

# Лабораторная работа №1

Целью лабораторной работы является:

- Программирование классов на языке C++
- Управление памятью в языке C++
- Изучение базовых понятий ООП.
- Знакомство с классами в C++.
- Знакомство с перегрузкой операторов.
- Знакомство с дружественными функциями.
- Знакомство с операциями ввода - вывода из стандартных библиотек.

**Задача:** Необходимо спроектировать и запрограммировать на языке C++ классы фигур, согласно вариантов задания.

Классы должны удовлетворять следующим правилам:

- Должны иметь общий родительский класс Figure.
- Должны иметь общий виртуальный метод Print, печатающий параметры фигуры и ее тип в стандартный поток вывода cout.
- Должны иметь общий виртуальный метод расчета площади фигуры – Square.
- Должны иметь конструктор, считывающий значения основных параметров фигуры из стандартного потока cin.
- Должны быть расположены в отдельных файлах: отдельно заголовки (.h), отдельно описание методов (.cpp)

**Фигура:** треугольник, квадрат, октагон.

# 1 Описание

Абстракция данных — Абстрагирование означает выделение значимой информации и исключение из рассмотрения незначимой. В ООП рассматривают лишь абстракцию данных (нередко называя её просто «абстракцией»), подразумевая набор значимых характеристик объекта, доступный остальной программе.

Инкапсуляция — свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе. Одни языки (например, C++, Java или Ruby) отождествляют инкапсуляцию с сокрытием, но другие (Smalltalk, Eiffel, OCaml) различают эти понятия.

Наследование — свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствуемой функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником, дочерним или производным классом.

Полиморфизм подтипов (в ООП называемый просто «полиморфизмом») — свойство системы, позволяющее использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта. Другой вид полиморфизма — параметрический — в ООП называют обобщённым программированием.

Класс — универсальный, комплексный тип данных, состоящий из тематически единого набора «полей» (переменных более элементарных типов) и «методов» (функций для работы с этими полями), то есть он является моделью информационной сущности с внутренним и внешним интерфейсами для оперирования своим содержимым (значениями полей). В частности, в классах широко используются специальные блоки из одного или чаще двух спаренных методов, отвечающих за элементарные операции с определенным полем (интерфейс присваивания и считывания значения), которые имитируют непосредственный доступ к полю. Эти блоки называются «свойствами» и почти совпадают по конкретному имени со своим полем (например, имя поля может начинаться со строчной, а имя свойства — с заглавной буквы). Другим проявлением интерфейсной природы класса является то, что при копировании соответствующей переменной через присваивание, копируется только интерфейс, но не сами данные, то есть класс — ссылочный тип данных. Переменная-объект, относящаяся к заданному классом типу, называется экземпляром этого класса. При этом в некоторых исполняющих системах класс также может представляться некоторым объектом при выполнении программы посредством динамической идентификации типа данных. Обычно классы разрабатывают таким образом, чтобы обеспечить отвечающие природе объ-

екта и решаемой задаче целостность данных объекта, а также удобный и простой интерфейс. В свою очередь, целостность предметной области объектов и их интерфейсов, а также удобство их проектирования, обеспечивается наследованием.

Объект — сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса (например, после запуска результатов компиляции и связывания исходного кода на выполнение).

## 2 Исходный код

hexagon.cpp	
Foursquare(std::istream &is);	Ввод из потока std::istream
Foursquare(int32t i);	Конструктор класса
double Square() override;	Получение площади
void Print() override;	Печать фигуры
virtual ~Foursquare();	Деструктор класса
triangle.cpp	
Triangle(std::istream &is);	Ввод из потока std::istream
Triangle(sizet i,sizet j,sizet k);	Конструктор класса
double Square() override;	Получение площади
void Print() override;	Печать фигуры
virtual ~Triangle();	Деструктор класса
octagon.cpp	
Octagon(std::istream &is);	Ввод из потока std::istream
Octagon(int32t i);	Конструктор класса
double Square() override;	Получение площади
void Print() override;	Печать фигуры
virtual ~Octagon();	Деструктор класса

```

1 |
2 | class Octagon : public Figure{
3 | public:
4 |     Octagon();
5 |     Octagon(std::istream &is);
6 |     Octagon(int32_t i);
7 |     Octagon(const Octagon& orig);
8 |     double Square() override;
9 |     void Print() override;
10 |     virtual ~Octagon();
11 | private:

