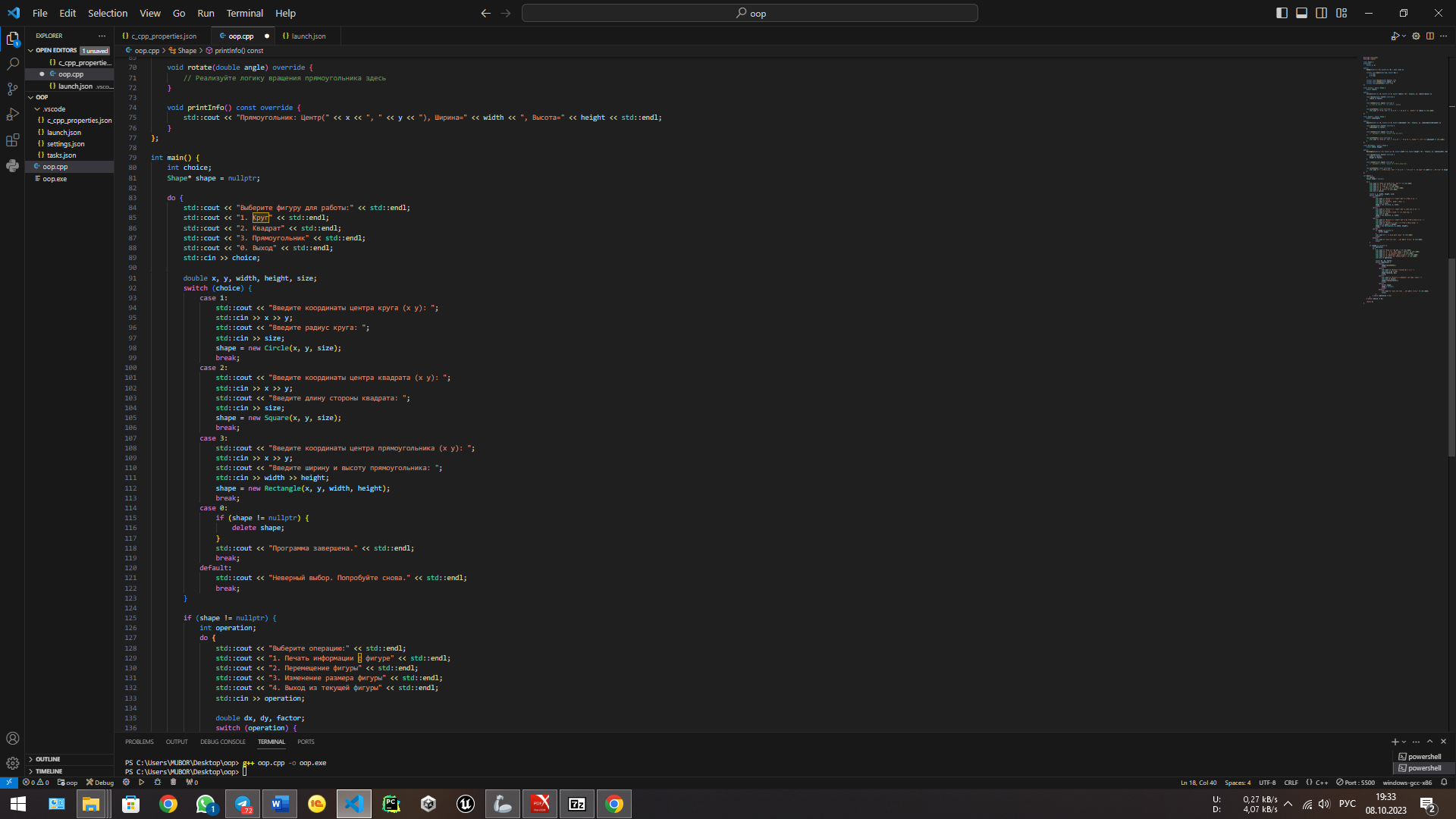
заданный угол.

