

Методика выполнения работы

Выполнение задания работы предполагает решение следующих задач.

- # Написать программу, демонстрирующую работу с объектами двух типов O1 и O2. Создать иерархию соответствующих классов: базовый и два наследника.
- # Клиент (функция main) должен работать со списком объектов. Реализовать работу со списком указателей на объекты базового класса.

Пусть имеется система классов плоских фигур. Составить программу с базовым объектом Shape и объектами-потомками Disk и Square. Класс Shape должен содержать поле Name и методы доступа к нему, а также виртуальную функцию вычисления площади фигуры. Потомки должны содержать виртуальные функции вычисления площади соответствующей фигуры. Базовый класс содержит функцию вычисления объема призмы над фигурой по заданной высоте. В программе необходимо реализовать вычисление объема призмы объектов различных плоских фигур и выдачу результата.

Для каждого типа фигуры имеется свой метод вычисления площади. Однако способ вычисления объемов призм, в основаниях которых лежат фигуры, аналогичен. Если применять обычные методы, то для вычисления объемов призм разных фигур надо создавать разные функции. Применение виртуальных функций позволяет избежать этого и, тем самым, обеспечить повторное использование программного кода. Этим не ограничивается польза виртуальных функций. Можно создать массив указателей на абстрактные фигуры. Далее элементам массива присваиваем указатели на конкретные квадраты, круги и т.п.

По своей сути «фигура вообще» является абстрактным понятием. Ее площадь нельзя вычислить. Поэтому целесообразно сделать класс «фигура» абстрактным, то есть содержащим чисто виртуальную функцию «площадь», которая в данном классе ничего не делает, но работает в производных классах (рис.1). Для реализации подобного замысла нужно в описании класса вставить строку: **virtual double area() =0;**

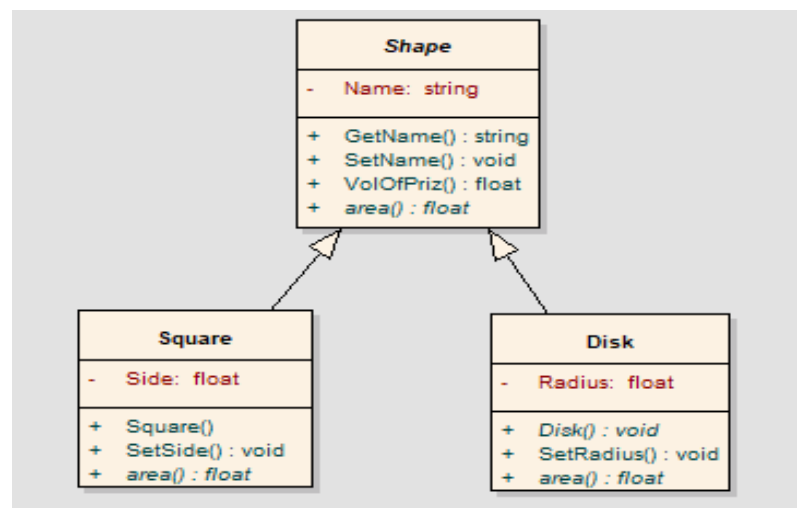


Рис.1 Иерархия классов плоских фигур

Нельзя создать объект абстрактного базового класса. Однако в языке допустимо присвоение переменной типа «указатель на родительский объект» указателя на объект-потомок. Это существенно образом повышает мощь языка. Например, мы можем создать массив указателей на абстрактные геометрические фигуры. Далее элементам массива присваиваем указатели на конкретные квадраты, круги и т.п. После этого можно производить массовые циклические операции типа вычисления площадей, вывод и т.п.