```

```

12     int32_t side;
13 };
14
15 class Triangle : public Figure{
16 public:
17     Triangle();
18     Triangle(std::istream &is);
19     Triangle(size_t i,size_t j,size_t k);
20     Triangle(const Triangle& orig);
21     double Square() override;
22     void Print() override;
23     virtual ~Triangle();
24 private:
25     size_t side_a;
26     size_t side_b;
27     size_t side_c;
28 };
29
30 class Foursquare : public Figure{
31 public:
32     Foursquare();
33     Foursquare(std::istream &is);
34     Foursquare(int32_t i);
35     Foursquare(const Foursquare& orig);
36     double Square() override;
37     void Print() override;
38     virtual ~Foursquare();
39 private:
40     int32_t side;
41 };

```

### 3 КОНСОЛЬ

Menu:

- 1) Initialize A Triangle
- 2) Initialize A Foursquare
- 3) Initialize An Octagon
- 4) Print A Triangle's sides
- 5) Print A Foursquare's sides
- 6) Print An Octagon's sides
- 7) Print A Triangle's Square
- 8) Print A Foursquare's Square
- 9) Print An Octagon's Square
- 0) Exit

1

Enter triangle's sides

```

1
2
3
Menu:
1) Initialize A Triangle
2) Initialize A Foursquare
3) Initialize An Octagon
4) Print A Triangle's sides
5) Print A Foursquare's sides
6) Print An Octagon's sides
7) Print A Triangle's Square
8) Print A Foursquare's Square
9) Print An Octagon's Square
0) Exit
2
Enter foursquare's side
10
Menu:
1) Initialize A Triangle
2) Initialize A Foursquare
3) Initialize An Octagon
4) Print A Triangle's sides
5) Print A Foursquare's sides
6) Print An Octagon's sides
7) Print A Triangle's Square
8) Print A Foursquare's Square
9) Print An Octagon's Square
0) Exit
3
Enter octagon's side
20
Menu:
1) Initialize A Triangle
2) Initialize A Foursquare
3) Initialize An Octagon
4) Print A Triangle's sides
5) Print A Foursquare's sides
6) Print An Octagon's sides
7) Print A Triangle's Square
8) Print A Foursquare's Square
9) Print An Octagon's Square

```

0) Exit

4

a=1,b=2,c=3

Menu:

1) Initialize A Triangle

2) Initialize A Foursquare

3) Initialize An Octagon

4) Print A Triangle's sides

5) Print A Foursquare's sides

6) Print An Octagon's sides

7) Print A Triangle's Square

8) Print A Foursquare's Square

9) Print An Octagon's Square

0) Exit

5

a=10,b=10,c=10,d=10

Menu:

1) Initialize A Triangle

2) Initialize A Foursquare

3) Initialize An Octagon

4) Print A Triangle's sides

5) Print A Foursquare's sides

6) Print An Octagon's sides

7) Print A Triangle's Square

8) Print A Foursquare's Square

9) Print An Octagon's Square

0) Exit

6

a=20,b=20,c=20,d=20,e=20,f=20,g=20,h=20

Menu:

1) Initialize A Triangle

2) Initialize A Foursquare

3) Initialize An Octagon

4) Print A Triangle's sides

5) Print A Foursquare's sides

6) Print An Octagon's sides

7) Print A Triangle's Square

8) Print A Foursquare's Square

9) Print An Octagon's Square

0) Exit

7

Triangle's square: 0

Menu:

- 1) Initialize A Triangle
- 2) Initialize A Foursquare
- 3) Initialize An Octagon
- 4) Print A Triangle's sides
- 5) Print A Foursquare's sides
- 6) Print An Octagon's sides
- 7) Print A Triangle's Square
- 8) Print A Foursquare's Square
- 9) Print An Octagon's Square
- 0) Exit

8

Foursquare's square: 100

Menu:

- 1) Initialize A Triangle
- 2) Initialize A Foursquare
- 3) Initialize An Octagon
- 4) Print A Triangle's sides
- 5) Print A Foursquare's sides
- 6) Print An Octagon's sides
- 7) Print A Triangle's Square
- 8) Print A Foursquare's Square
- 9) Print An Octagon's Square
- 0) Exit

9

Octagon's square: 1931.37

Menu:

- 1) Initialize A Triangle
- 2) Initialize A Foursquare
- 3) Initialize An Octagon
- 4) Print A Triangle's sides
- 5) Print A Foursquare's sides
- 6) Print An Octagon's sides
- 7) Print A Triangle's Square
- 8) Print A Foursquare's Square
- 9) Print An Octagon's Square
- 0) Exit

0

Triangle deleted

Triangle deleted

Foursquare deleted  
Foursquare deleted  
Octagon deleted  
Octagon deleted

## 4 Выводы

В этой лабораторной были реализованы классы фигур(треугольника, шестиугольника, восьмиугольника). Также были изучены понятия ООП: класс, объект, инкапсуляция, абстрактный тип данных, наследование, полиморфизм.