

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

Пэтайчук Н.Г.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2019

Цель работы

Изучение концепции наследования и её реализации в языке программирования C++;

Постановка задачи

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.

Описание решения

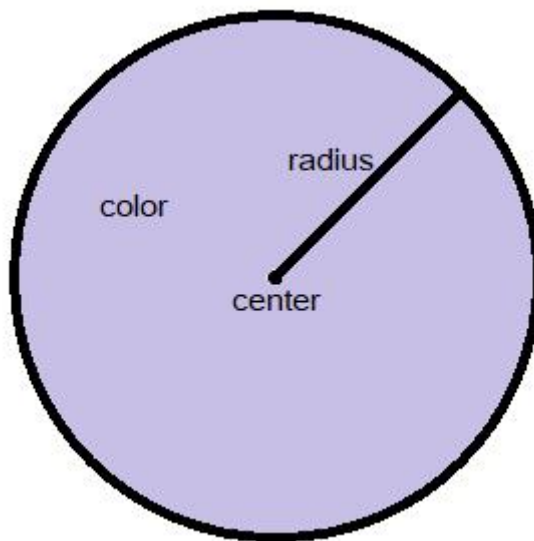
В данной лабораторной работе будет рассмотрена реализация следующих геометрических фигур:

1. Круг;
2. Ромб;
3. Трапеция;

- Первым делом был реализован класс радиус-вектора, который необходим для задания координат центра и вершин фигуры. Класс радиус-вектора включает в себя для поля вещественного типа, ответственные за координаты по оси X и Y, конструктор от двух вещественных значений (по умолчанию считается, что радиус-вектор указывает на начало координат), конструктор копирования, переопределенный оператор присваивания, методы установки координат и константные методы получения значений координат. Также для выполнения математических операций над радиус-векторами были перегружены операторы сложения, вычитания и умножения на вещественное число;
- Далее был создан класс цвета фигуры в RGB кодировке, то есть у данного класса есть три поля беззнакового целого типа, определяющие насыщенность красного, зелёного и синего оттенков в самом цвете, есть методы, устанавливающие значение компонентов цвета, и есть константные методы получения значений каждого компонента. У данного класса есть ещё конструктор от трёх беззнаковых целых чисел (причём в качестве стандартного цвета выступает чёрный), конструктор копирования и

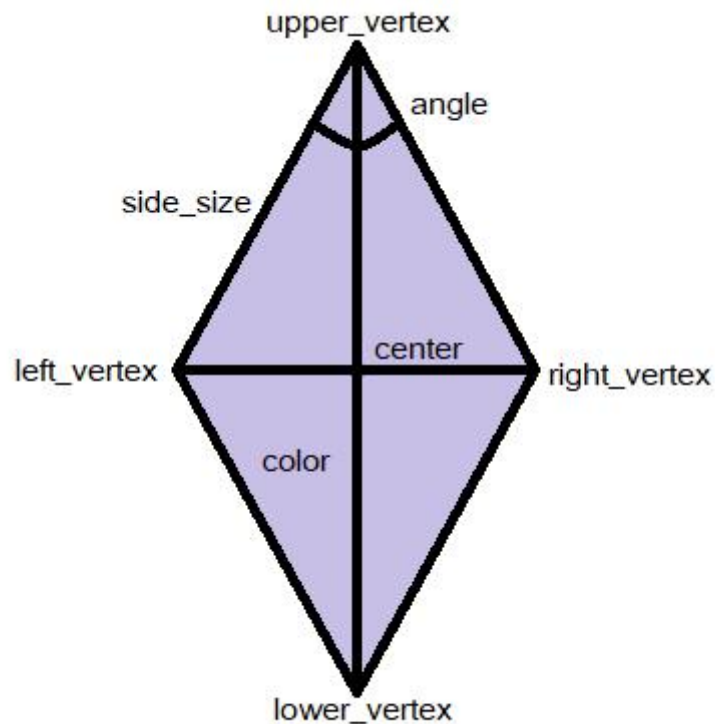
переопределенный оператор присваивания. Данная реализация предоставляет широкий спектр для раскраски фигур;

