

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**
(БГТУ им. В. Г. Шухова)

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Дисциплина: Объектно-ориентированное программирование

Лабораторная работа № 5

Тема: «Классы, виды отношений. Наследование»

Выполнил:

Студент группы ПВ-211

Пономарева Е. А.

Проверил:

Харитонов С. Д.

Белгород 2023 г.

Цель работы: получение теоретических знаний в области разработки классов, получение практических навыков реализаций классов и отношений между ними

Задание к работе: в соответствии с вариантом (((номер по списку +5)%10)+1) выполнить построение объектной модели (использовать не менее 5 объектов) заданной предметной области (задание 1), разработать диаграмму классов для описанной объектной модели (не менее 7 классов), и реализовать предложенные классы (задание 2)

Вариант №1

Вариант 1

Задание 1

Графический редактор.

Задание 2

1. Создать абстрактный класс Figure с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран.
2. Создать производные классы: Rectangle (прямоугольник), Circle (круг), Triangle (треугольник) со своими методами вычисления площади и периметра.
3. Создать массив n фигур и вывести полную информацию о фигурах на экран.

Объектная модель

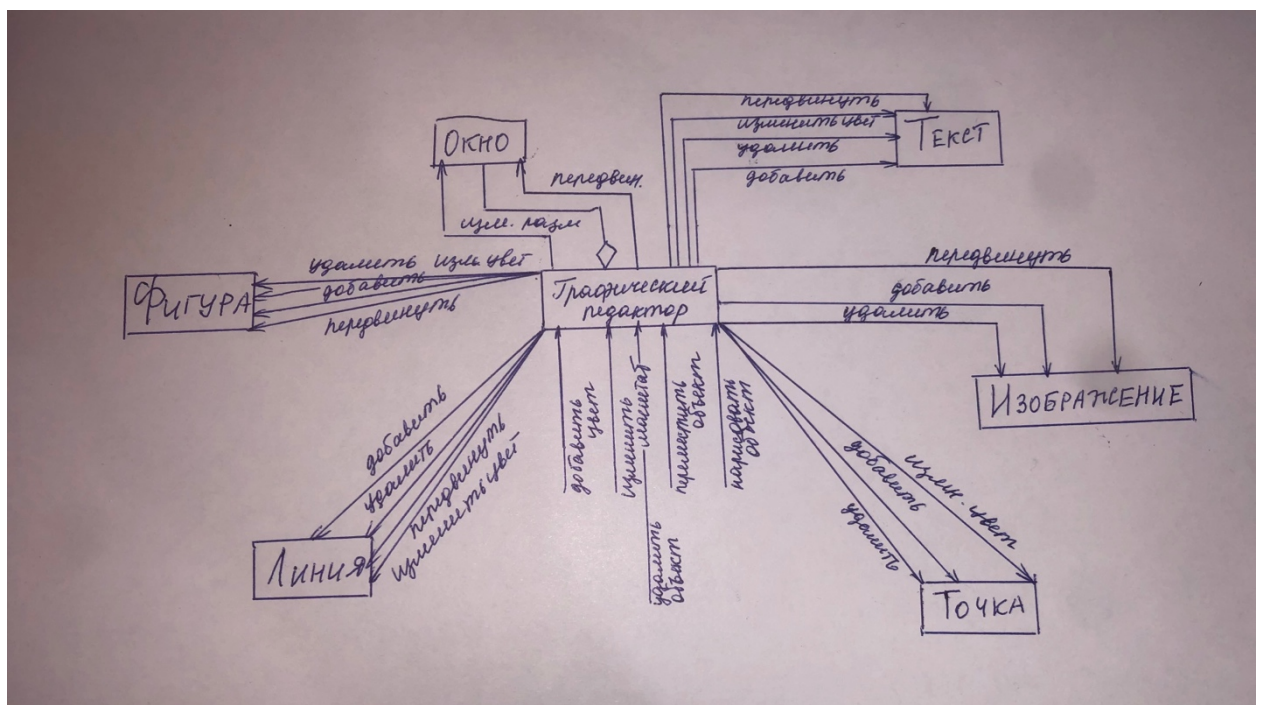
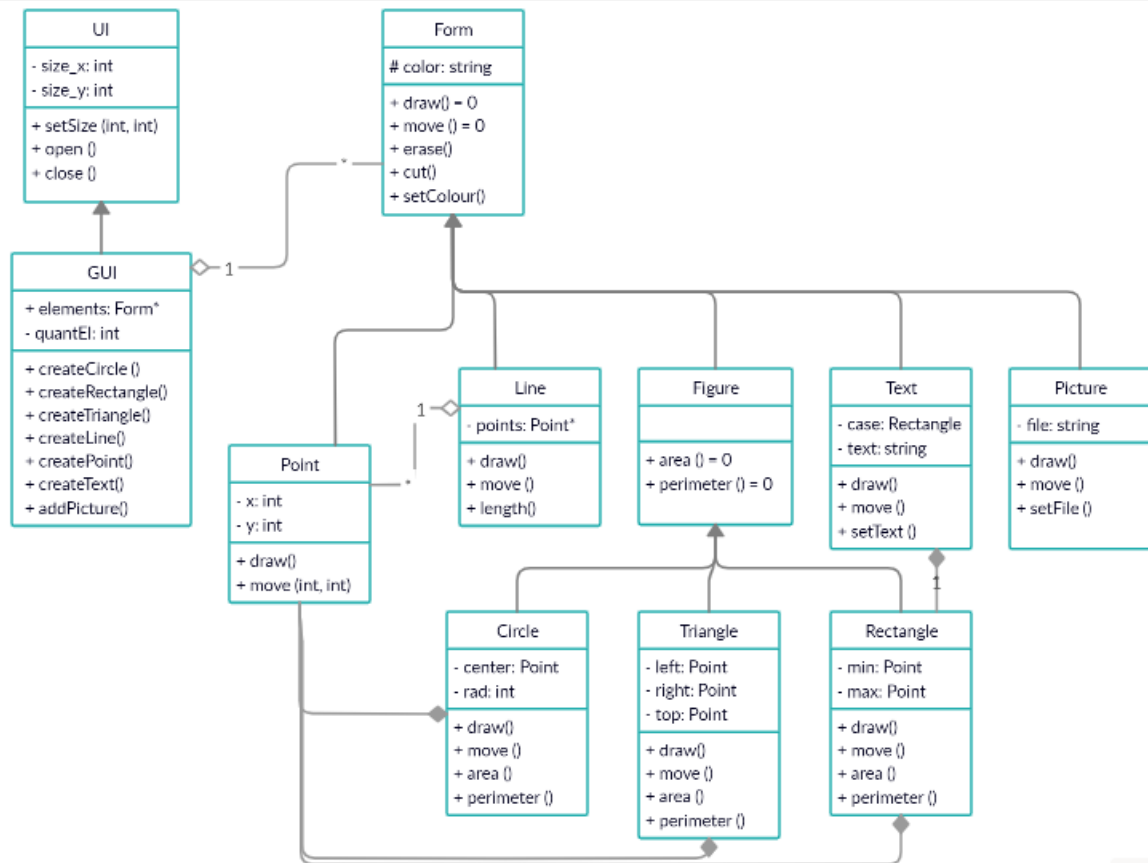


