

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.

Студентка гр.7304

Каляева А.В

Преподаватель

Размочаева Н.В

г. Санкт-Петербург

2019 г.

Цель работы:

Необходимо спроектировать систему классов для моделирования геометрических фигур квадрата, параллелограмма, ромба. 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++.

Ход работы:

Для выполнения поставленной задачи были реализованы следующие классы:

1. Класс Point содержит два поля, которые описывают координаты x и y точки. Так же класс Point содержит методы для получения и установления координат.
2. Класс Color содержит три поля, в которых хранятся числа от 0 до 255 и характеризуют цвет фигуры. Так же класс Color содержит методы для получения информации о цвете.
3. Абстрактный класс Shape содержит поля цвета, номера фигуры id, координаты центра фигуры, вектор, хранящий координаты вершин фигур. Класс Shape содержит следующие методы:
 - 3.1. void set_color(Color color) для установления заданного цвета фигуры.
 - 3.2. Color get_color() const для получения информации об установленном цвете фигуры.
 - 3.3. unsigned long int get_id() const для получения информации об id фигуры.
 - 3.4. void moving(Point p) для смещения фигуры в заданную точку.
 - 3.5. void rotation(double grade) для поворота фигуры на заданный угол.
 - 3.6. virtual void scaling(double coefficient)=0 – чисто виртуальный метод для масштабирования фигуры на заданный коэффициент.
 - 3.7. virtual ostream& print_shape(ostream& stream, Shape& shape) = 0 – чисто виртуальный метод для вывода информации о фигуре на экран.
 - 3.8. friend ostream& operator << (ostream& stream, Shape& shape) для переопределения оператора вывода на экран.
4. Класс Square, который наследуется от абстрактного класса Shape. Класс имеет поле, которое характеризует длину стороны квадрата. В конструкторе данного класса вычисляются все вершины квадрата, имея информацию о координатах центра квадрата и длине его стороны. В

классе был переопределен метод `scaling`, который масштабирует квадрат на заданный коэффициент. А так же метод `print_shape`, который выводит информацию о фигуре.

5. Класс `Parallelogram`, который наследуется от абстрактного класса `Shape`. Класс `Parallelogram` имеет три дополнительных поля, которые характеризуют вершины параллелограмма. В конструкторе данного класса вычисляется последняя вершина фигуры, а так же координаты ее центра. В классе был переопределен метод `scaling`, который масштабирует параллелограмм на заданный коэффициент. А так же метод `print_shape`, который выводит информацию о фигуре.
6. Класс `Rhombus` который наследуется от абстрактного класса `Shape`. Класс `Rhombus` имеет два дополнительных поля, содержащих информацию о длинах диагоналей ромба. В конструкторе данного класса вычисляются координаты вершин ромба, имея информацию о координатах одной из вершин и длинах диагоналей. В классе был переопределен метод `scaling`, который масштабирует ромб на заданный коэффициент. А так же метод `print_shape`, который выводит информацию о фигуре.

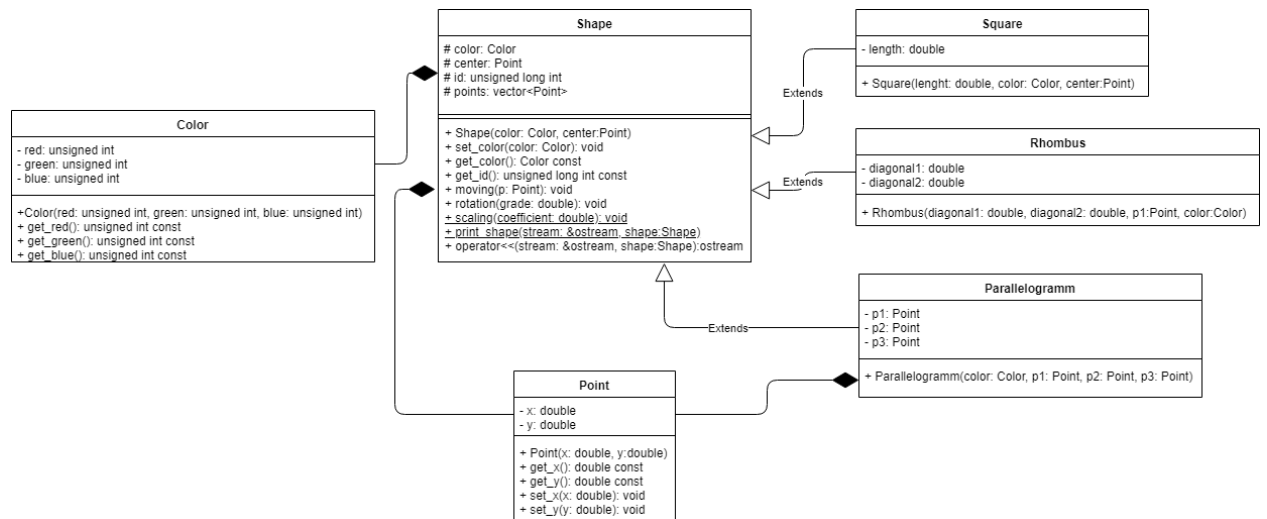
Обоснование решения:

В данной лабораторной работе был реализован абстрактный класс `Shape`. Поля `цвет`, `координаты центра фигуры` и `координаты вершин фигуры` являются общими, поэтому они содержатся в абстрактном классе `Shape`. Для реализации квадрата необходима информация о вершинах квадрата. Для получения данной информации достаточно знать центр квадрата и длину его стороны. Для реализации параллелограмма необходима информация о вершинах параллелограмма. Для получения данной информации достаточно знать информацию о трех точках. Для реализации ромба необходима информация о вершинах ромба. Для получения данной информации достаточно знать одну из точек ромба и длины двух диагоналей ромба.

`Moving` - это перемещение в заданную точку. Для любой фигуры можно найти расстояние между новой точкой и текущим центром фигуры. Перемещение фигуры – это смещение каждой из вершин этой фигура на полученное расстояние.

`Rotation` – это поворот на заданный угол. Для каждой фигуры поворот на заданный угол можно получить, умножив все координаты фигуры на матрицу поворота.

UML диаграмма классов:



Заключение :

В ходе выполнения данной лабораторной работы была изучена тема наследование. Была спроектирована система классов для моделирования геометрических фигур квадрат, параллелограмм, ромб. Были использованы виртуальные функции в иерархии наследования. Были разработаны классы, которые являются наследниками абстрактного класса Shape.