- Следующим шагом стала реализация абстрактного класса фигуры. Данный класс хранит в себе следующие поля: координаты центра, цвет, идентификатор фигуры и статическое поле, определяющее следующий идентификационный номер. Класс фигуры определяет общий набор операций, которые можно совершать с фигурами, а именно поворот, масштабирование, перемещение, изменение и получение цвета фигуры, при этом первые три функции объявлены в классе как чисто виртуальные, поскольку в общем случае эти операции необходимо определять для каждой фигуры по-своему. Поскольку класс абстрактный, то достаточно для него определить один конструктор от координат центра и цвета (ID определяется автоматически), но в то же время необходимо определить виртуальный конструктор. Были определены ещё константные методы получения координат центра и ID;
- Первым реализованным классом-наследником является класс круга, задаваемый при помощи координат центра, цвета и радиуса круга. Данный класс является самым простым в плане реализации из всех, поскольку для перемещения круга достаточно поменять его центр, для масштабирования - изменить радиус, а функцию поворота можно вообще оставить пустой;

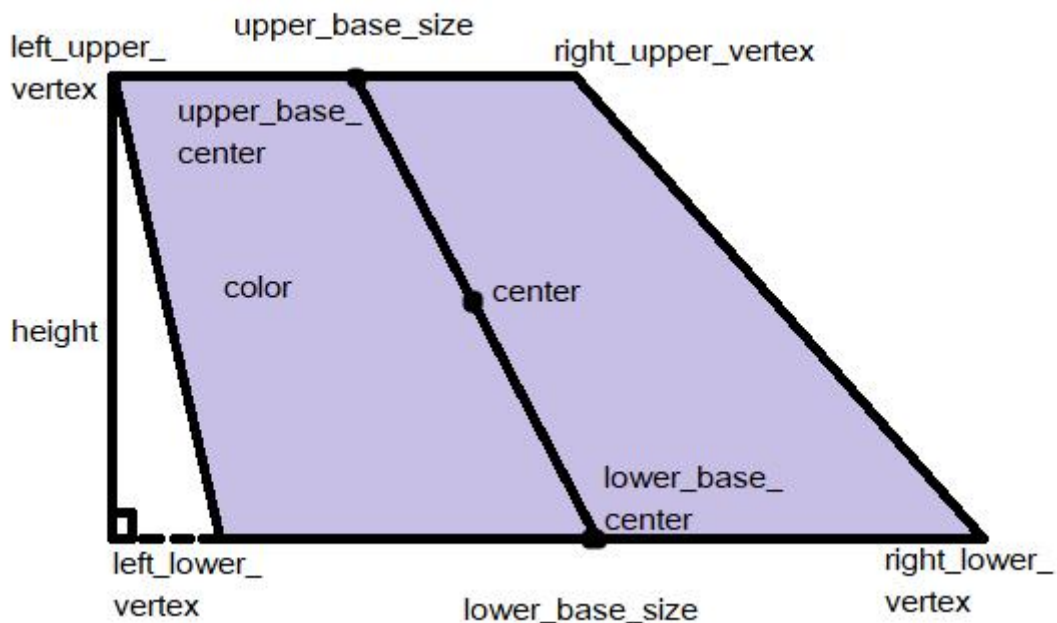


- Следующей реализованной фигурой является, который задаётся, помимо местоположения центра и цвета, длиной стороны и углом, по которым с помощью вычисления длин диагоналей определяются координаты вершин ромба (по умолчанию ромб расположен так, что его диагонали параллельны соответствующим осям координат). Перемещение фигуры осуществляется при помощи соответствующего изменения координат вершин и центра; масштабирование ромба представляет из себя изменение длин, характеризующих ромб (а именно стороны) и изменение длин векторов, соединяющих центр фигуры с вершинами; поворот выполняется с помощью

перемещения фигуры в центр координатной плоскости, умножения координат вершин на матрицу поворота и возврата на прежнее местоположение;

