

**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»**  
**Вариант №4**

Студент гр. 7304

\_\_\_\_\_

Дементьев М.Е.

Преподаватель

\_\_\_\_\_

Размочаева Н.В.

Санкт-Петербург  
2019

## Цель работы

Изучить наследование в C++ при помощи проектирования системы классов для моделирования геометрических фигур (круг, пятиконечная звезда, шестиконечная звезда).

## 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++.

## Ход работы

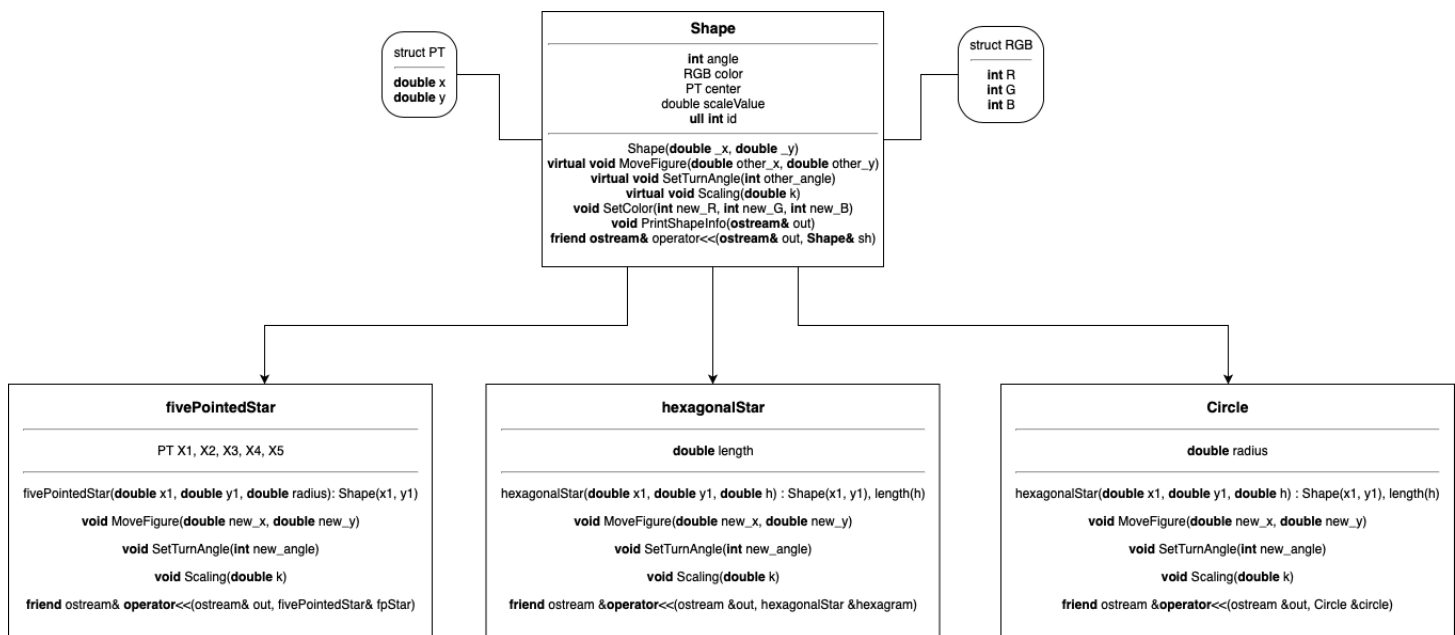
1) Для хранения цвета и координат были созданы структуры PT и RGB, соответственно.

2) Для последующего создания необходимых геометрических фигур был создан класс Shape, который объединяет все общее для фигур, а именно: координаты центра фигуры, масштаб, угол поворота, цвет и id.

Так же данных классов были написаны общие методы: задание цвета, печать информации о фигуре, а так же объявлены методы перемещения, масштабирования и поворота фигуры.

Для удобства вывода необходимой информации в консоль без дублирования кода был перегружен оператор "<<" (вывода на консоль)

- 3) Класс `fivePointerStar` создает пятиконечную звезду, которая определяется в пространстве координатами центра и длиной радиуса, с помощью которых определяются пять точек – вершин лучей звезды. Также для данного класса написаны функции, объявленные в `Shape`. Еще был перегружен оператор “<<” для вывода полной информации о фигуре в консоль.
- 4) Класс `hexagonalStar` создает шестиконечную звезду, которая определяется координатами центра и расстоянием между противоположными вершинами. Остальные возможности класса аналогичны `fivePointerStar`;
- 5) Класс `Circle` создает круг, который определяется координатами центра и радиусом. Остальные возможности класса аналогичны `fivePointerStar`, за исключением функции поворота фигуры, которая отсутствует у круга.
- 6) На основе вышеуказанной иерархии была построена UML-диаграмма:



## Результаты работы программы

```
int main() {  
  
    fivePointedStar pent(20, 30, 11);  
    cout << pent;  
    pent.MoveFigure(0, 0);  
    pent.SetTurnAngle(30);  
    cout << pent;  
  
    hexagonalStar hex(20, 20, 10);  
    cout << hex;  
    hex.Scaling(2);  
    cout << hex;  
  
    Circle circl(0, 10, 5);  
    cout << circl;  
    circl.SetColor(256, 256, 256);  
    cout << circl;  
  
    return 0;  
}
```

```
five-pointed star  
Shape ID: 0  
Figure centre: (20, 30)  
Angle of rotation = 0  
Color (RGB): (0, 0, 0)  
Scale factor = 1
```

POINTS COORDINATES:

```
POINT X1: (31, 30)  
POINT X2: (23, 40)  
POINT X3: (11, 36)  
POINT X4: (11, 24)  
POINT X5: (23, 20)
```

---

```
five-pointed star  
Shape ID: 0  
Figure centre: (0, 0)  
Angle of rotation = 30  
Color (RGB): (0, 0, 0)  
Scale factor = 1
```

POINTS COORDINATES:

```
POINT X1: (9.5, 0)  
POINT X2: (-2.3, 14)  
POINT X3: (-11, 8.8)  
POINT X4: (-4.5, -8.8)  
POINT X5: (8.2, -14)
```

---

```
hexagonalStar
Shape ID: 1
Figure centre: (20, 20)
Angle of rotation = 0
Color (RGB): (0, 0, 0)
Scale factor = 1
```

```
INFO ABOUT SIZE:
Distance between opposite vertices: 10
-----
```

```
hexagonalStar
Shape ID: 1
Figure centre: (20, 20)
Angle of rotation = 0
Color (RGB): (0, 0, 0)
Scale factor = 1
```

```
INFO ABOUT SIZE:
Distance between opposite vertices: 20
-----
```

```
Circle
Shape ID: 2
Figure centre: (0, 10)
Angle of rotation = 0
Color (RGB): (0, 0, 0)
Scale factor = 1
```

```
INFO ABOUT SIZE:
Circle radius: 5
-----
```

```
Circle
Shape ID: 2
Figure centre: (0, 10)
Angle of rotation = 0
Color (RGB): (256, 256, 256)
Scale factor = 1
```

```
INFO ABOUT SIZE:
Circle radius: 5
-----
```

## **Вывод**

В ходе выполнения лабораторной было изучено наследование в C++, спроектирована система классов для моделирования геометрических фигур (круг, пятиконечная звезда, шестиконечная звезды).