

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. 7381

Кревчик А.Б.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2019

Цель работы.

Изучение наследования классов. Разработка абстрактного класса. Изучение виртуальных функций.

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.

Решение должно содержать:

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

Индивидуализация: круг, эллипс, сектор эллипса.

Ход работы.

Рассмотрим базовый класс Shape:

```
class Shape
{
protected:
    string Colour;//цвет
    int ID;//идентификационный номер
    double Angle;//угол поворота в градусах
public:
    Shape(): ID(id++), Colour("White"), Angle(0.0){};

    ~Shape(){};

    virtual void Move(double, double)=0;//перемещение в указанные
координаты
```

```

void Rotation(double angle){//вращение на заданный угол
    if (angle >= 360.0) {
        while (angle >= 360.0)
            angle -= 360.0;
    }
    if (angle < 0.0) {
        while (angle < 0.0)
            angle += 360.0;
    }
    Angle = Angle + angle;
    while(Angle >= 360)
        Angle = Angle - 360;
    cout << "New angle is " << Angle <<endl;
}

virtual void Scaling(double)=0;//масштабирование

void setColour(string colour){//установка цвета
    Colour = colour;
    cout << "New colour is " << Colour <<endl;
}

string getColour() const{//получение цвета
    return Colour;
}

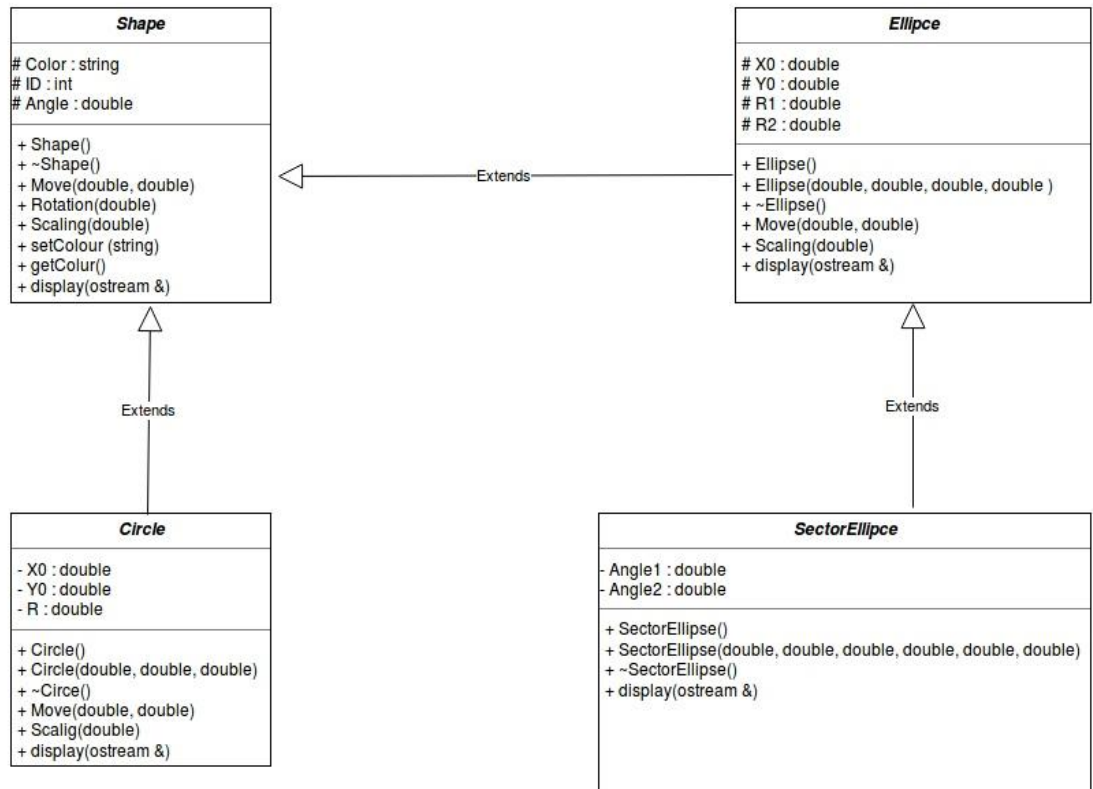
friend ostream & operator << (ostream&out, Shape &figure)
{ //оператор вывода в поток вывода
    figure.Shape::display(out);
    figure.display(out);
    return out;
}

virtual void display(ostream &out){
    out << "-----" <<
endl;
    out << "Base class output:" << endl;
    out << "ID = " << ID << endl;
    out << "Colour: " << Colour << endl;
    out << "Angle of rotation: " << Angle << endl;
    out << "-----"
-" << endl;
}
};

```

Так как все объекты классов-наследников будут иметь цвет, угол поворота и идентификационный номер, мы объявляем их в базовом классе. Так же описаны методы установки и получения цвета, поворота, а также перегружен оператор вывода. Присутствуют и чисто-виртуальные функции, реализация которых для каждого класса своя (методы перемещения и масштабирования).

Наследование.



Выводы.

В ходе выполнения данной лабораторной работы была спроектирована система классов для работы с геометрическими фигурами. В работе использовались виртуальные функции и виртуальный базовый класс. Были реализованы методы для работы с геометрическими фигурами.