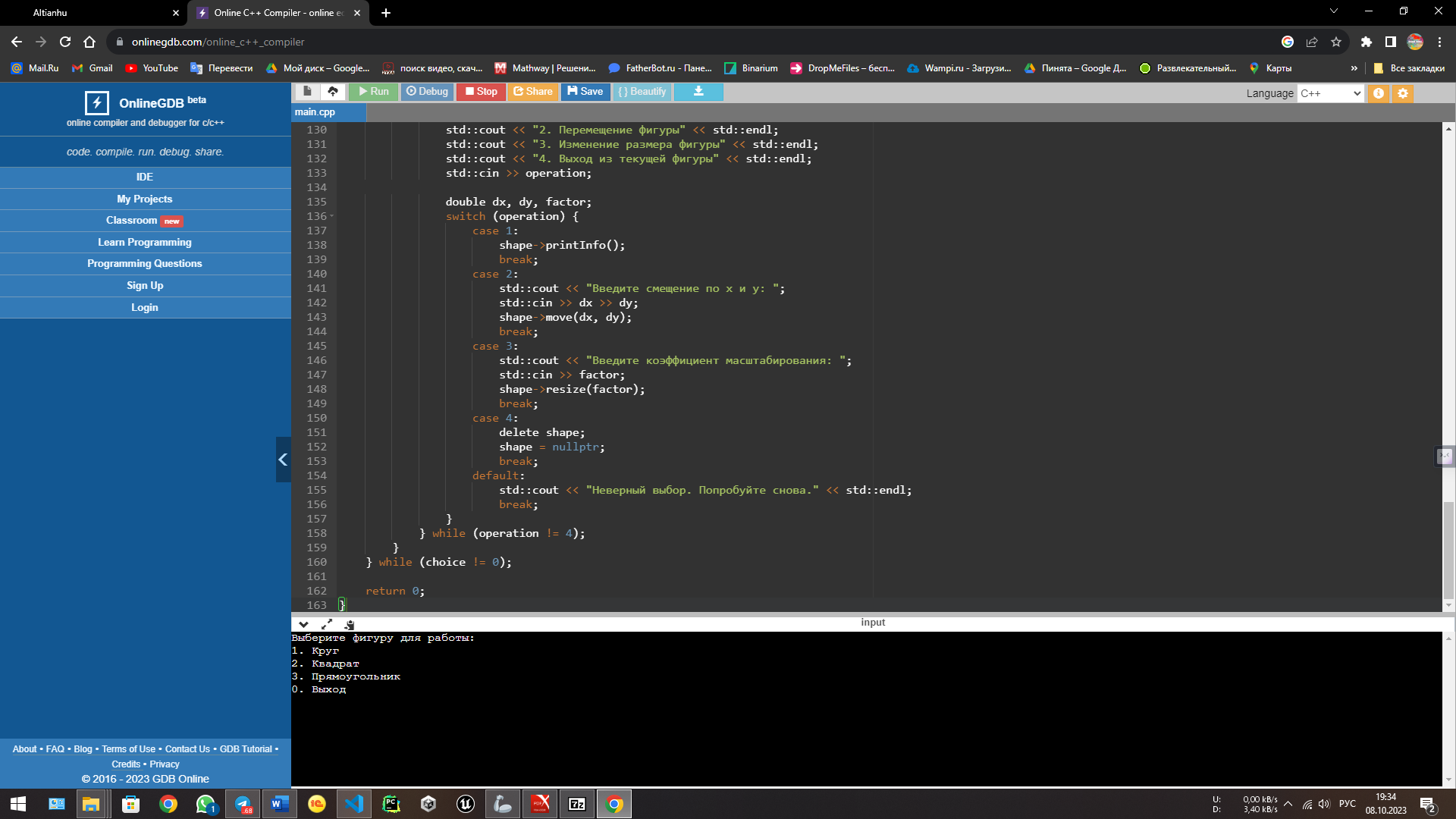
Написать программу, демонстрирующую работу с этими классами.

