

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

Павлова В.Р.

Преподаватель

Размочаева Н.В.

г. 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. Решение должно содержать:

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

Вариант 14: круг, трапеция, ромб.

Ход работы:

Для выполнения лабораторной работы были реализованы следующие классы:

- 1) Класс Point, содержащий координаты точки x и y.
- 2) Класс Colour, содержащий информацию цвета поля (r, g b).
- 3) Класс Shape, который содержит информацию точки центра фигуры (center), номер фигуры (id), цвет фигуры (color) и угол, на которую повернута фигура (angle).

Методы класса:

- `size_t getId()` - получает информацию об id фигуры;
- `double getAngle()` – получает информацию об угле, на которую повернута фигура;

- `double getCenterX(), double getCenterY()` – получает информацию об координатах центра фигуры;
- `void newColour(Colour color)` – устанавливает новый цвет фигуры;
- `Colour getColour()` – получает информацию о цвете фигуры;
- `virtual void set_move(Point cntr) = 0` – виртуальный метод, который перемещает центр фигуры на указанную точку;
- `virtual void rotating(double angl) = 0` – виртуальный метод, который поворачивает фигуру на указанный угол
- `virtual void scaling(double k) = 0` – виртуальный метод, который увеличивает масштаб фигуры на коэффициент `k`;
- `virtual ostream& out_shape(ostream& stream, Shape& shape)` – виртуальный метод, который выводит информацию фигуры
- `friend ostream& operator << (ostream& stream, Shape& shape)` – переопределяет оператор вывода на экран

4) Класс `Circle`, который наследуется от абстрактного класса `Shape`. Содержит информацию о радиусе круга и одной точки, которая лежит на окружности круга. Были переопределены методы `set_move`, `rotating`, `scaling`, а также метод вывода данных.

5) Класс `Rhombus`, который наследуется от абстрактного класса `Shape`. Содержит информацию о диагоналях ромба, а также координаты вершины ромба. Были переопределены методы `set_move`, `rotating`, `scaling`, а также метод вывода данных.

6) Класс `Trapezium`, который наследуется от абстрактного класса `Shape`. Содержит информацию о вершинах трапеции. Были переопределены методы `set_move`, `rotating`, `scaling`, а также метод вывода данных.

Обоснование решения:

Был реализован абстрактный класс Shape, от которого наследуются классы Circle, Rhombus, Trapezium. Класс Shape содержит информацию о цвете фигуры, координаты центра фигуры, id фигуры и угол, на которую повернута фигура, т.к. эта информация является общей для всех фигур (круга, ромба, трапеции).

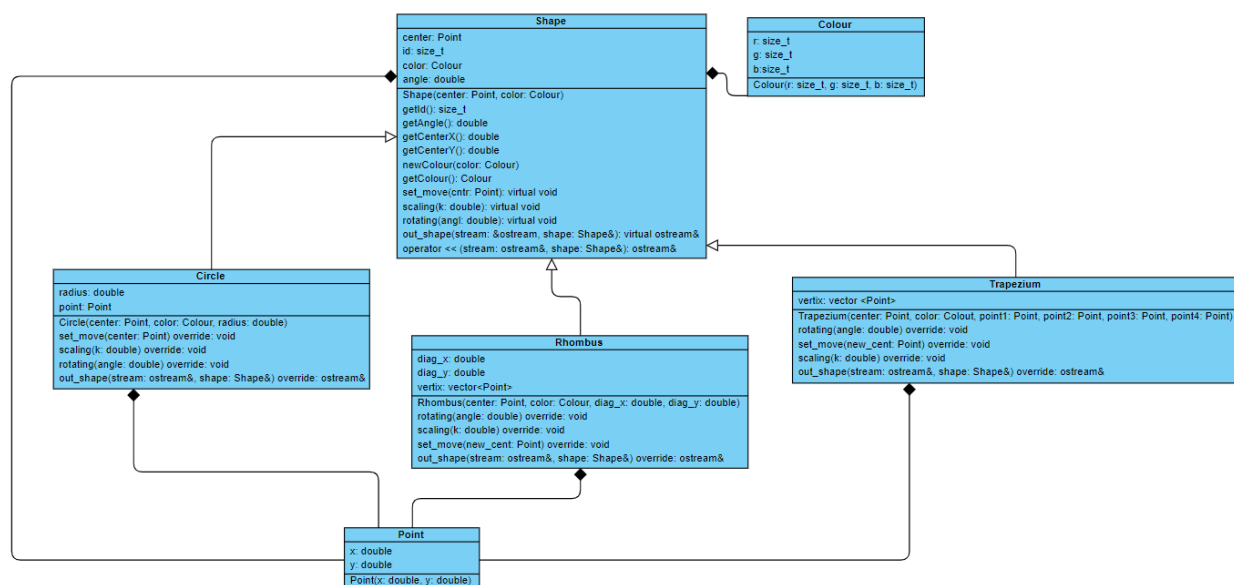
Чтобы реализовать круг, необходимо знать его радиус. Точка окружности круга задана для наглядности того, что круг был повернут.

Чтобы реализовать ромб, необходимо знать его диагонали. Для удобства ромб содержит информацию о вершинах.

Чтобы реализовать трапецию, необходимо знать вершины трапеции. Если же вершины были введены неверно, т.е. основания трапеции не параллельны друг другу, то фигура не создается.

set_move перемещает фигуру в заданную точку, т.е. он смещает вершины фигуры (ромб, трапеция) или же точку окружности (круг) на разность нового и старого центров. Меняется центр фигуры. scaling меняет масштаб фигуры на заданный коэффициент. rotating поворачивает фигуру на заданный угол. Для этого все вершины фигуры (ромб, трапеция) или точка окружности (круг) были умножены на матрицу поворота.

UML-диаграмма разработанных классов:



Выводы:

В ходе выполнения данной лабораторной работы было изучено наследование, проектирование схем классов, применение виртуальных функций.