Решение задачи представлено на Листинге 4.

```
#include <iostream>
#include <string>

using namespace std;
//-----абстрактный класс - базовый-----

class Shape {
private:
    string Name;
public:
    void SetName(string s);
    string GetName();
    virtual float area() = 0;           // чисто виртуальный метод
    float VolOfPriz(float H);
};

class Square : public Shape {
private:
    float Side;
public:
    Square();
    void SetSide(float s);
    float area();
};

class Disk : public Shape {
private:
    float Radius;
public:
    Disk();
    void SetRad(float r);
    float area();
};

//----- методы класса Shape-----

void Shape::SetName(string s)
{
    Name = s;
}

string Shape::GetName()
{
    return Name;
}

float Shape::VolOfPriz(float H)
{
    return H * area();    // вызов виртуального метода
}
```

Задание

Разработать программу, реализующую работу с объектами двух типов O1 и O2. Создать иерархию классов: базовый и два наследника.

Каждый объект разработанной иерархии должен иметь имя (произвольная строка символов) и одно или несколько полей для хранения атрибутов объекта, набор обязательных для всех объектов иерархии операций (полиморфные методы) и набор заданных операций (методы обработки).

При необходимости в классы иерархии добавляются дополнительные методы (например, конструктор копирования, операция присваивания и т. п.) для обеспечения надлежащего функционирования этих классов.

Клиент (функция main) должен работать со **списком объектов** (массив, список, последовательные классы STL), где предусмотрены следующие операции: добавить объект в список, удалить объект из списка, показать идентификатор, атрибуты, характеристики объекта и дополнительные операции (зависят от варианта). Реализовать работу со списком указателей на объекты базового класса.

Предусмотреть форму интерфейса, позволяющую продемонстрировать все методы разработанных типов.

Все поля, методы и т.д. должны быть комментированы с использованием тэга summary (подробно см. <https://docs.microsoft.com/ru-ru/cpp/build/reference/summary-visual-cpp?view=vs-2017>).

Варианты заданий представлены в Таблицах 1-3.

Варианты заданий

Таблица 1

Описание базовых классов

| Базовый класс | Полиморфные методы | Статические методы |
|---------------------|---|--|
| Фигура на плоскости | Перемещение (изменение координат на величину) =0 Масштабировать объект на коэффициент =0 Показать характеристики фигуры =0 Показать атрибуты=0 | Вычисление суммы площадей массива фигур Сравнение двух фигур по заданной характеристике |

Таблица 2

Спецификация типов объектов и их операций

| Типы объектов | Атрибуты | Характеристики фигуры |
|--------------------|-------------------------------------|--|
| Точка на плоскости | Координаты (x,y) | Расстояние между двумя точками на плоскости |
| Треугольник | Три точки, заданные координатами | Площадь, периметр, вырожденный/невырожденный, вычисление всех углов по теореме косинуса, высота треугольника |
| Параллелограмм | Две диагональные точки, длина одной | Площадь, периметр, длина диагонали, длина, ширина, высота, размеры углов |

| | | |
|--|---|---|
| | стороны | |
| Трапеция | Две точки Длины оснований | Площадь, периметр, высота, диагонали, длина средней линии, длины боковых сторон |
| Окружность | Координаты точки центра, Длина радиуса | Площадь, длина, диаметр, площадь сектора заданного центрального угла |
| Пентагон (правильный пятиугольник) | Координаты точки центра, Радиус описанной окружности | Площадь, длина стороны, периметр, высота, радиус вписанной окружности |

Таблица 3

Спецификация вариантов

| Вариант | O1 | O2 | Характеристика для сравнения |
|---------|----------------|----------------|---------------------------------|
| 1. | Треугольник | Параллелограмм | Площадь |
| 2. | Параллелограмм | Трапеция | Периметр |
| 3. | Трапеция | Треугольник | Высота |
| 4. | Окружность | Треугольник | Периметр, длина окружности |
| 5. | Пентагон | Параллелограмм | Периметр |
| 6. | Треугольник | Пентагон | Периметр |
| 7. | Параллелограмм | Окружность | Площадь |
| 8. | Трапеция | Пентагон | Периметр |
| 9. | Окружность | Трапеция | Длина диагонали, диаметр |
| 10. | Пентагон | Окружность | Длина диаметра |