

## Теоретическое введение

**Тема: Язык C++. Программа на языке C++. Операторы языка C++. Простейшие классы и объекты**

*Программа на языке C++ имеет такой общий вид.*

```
# директива include  
  
Объявление базовых классов  
  
Объявление производных классов  
  
Прототипы обычных функций  
  
int main()  
  
{  
  
...  
  
}  
  
Определения обычных функций
```

### **НОВЫЕ ЗАГОЛОВКИ**

<iostream.h> → <iostream>

Когда мы включаем в программу заголовок нового стиля, содержание этого заголовка оказывается в пространстве имен *std*.

**Пространство имен** – это просто некая объявляемая область, необходимая для того, чтобы избежать конфликтов имен идентификаторов. Чтобы пространство имен стало видимым, используем инструкцию

```
<iostream>  
using namespace std;
```

### **КОНСОЛЬНЫЙ ВВОД/ВЫВОД**

В C++ ввод/вывод выполняется с использованием операторов, а не функций ввода/вывода.

**Оператор вывода** <<

**Оператор ввода** >>

```
cout<<"Эта строка выводится на экран!";
```

Эта инструкция осуществляет вывод строки в заранее определенный поток *cout*, который автоматически связывается с терминалом, когда программа C++ начинает выполняться.

*cout<< выражение;*

В одном выражении ввода/вывода можно выводить более одной величины.

```
cout<<i<< ' '<<j<< ' '<<d;
```

Для считывания значений с клавиатуры используется оператор ввода >>

```
Int num;
```

В общем виде: *cin>> переменная;*

```
cin>>num;
```

**Задание:** Написать программу ввода и вывода целого числа, числа с плавающей точкой и строки символов.

### КОММЕНТАРИЙ

*Однострочный : //*

*Многострочный: /\* ...\*/*

### КЛАССЫ

Классы представляют абстрактные типы данных с открытым интерфейсом и скрытой внутренней реализацией. В классах реализованы базовые принципы объектно-ориентированного программирования (ООП):

- 1) *абстракция данных*;
- 2) *инкапсуляция* – в классах объединяются данные и методы функции для работы с ними, так как лишь через методы возможен доступ к скрытым данным класса;
- 3) *наследование* – в производных классах наследуются члены базового класса;
- 4) *полиморфизм* – возможность использования одних и тех же методов для работы с различными объектами базового и порожденных им классов.

Классы создаются с помощью ключевого слова *class*.

Объявление класса определяет новый тип, связывающий код и данные между собой. Таким образом, класс является логической абстракцией, а объект – ее физическим воплощением, т.е. объект – это экземпляр класса.

```
class имя_класса  
{
```

*Закрытые функции и переменные класса*

**public:**

*открытые функции и переменные класса*

```
} список объектов;
```

По умолчанию функции и данные объявленные внутри класса, считаются закрытыми и доступны лишь функциям этого класса.

Спецификатор **public** – открывает доступ к функциям данным класса из других частей программы.

Спецификатор **protected** необходим при наследовании классов.

```
class number
```

```
{
```

```
int n;
```

```
double w;
```

```
public:
```

```
void set_n (int n);
```

```
int get_n ();
```

```
...
```

```
};
```

Функции, объявленные внутри класса называются **функциями – членами класса**. Они имеют доступ ко всем элементам класса, включая закрытые.

Но хотя функции объявлены в классе, они еще не определены. Для определения функции мы должны связать имя класса с именем функции.

Общий вид:

Тип\_возвращ\_значения **имя класса ::** имя функции (список параметров)

```
{
```

```
Тело функции
```

```
}
```

**::** - оператор расширения области видимости.

```
void number::set_n (int nn)
```

```
{
```

```
n=nn;
```

```
}
```

Чтобы создать объект класса нужно использовать имя класса как спецификатор типа данных.

```

int main()
{
    number ob; //объект типа number
    .....
    ob.set_n(10);
    cout<<"Целое число:"<<ob.get_n();
    return 0;
}

```

После того как объект класса создан можно обратиться к открытым членам класса через оператор – точка (.).

### **Задание:**

**Создать класс (выбрать из предложенных вариантов).**

**Класс должен содержать указанные поля.**

**Добавить интерфейсные функции доступа к полям класса.**

**Написать программу, демонстрирующую работу с данным классом.**

**Вариант 1**  
**СТУДЕНТ**  
 имя - char\*  
 курс – int

**Вариант 3**  
**БИБЛИОТЕКА**  
 название - char\*  
 автор - char\*  
 цена – float

**Вариант 5**  
**КВИТАНЦИЯ**  
 номер - int  
 дата - char\*  
 сумма - float

**Вариант 2**  
**АДРЕС**  
 имя - char\*  
 улица - char\*  
 дом – int

**Вариант 4**  
**ТОВАР**  
 наименование - char\*  
 количество - int  
 цена - float

**Вариант 6**  
**АВТОМОБИЛЬ**  
 марка - char\*  
 мощность - int  
 стоимость - float