

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

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Объектно-ориентированное программирование»**  
**Тема: Наследование**

Студент гр. 7303

\_\_\_\_\_

Петров С.А.

Преподаватель

\_\_\_\_\_

Размочаева Н.В.

Санкт-Петербург

2019

### **Цель работы.**

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.

### **Задание.**

Треугольник, Эллипс, Прямоугольный треугольник.

### **Ход работы.**

Для удобства были созданы вспомогательные классы, содержащие представление точки и цвета – Point и Color.

Был создан абстрактный класс Shape, содержащий поля для представления цвета фигуры, центра фигуры, угла поворота, идентификатор - для однозначного представления фигуры. Были добавлены методы для установки цвета, получения точки с координатами центра, получения идентификатора, угла. Однозначная идентификация обеспечивается статической переменной. Были добавлены чистые виртуальные методы: move() – для перемещения фигуры, scale() – для увеличения линейных размеров фигуры, turn() – для поворота фигуры на определенный угол; для последующего переопределения в классах-наследниках. Конструктор класса принимает цвет и позицию центра, инициализирует начальный угол фигуры нулем. Также был переопределен оператор вывода в поток, выводящий сведения о фигуре.

Был создан класс Triangle, содержащий представление треугольника. Класс унаследован от Shape, добавлены поля points – вектор, содержащий три

точки пересечения сторон треугольника, и поля, содержащий длины сторон треугольника –  $a$ ,  $b$ ,  $c$ . Был добавлен метод, возвращающий вектор сторон треугольника, переопределен оператор вывода в поток. Были переопределены чистые виртуальные метода класса Shape – `move()` перемещает центр треугольника в заданную точку и пересчитывает координаты точке пересечения сторон, `turn()` прибавляет к текущему углу треугольника заданный и пересчитывает координаты точке пересечения сторон, `scale()` умножает координаты точек пересечения сторон на заданный коэффициент и пересчитывает длину сторон. Конструктор класса принимает три точки и цвет. Выполнимость правила треугольника, построенного на этих точках, остается на пользователе. Также был определен конструктор копирования.

Был создан класс `Ellipse`, содержащий представление эллипса. Класс унаследован от `Shape`, были добавлены поля  $a$  и  $b$ , содержащие два радиуса эллипса. Были добавлены метода, возвращающие данные радиусы, и переопределен оператор вывода в поток. Были переопределены чистые виртуальные метода класса Shape – `move()` перемещает центр эллипса в заданную точку, `turn()` прибавляет к текущему углу эллипса заданный, `scale()` умножает радиусы на заданный коэффициент. Конструктор класса принимает центр эллипса, длины обоих радиусов и цвет. Был определен конструктор копирования.

Был создан класс `RightTriangle`, содержащий представление прямоугольного треугольника. Так как это частный случай треугольника, класс унаследован от `Triangle`, и, так как прямоугольный треугольник можно задать с помощью двух катетов, был определен конструктор, принимающий точку пересечения двух катетов, длины двух катетов, пересекающихся в этой точке, и цвет фигуры.

UML-диаграмма классов представлена на рис. 1.

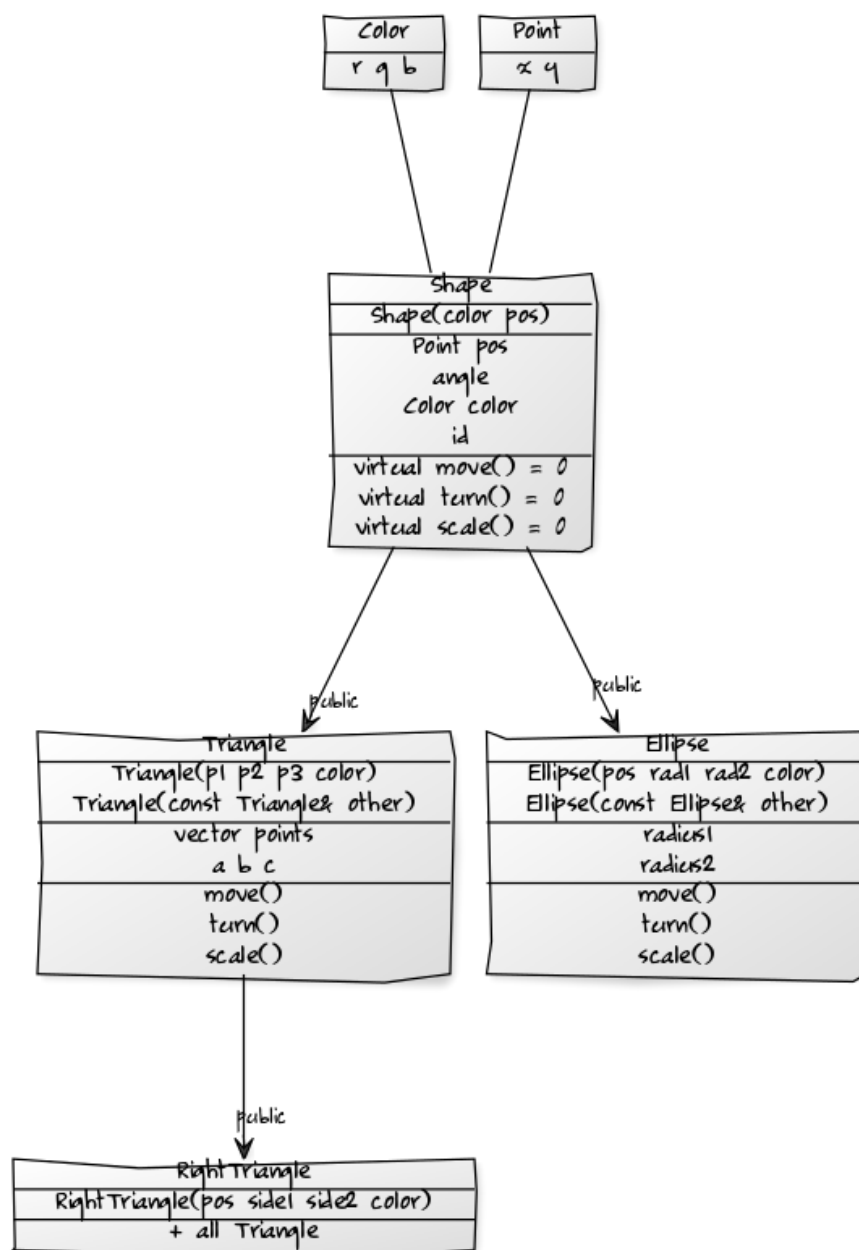


Рисунок 1 – UML-диаграмма классов

## Вывод.

Была спроектирована система классов для моделирования геометрических фигур (треугольника, эллипса, прямоугольного треугольника). Разработанные классы являются 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.