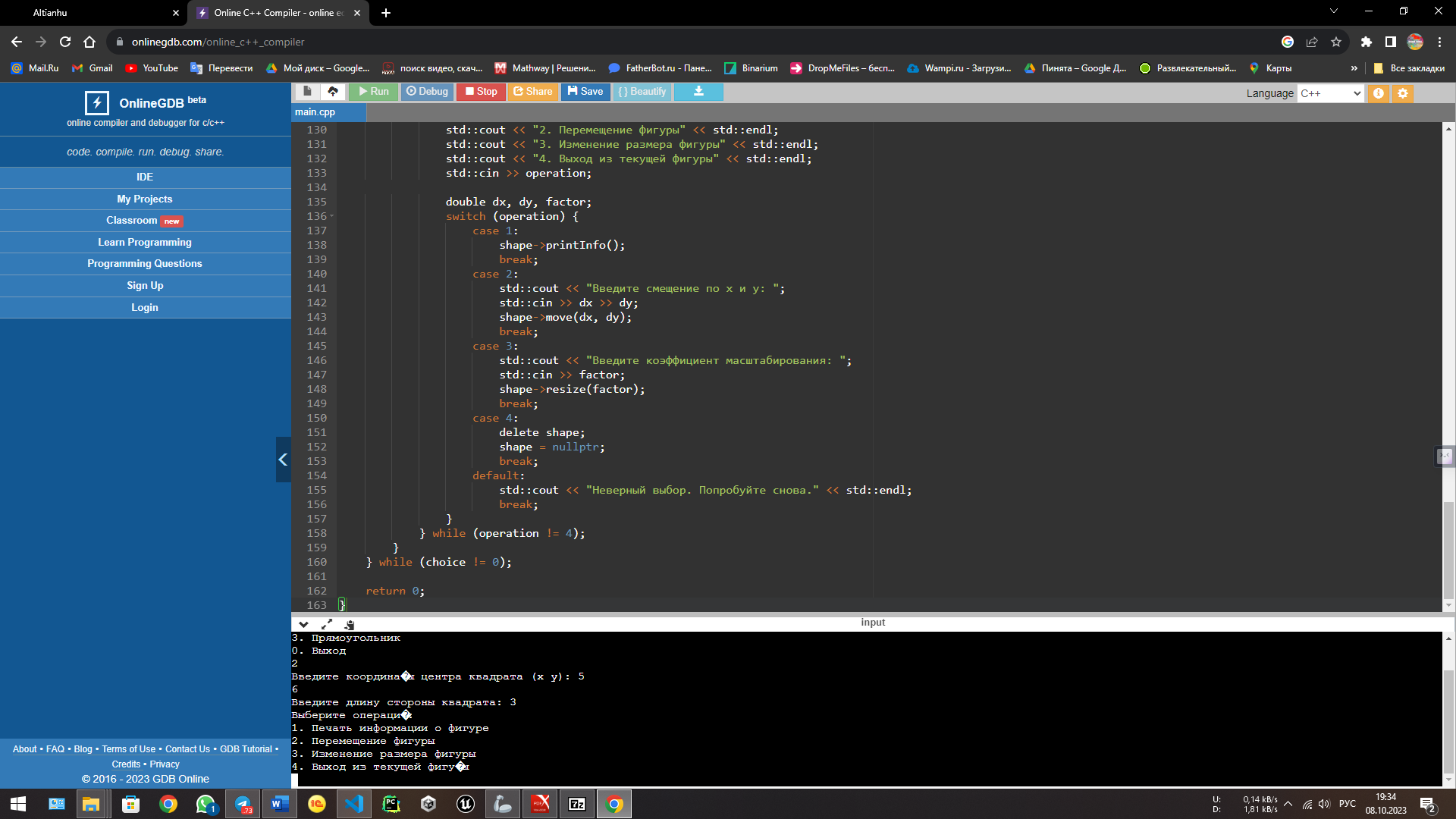
Программа должна содержать меню, позволяющее осуществить проверку всех

методов классов.