Диаграмма классов



Заголовочный файл (.h)

```

#ifndef FIGURE_H
#define FIGURE_H

class Figure
{
public:
    Figure();
    virtual ~Figure();
    virtual double area() = 0;
    virtual double perimeter() = 0;
    void outputInfo();
};

class Circle: public Figure
{
public:
    double x, y, radius;

    Circle();
    ~Circle();

    void setCenter(double, double);
    void setRadius(double);
    double area() override;
    double perimeter() override;
};

```

```

};

class Rectangle: public Figure
{
    double xMin, yMin, xMax, yMax;

public:
    Rectangle();
    ~Rectangle();

    void setMin(double, double);
    void setMax(double, double);
    double area() override;
    double perimeter() override;
};

class Triangle: public Figure
{
    double x1, y1, x2, y2, x3, y3;

public:
    Triangle();
    ~Triangle();

    void setXY1(double, double);
    void setXY2(double, double);
    void setXY3(double, double);
    double area() override;
    double perimeter() override;
};

```

Исполнительный файл (.cpp)

```

#include "figure.h"
#include <math.h>
#include <iostream>

Figure::Figure() {}

Figure::~~Figure() {}

void Figure::outputInfo()
{
    std::cout << "Area = " << area() << std::endl;
    std::cout << "Perimeter = " << perimeter() << std::endl;
}

// - Окружность

Circle::Circle()
{
    setCenter(0, 0);
    setRadius(0);
}

Circle::~~Circle() {}

void Circle::setCenter(double xBuf, double yBuf)
{
    x = xBuf;
    y = yBuf;
}

