

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Объектно-ориентированное программирование»
ТЕМА: НАСЛЕДОВАНИЕ

Студент гр. 7304

Моторин Е.В.

Преподаватель

Размочаева Н.В

Санкт-Петербург

2019

Цель работы:

Изучить наследование в языке C++. Научиться строить системы классов.

Задача:

Необходимо спроектировать систему классов для моделирования геометрических фигур (в соответствии с полученным индивидуальным заданием). Задание предполагает использование виртуальных функций в иерархии наследования, проектирование и использование абстрактного базового класса. Разработанные классы должны быть наследниками абстрактного класса Shape, содержащего методы для перемещения в указанные координаты, поворота на заданный угол, масштабирования на заданный коэффициент, установки и получения цвета, а также оператор вывода в поток.

Необходимо также обеспечить однозначную идентификацию каждого объекта.

Решение должно содержать:

- условие задания;
- UML диаграмму разработанных классов;
- текстовое обоснование проектных решений;
- реализацию классов на языке C++.

Вариант 10 – прямоугольник, трапеция, равнобедренная трапеция.

Реализация:

- **Color.** Класс разработан для хранения цвета модели. Цвет хранится в формате RGB, то есть 3 базовых цвета: красный, зеленый и голубой. Для каждого цвета насыщенность в диапазоне 0 – 255.
- **Point.** Класс представляет из себя абстракцию точки на плоскости.
- **Shape.** Абстракция модели геометрической фигуры. Характеристики :
 - *color* – цвет фигуры
 - *angle* – угол поворота
 - *centerOffset* – смещение центра фигуры относительно начала системы координат
 - *mv* – изменить смещение центра на заданное расстояние для каждой оси
 - *scale* – масштаб фигуры
 - *id* – уникальный номер каждого экземпляра класса.
- **Rectangle.** Реализация класса прямоугольника. Прямоугольник строится основываясь на высоте и ширине. Масштабирование осуществляется за счет умножения высоты и ширины фигуры на коэффициент.
- **Trapezium.** Реализация класса трапеции. Для построения необходимо знать смещение верхнего оснований относительно центра вертикали, длины оснований и высоту. При масштабировании длины оснований, высота и смещение умножаются на коэффициент.
- **RightTrapezium.** Реализация класса равнобедренной трапеции. Для построения необходимо знать высоту и длины оснований. При масштабировании высота и длины оснований умножаются на коэффициент.

Во всех классах перегружен оператор вывода для удобства отслеживания изменений фигур.

