

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 центры оснований и их длины. При повороте центром вращения является центр нижнего основания, то есть меняется центр верхнего основания и угол поворота оснований, длины оснований остаются неизменными. При масштабировании длины оснований и высота верхнего основания 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.

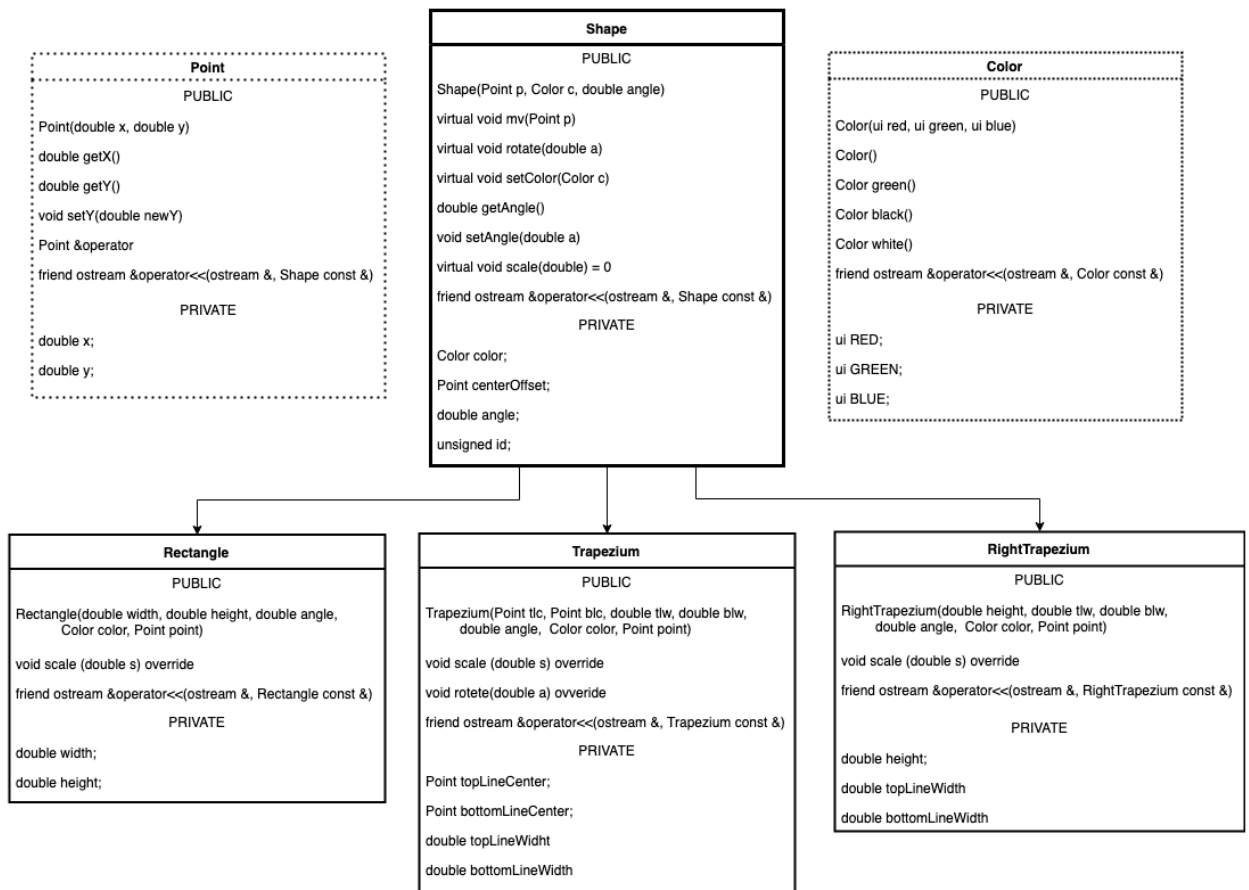


Рисунок 1. (UML диаграмма)

Результат:

- Тело главной функции.

```

Rectangle rect(99.5, 200.0);
Trapezium trap(Point(60.0, 30.0), Point(70.0, 0.0), 30.0, 100.0);
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(100.0);
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:

ID: 1
COLOR: (R = 255, G = 255, B = 255)
CENTER OFFSET: (x = 0,y = 0)
ANGLE: 0
SHAPE: Rectangle
WIDTH: 99.5
HEIGHT: 200

ID: 2
COLOR: (R = 255, G = 255, B = 255)
CENTER OFFSET: (x = 0,y = 0)
ANGLE: 0
SHAPE: Trapezium
TOP LINE WIDTH: 30, CENTER: (x = 60,y = 30)
BOTTOM LINE WIDTH: 100, CENTER: (x = 70,y = 0)

ID: 3
COLOR: (R = 255, G = 255, B = 255)
CENTER OFFSET: (x = 0,y = 0)
ANGLE: 0
SHAPE: RightTrapezium
TOP LINE WIDTH: 20
BOTTOM LINE WIDTH: 60
HEIGHT: 10

AFTER:

ID: 1
COLOR: (R = 0, G = 255, B = 0)
CENTER OFFSET: (x = 0,y = 0)
ANGLE: 100
SHAPE: Rectangle
WIDTH: 348.25
HEIGHT: 700

ID: 2
COLOR: (R = 40, G = 5, B = 0)
CENTER OFFSET: (x = 10,y = -5)
ANGLE: 10
SHAPE: Trapezium
TOP LINE WIDTH: 9.9, CENTER: (x = 94.7113,y = -6.51154)
BOTTOM LINE WIDTH: 33, CENTER: (x = 70,y = 0)

ID: 3
COLOR: (R = 255, G = 255, B = 255)
CENTER OFFSET: (x = 12,y = -20)
ANGLE: -40
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.