```

```

void Circle::setRadius(double rBuf)
{
    radius = rBuf;
}

double Circle::area()
{
    return M_PI*radius*radius;
}

double Circle::perimeter() // Длина окружности
{
    return 2*M_PI*radius;
}

// - Прямоугольник

Rectangle::Rectangle()
{
    setMax(0, 0);
    setMin(0, 0);
}

Rectangle::~Rectangle() {}

void Rectangle::setMax(double xBuf, double yBuf)
{
    xMax = xBuf;
    yMax = yBuf;
}

void Rectangle::setMin(double xBuf, double yBuf)
{
    xMin = xBuf;
    yMin = yBuf;
}

double Rectangle::area()
{
    return sqrt((xMax - xMin)*(xMax - xMin)) * sqrt((yMax - yMin)*(yMax - yMin));
}

double Rectangle::perimeter()
{
    return 2*sqrt((xMax - xMin)*(xMax - xMin)) + 2*sqrt((yMax - yMin)*(yMax - yMin));
}

// - Треугольник

Triangle::Triangle()
{
    setXY1(0, 0);
    setXY2(0, 0);
    setXY3(0, 0);
}

Triangle::~Triangle() {}

void Triangle::setXY1(double xBuf, double yBuf)
{
    x1 = xBuf;

```

```

        y1 = yBuf;
    }

void Triangle::setXY2(double xBuf, double yBuf)
{
    x2 = xBuf;
    y2 = yBuf;
}

void Triangle::setXY3(double xBuf, double yBuf)
{
    x3 = xBuf;
    y3 = yBuf;
}

double Triangle::area()
{
    double p = perimeter() / 2;
    return sqrt(p * (p - sqrt((x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1))) *
(p - sqrt((x3 - x2)*(x3 - x2) + (y3 - y2)*(y3 - y2))) * (p - sqrt((x1 -
x3)*(x1 - x3) + (y1 - y3)*(y1 - y3))));
}

double Triangle::perimeter()
{
    return sqrt((x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1)) + sqrt((x3 -
x2)*(x3 - x2) + (y3 - y2)*(y3 - y2)) + sqrt((x1 - x3)*(x1 - x3) + (y1 -
y3)*(y1 - y3));
}

```

Основная программа

```

#include <iostream>
#include <cstdlib>
#include "figure.h"
using namespace std;

int main()
{
    setlocale(LC_ALL, "rus");
    Figure** arr = NULL;
    int n;
    cout << "Пожалуйста, введите количество фигур" << endl;
    cin >> n;
    arr = new Figure*[n];

    for (int i = 0; i < n; i++)
    {
        char r;
        cout << "Пожалуйста, выберите фигуру (R - Прямоугольник, С -
Окружность, Т - Треугольник)" << endl;
        getchar();
        cin >> r;
        switch (r)
        {
            case 'R':
            {
                Rectangle* buf = new Rectangle;
                double x, y;
                cout << "Пожалуйста, введите xMin, yMin, xMax, yMax" << endl;
                cin >> x >> y;
                buf->setMin(x, y);
                cin >> x >> y;
                buf->setMax(x, y);
            }
        }
    }
}

```

```

        arr[i] = (Figure*) buf;
        break;
    }
    case 'C':
    {
        Circle* buf = new Circle;
        double x, y;
        cout << "Пожалуйста, введите x, y, radius" << endl;
        cin >> x >> y;
        buf->setCenter(x, y);
        cin >> x;
        buf->setRadius(x);
        arr[i] = (Figure*) buf;
        break;
    }
    case 'T':
    {
        Triangle* buf = new Triangle;
        double x, y;
        cout << "Пожалуйста, введите x1, y1, x2, y2, x3, y3" << endl;
        cin >> x >> y;
        buf->setXY1(x, y);
        cin >> x >> y;
        buf->setXY2(x, y);
        cin >> x >> y;
        buf->setXY3(x, y);
        arr[i] = (Figure*) buf;
        break;
    }
    default:
    {
        cout << "Неверная фигура, повторите" << endl;
        i--;
        continue;
    }
}

for (int i = 0; i < n; i++)
{
    cout << "Фигура " << i+1 << endl;
    arr[i]->outputInfo();
    delete arr[i]; // После получения информации сразу удаляем объект (для упрощения
кода)
}

delete [] arr;
system("pause");

return 0;
}

```

Результат работы программы:

```
/Users/evgeniaponomareva/CLionProjects/untitled/cmake-build-debug/untitled
Пожалуйста, введите количество фигур
3
Пожалуйста, выберите фигуру (R - Прямоугольник, C - Окружность, T - Треугольник)
R
Пожалуйста, введите xMin, yMin, xMax, yMax
1 4 5 3
Пожалуйста, выберите фигуру (R - Прямоугольник, C - Окружность, T - Треугольник)
C
Неверная фигура, повторите
Пожалуйста, выберите фигуру (R - Прямоугольник, C - Окружность, T - Треугольник)
C
Пожалуйста, введите x, y, radius
3 4 5
Пожалуйста, выберите фигуру (R - Прямоугольник, C - Окружность, T - Треугольник)
T
Пожалуйста, введите x1, y1, x2, y2, x3, y3
3 5 7 2 6 2
Фигура 1
Area = 4
Perimeter = 10
Фигура 2
Area = 78.5398
Perimeter = 31.4159
Фигура 3
Area = 1.5
Perimeter = 10.2426
```