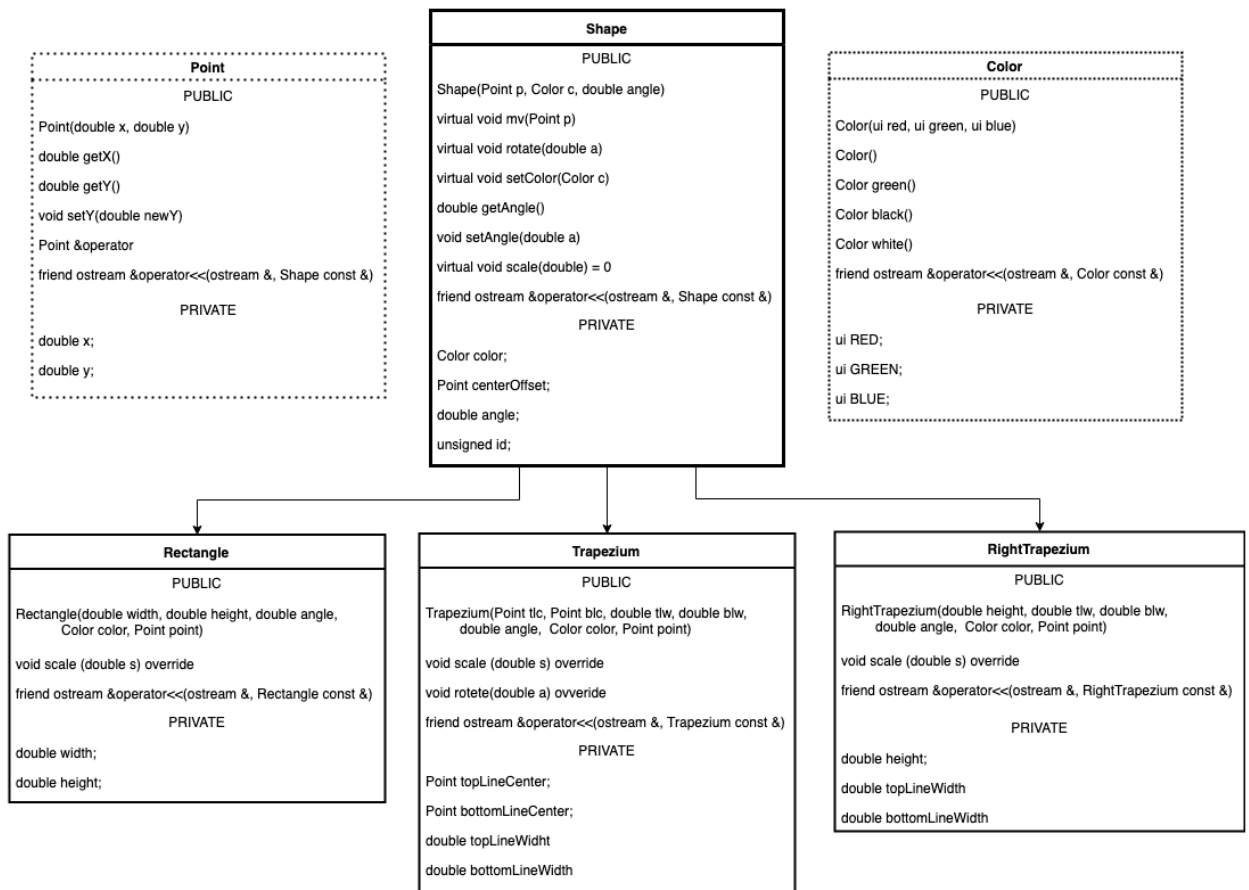


Рисунок 1. (UML диаграмма)

Результат:

- Тело главной функции.

```

int main() {
    Rectangle rect(100.0, 200.0);
    Trapezium trap(30.0, 100.0, 20.0, 40.0);
    RightTrapezium rtrap(10.0, 20.0, 60.0);

    cout << "BEFORE: \n\n";
    cout << rect << endl;
    cout << trap << endl;
    cout << rtrap << endl;

    cout << "\nAFTER: \n\n" ;
    rect.rotate(90.0);
    rect.scale(3.5);
    rect.setColor(Color().green());

    trap.rotate(10.0);
    trap.mv(Point(10.0, -5.0));
    trap.scale(0.33);
    trap.setColor(Color(40.0, 5.0, 0.0));

    rtrap.rotate(-40.0);
    rtrap.mv(Point(12.0, -20.0));
    rtrap.setColor(Color().white());

    cout << rect << endl;
    cout << trap << endl;
    cout << rtrap << endl;

    return 0;
}
  
```

Рисунок 2.

■ Результат выполнения.

```
BEFORE:

IDENTIFIRE: 1|
COLOR: (R = 255, G = 255, B = 255)
CENTER OFFSET: (x = 0,y = 0)
ANGLE: 0
p1:(x = -50,y = -100), p2:(x = -50,y = 100), p3:(x = 50,y = 100), p4:(x = 50,y = -100)
SHAPE: Rectangle
ID 1
WIDTH: 100
HEIGHT: 200

IDENTIFIRE: 2
COLOR: (R = 255, G = 255, B = 255)
CENTER OFFSET: (x = 0,y = 0)
ANGLE: 0
p1:(x = -50,y = -20), p2:(x = 5,y = 20), p3:(x = 35,y = 20), p4:(x = 50,y = -20)
SHAPE: Trapezium
TOP LINE WIDTH: 30
HEIGHT: 40
BOTTOM LINE WIDTH: 100
TOP LINE OFFSET: 20

IDENTIFIRE: 3
COLOR: (R = 255, G = 255, B = 255)
CENTER OFFSET: (x = 0,y = 0)
ANGLE: 0
p1:(x = -30,y = -5), p2:(x = -10,y = 5), p3:(x = 10,y = 5), p4:(x = 30,y = -5)
SHAPE: RightTrapezium
TOP LINE WIDTH: 20
BOTTOM LINE WIDTH: 60
HEIGHT: 10

AFTER:

IDENTIFIRE: 1
COLOR: (R = 0, G = 255, B = 0)
CENTER OFFSET: (x = 0,y = 0)
ANGLE: 90
p1:(x = 350,y = -175), p2:(x = -350,y = -175), p3:(x = -350,y = 175), p4:(x = 350,y = 175)
SHAPE: Rectangle
ID 1
WIDTH: 350
HEIGHT: 700

IDENTIFIRE: 2
COLOR: (R = 40, G = 5, B = 0)
CENTER OFFSET: (x = 10,y = -5)
ANGLE: 10
p1:(x = -5.10325,y = -14.3649), p2:(x = 10.4789,y = 1.78625), p3:(x = 20.2285,y = 3.50537), p4:(x = 27.3954,y = -8.63454)
SHAPE: Trapezium
TOP LINE WIDTH: 9.9
HEIGHT: 13.2
BOTTOM LINE WIDTH: 33
TOP LINE OFFSET: 6.6

IDENTIFIRE: 3
COLOR: (R = 255, G = 255, B = 255)
CENTER OFFSET: (x = 12,y = -20)
ANGLE: -40
p1:(x = -10.5321,y = 18.4877), p2:(x = 11.2167,y = 13.2924), p3:(x = 26.5376,y = 0.43686), p4:(x = 35.4306,y = -20.0795)
SHAPE: RightTrapezium
TOP LINE WIDTH: 20
BOTTOM LINE WIDTH: 60
HEIGHT: 10

Program ended with exit code: 0
```

Рисунок 3.

Из рисунков 2 и 3 видно, что результат выполнения работы удовлетворяет условиям задачи.

Вывод:

Таким образом, в ходе данной лабораторной работы был подробно изучен механизм наследования в языке программирования C++. Была построена система классов.