Код:  
#include <iostream>

#include <cmath>

class Shape {

protected:

    double x, y;

public:

    Shape(double x = 0, double y = 0) : x(x), y(y) {}

    virtual void move(double dx, double dy) {

        x += dx;

        y += dy;

    }

    virtual void resize(double factor) = 0;

    virtual void rotate(double angle) = 0;

    virtual void printInfo() const = 0;

};

class Circle : public Shape {

    double radius;

public:

    Circle(double x = 0, double y = 0, double radius = 0) : Shape(x, y), radius(radius) {}

    void resize(double factor) override {

        radius \*= factor;

    }

    void rotate(double angle) override {

        // Круг не меняет свою форму при вращении

    }

    void printInfo() const override {

        std::cout << "Круг: Центр(" << x << ", " << y << "), Радиус=" << radius << std::endl;

    }

};

class Square : public Shape {

    double sideLength;

public:

    Square(double x = 0, double y = 0, double sideLength = 0) : Shape(x, y), sideLength(sideLength) {}

    void resize(double factor) override {

        sideLength \*= factor;

    }

    void rotate(double angle) override {

        // Реализуйте логику вращения квадрата здесь

    }

    void printInfo() const override {

        std::cout << "Квадрат: Центр(" << x << ", " << y << "), Длина стороны=" << sideLength << std::endl;

    }

};

class Rectangle : public Shape {

    double width, height;

public:

    Rectangle(double x = 0, double y = 0, double width = 0, double height = 0) : Shape(x, y), width(width), height(height) {}

    void resize(double factor) override {

        width \*= factor;

        height \*= factor;

    }

    void rotate(double angle) override {

        // Реализуйте логику вращения прямоугольника здесь

    }

    void printInfo() const override {

        std::cout << "Прямоугольник: Центр(" << x << ", " << y << "), Ширина=" << width << ", Высота=" << height << std::endl;

    }

};

int main() {

    int choice;

    Shape\* shape = nullptr;

    do {

        std::cout << "Выберите фигуру для работы:" << std::endl;

        std::cout << "1. Круг" << std::endl;

        std::cout << "2. Квадрат" << std::endl;

        std::cout << "3. Прямоугольник" << std::endl;

        std::cout << "0. Выход" << std::endl;

        std::cin >> choice;

        double x, y, width, height, size;

        switch (choice) {

            case 1:

                std::cout << "Введите координаты центра круга (x y): ";

                std::cin >> x >> y;

                std::cout << "Введите радиус круга: ";

                std::cin >> size;

                shape = new Circle(x, y, size);

                break;

            case 2:

                std::cout << "Введите координаты центра квадрата (x y): ";

                std::cin >> x >> y;

                std::cout << "Введите длину стороны квадрата: ";

                std::cin >> size;

                shape = new Square(x, y, size);

                break;

            case 3:

                std::cout << "Введите координаты центра прямоугольника (x y): ";

                std::cin >> x >> y;

                std::cout << "Введите ширину и высоту прямоугольника: ";

                std::cin >> width >> height;

                shape = new Rectangle(x, y, width, height);

                break;

            case 0:

                if (shape != nullptr) {

                    delete shape;

                }

                std::cout << "Программа завершена." << std::endl;

                break;

            default:

                std::cout << "Неверный выбор. Попробуйте снова." << std::endl;

                break;

        }

        if (shape != nullptr) {

            int operation;

            do {

                std::cout << "Выберите операцию:" << std::endl;

                std::cout << "1. Печать информации о фигуре" << std::endl;

                std::cout << "2. Перемещение фигуры" << std::endl;

                std::cout << "3. Изменение размера фигуры" << std::endl;

                std::cout << "4. Выход из текущей фигуры" << std::endl;

                std::cin >> operation;

                double dx, dy, factor;

                switch (operation) {

                    case 1:

                        shape->printInfo();

                        break;

                    case 2:

                        std::cout << "Введите смещение по x и y: ";

                        std::cin >> dx >> dy;

                        shape->move(dx, dy);

                        break;

                    case 3:

                        std::cout << "Введите коэффициент масштабирования: ";

                        std::cin >> factor;

                        shape->resize(factor);

                        break;

                    case 4:

                        delete shape;

                        shape = nullptr;

                        break;

                    default:

                        std::cout << "Неверный выбор. Попробуйте снова." << std::endl;

