

Лабораторная работа №8

Наследование и абстрактные классы

Цель: Изучить использование наследования классов

Краткие теоретические сведения

В программировании полиморфизм означает то, что один и тот же метод может быть определен для объектов различных типов. Конкретное поведение метода будет зависеть от типа объекта. C++ поддерживает полиморфизм с помощью динамического связывания и виртуальных функций-членов.

Описание класса *Shape*

По умолчанию конструктор задает базовую точку (0,0) в левом верхнем углу окна. Нулевой образец заполнения обычно предполагает его отсутствие. Методы *GetX* и *GetY* возвращают координаты *x* и *y* базовой точки. Метод *SetPoint* позволяет клиенту изменять базовую точку. Похожие методы *GetFill* и *SetFill* обеспечивают доступ к образцу заполнения. Метод *Draw* инициализирует графическую систему таким образом, что фигуры заполняются по образцу *fillpat*. Методы *Area* и *Perimeter* являются чисто виртуальными функциями.

```
class Shape
{
    public:
        int x; int y;
        TBrushStyle Fill;
        Shape(int xx = 0, int yy = 0);
        int GetX();
        int GetY();
        TBrushStyle GetFill();
        void SetPoint(int xx, int yy);
        void SetFill(TBrushStyle fillpat);
        virtual void Draw(TBrushStyle fillpat, TCanvas* Canvas);
        virtual float Area(void);
        virtual float Perimeter(void);
};
```

Задание

Создайте базовый класс *Shape*, определяющий точку, образец заполнения и методы доступа к этим параметрам. Этот класс наследуется классами геометрических фигур, которые выдают рисунок формы, а также вычисляют ее площадь и периметр. Затем создайте производные классы *Circle* и *Rectangle*. Радиус передается конструктору в момент создания объекта. В классе *Rectangle* базовой точкой является верхний левый угол объекта. В производных классах метод *Draw* базового класса подменяется своим собственным виртуальным методом *Draw*. Также определяются виртуальные методы *Area* и *Perimeter*.

Создайте программу, иллюстрирующую динамическое связывание и полиморфизм для базового класса *Shape* и производных классов.

Создайте множество объектов классов *Circle* и *Rectangle*, используя контейнерные классы. Реализуйте сортировку объектов по площади и по координатам, используя функциональные классы. Расширьте описание класса *Shape* виртуальными методами перемещения объектов.

Работу выполните с использованием GUI (для прорисовки, заливки, перемещения и сортировки объектов).