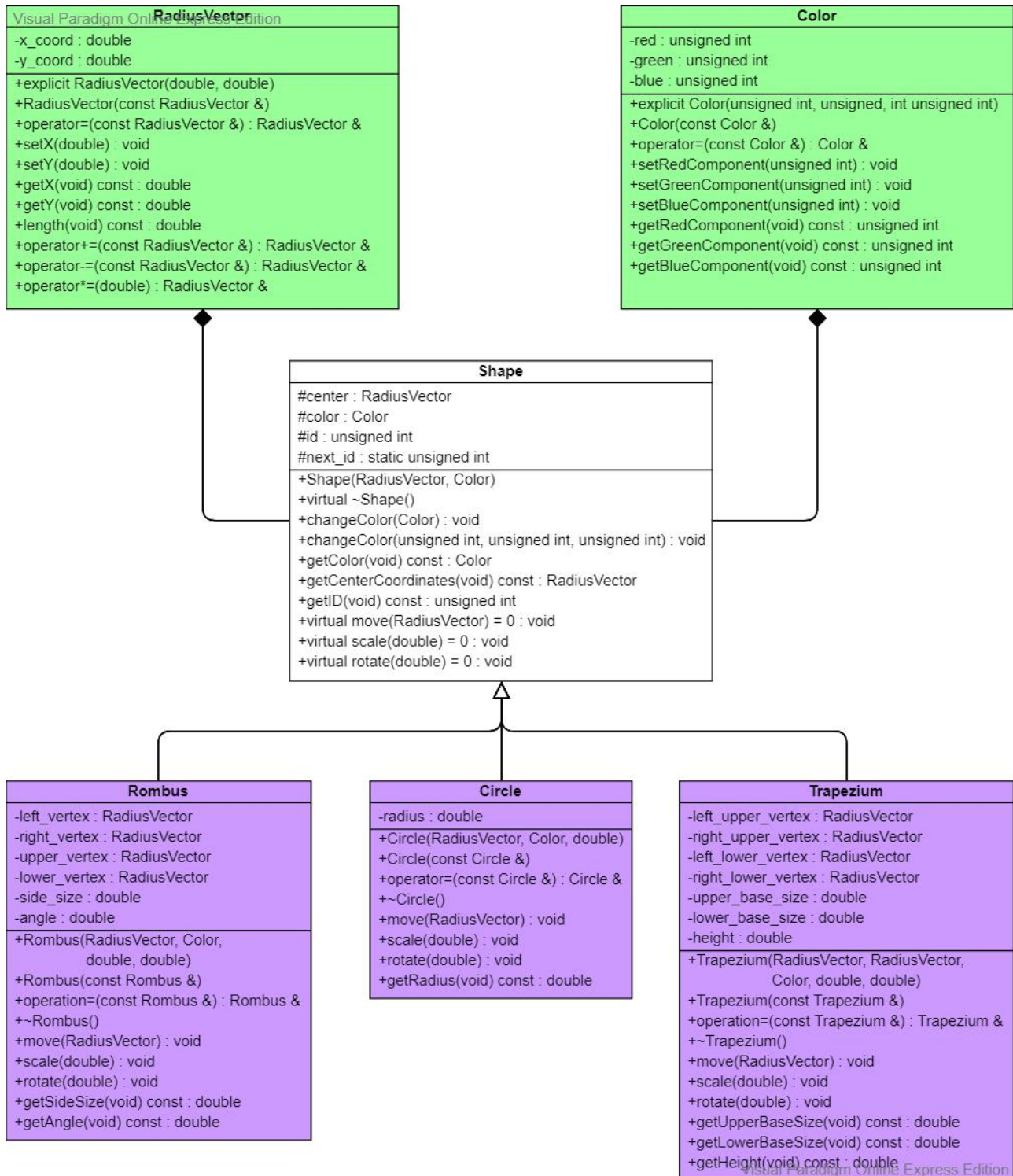


- Последней фигурой, которую требовалось реализовать в рамках лабораторной работы, является трапеция, которая определяется с помощью длин оснований и координат их центров; данный метод позволяет легко определить координаты вершин (по умолчанию считается, что основания параллельны оси абсцисс) и в то же время однозначно определяет местоположение центра трапеции. Принципы работы методов поворота, масштабирования и перемещения такие же, как и у ромба;



- В конце были перегружены операторы вывода в поток, использующие константные методы получения значения величин фигур, а также была написана головная функция, демонстрирующая работу с объектами классов фигур при помощи указателя на базовый класс;

Всю структуру взаимосвязей классов можно увидеть на UML-диаграмме, приведённой ниже:



Выводы

В ходе лабораторной работы была разработана иерархия классов, включающая в себя классы нескольких геометрических фигур, унаследованных от одного абстрактного класса фигуры, определяющего общий интерфейс работы с геометрическими фигурами. Помимо этого, были изучены виртуальные функции и соответственно динамический полиморфизм, а также были дополнительно реализованы классы радиус-вектора и цвета и перегружены операторы присваивания, сложения, вычитания, умножения и вывода в поток.