                        break;

                }

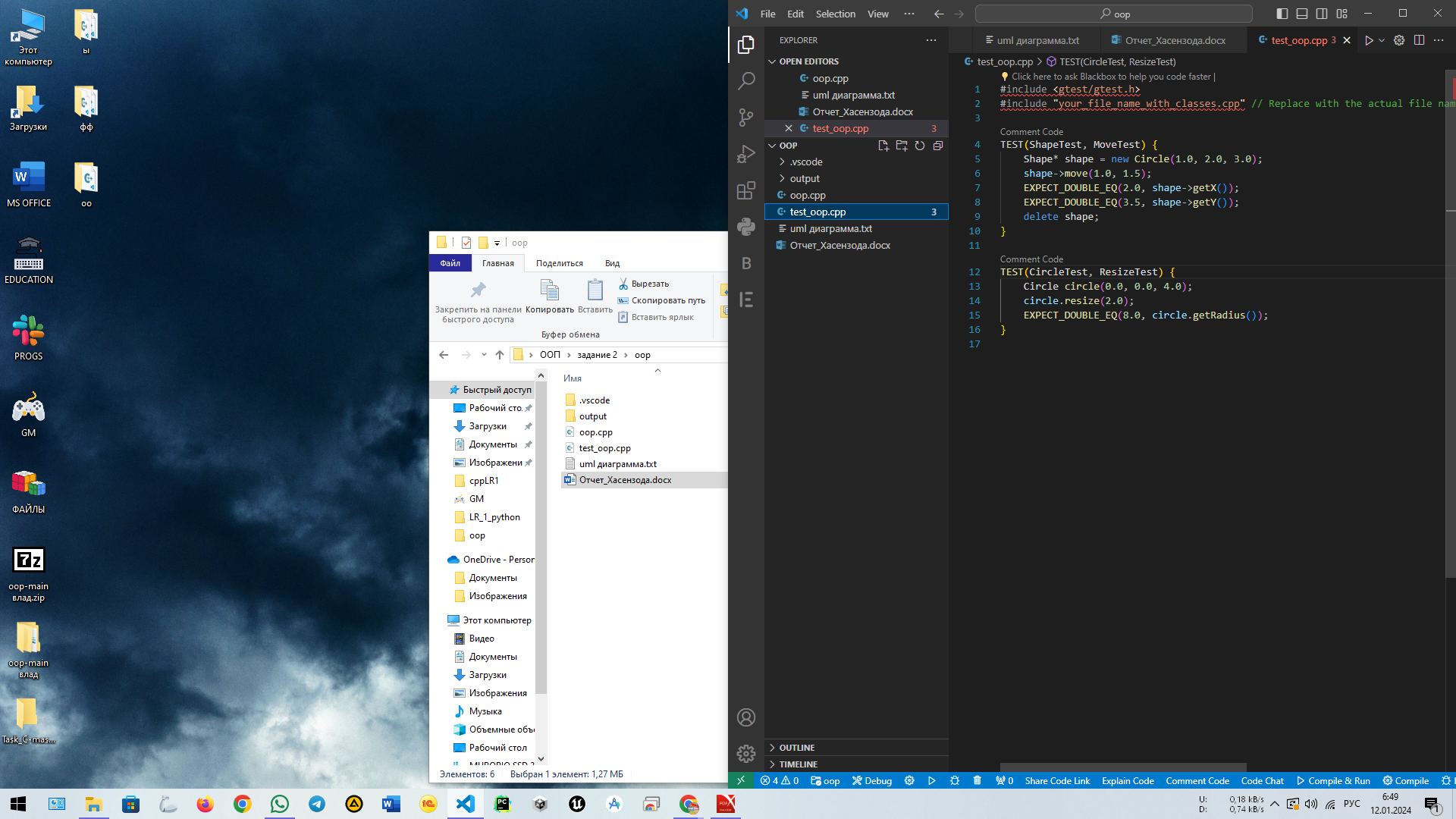
            } while (operation != 4);

        }

    } while (choice != 0);

    return 0;

}

Тест:  


Uml диаграмма:  
