

Практическое задание 7.

Использование функций в C++

Цель. Изучение принципов модульной технологии разработки программ и использование их в своем программировании. Изучение синтаксических правил описания функций и обращения к ним. Изучение механизмов обращения к функции, передачи данных в функцию и механизма возвращения данных.

ПЛАН

1. Краткое теоретическое введение	1
2. Функция, возвращающая значение. Оператор-выражение. Формальные и фактические параметры.....	2
3. Функция типа void. Оператор-функция. Фактические параметры при обращении к функции.....	3

1. Краткое теоретическое введение

Описание функции

Чтобы была функция, ее нужно **описать**.

Синтаксис **описания функции**, в общем виде, такой:

```
Тип_функции Имя_Функции (Список формальных параметров)
{
    // Описание локальных данных.
    Тело функции – блок.           // Описание алгоритма.
    return Выражение;               // Оператор выхода.
}
```

Когда функция вычисляет и возвращает значение, то она имеет тип такой же, как и у найденного значения. Найденное в теле функции значение, оператор **return** возвращает в вызывающую функцию.

В описании функции ее параметры – формальные, то есть не имеют реальных значений, а только описывают типы, количество и порядок следования данных, передаваемых в функцию. Записываются именем, например, **float** x.

В теле функции решается задача обработки данных, которые получены через список параметров. Описание алгоритма содержит детали реализации.

В теле функции могут быть описания локальных переменных. Их область действия – только тело функции.

Возврат в точку вызова обеспечивает оператор **return**:

```
return Выражение;
```

Где тип выражения совпадает с типом функции.

Обращение к функции

Чтобы использовать функцию, к ней нужно **обратиться**.

Обращение к функции, это передача управления в тело функции, а также передача фактических данных, с которыми функция отрабатывает вызов.

Если функция возвращает значение, то к ней обращаются через оператор-выражение, синтаксис которого:

```
Имя_Функции (Список фактических параметров)
```

Это значение может быть присвоено, или использоваться иным образом.

Параметры функции в обращении – фактические, это те реальные значения, с кото-

рыми функция обрабатывает очередной вызов. Фактическими параметрами могут быть константы, переменные, выражения, указатели.

2. Функция, возвращающая значение. Оператор-выражение. Формальные и фактические параметры

Создайте проект, скопируйте в папку проекта заголовочный файл `Function.h`, присоедините его к проекту, откройте и ознакомьтесь с содержимым. Все описания функций выполняются в заголовочном файле, все управления вызовами – в `Source` файле проекта.

Устно ответьте на вопросы.

1. Что содержит заголовочный файл.
2. Как он подключается к программе.
3. В чем преимущества использования таких файлов.

Упражнение 1. Обращение к функции

Обращение к функции, это исполнение алгоритма каждый раз с различными данными. Фактическими параметрами при обращении могут быть переменные или выражения. Функция возвращает одно значение, которое есть одно данное базового типа, поэтому обращение к ней, это оператор-выражение.

Напишите вызывающую программу, которая обращается к функции `Avg` тремя различными способами.

```
//Пример обращения к функции.
#include "Function.h"                // Доступ к описанию функции.
void main (void)
{
    float x1=2.5, x2=3.5, x3=5.0;
    float y = Avg (x1, x2, x3);      // Фактические параметры – переменные.
    // Выведем результат обращения
    cout << "x1=" << x1 << " x2=" << x2 << " x3=" << x3 << " Y=" << y << endl;
    y = Avg (2.,3.,7.);              // Фактические параметры – константы.
    // Выведите результат обращения.
    y = Avg (x1, 5, x3/2);           // Фактические параметры – выражения.
    // Выведите результат обращения.
}
```

Устно ответьте на вопросы.

1. Что такое формальные параметры функции.
2. Что такое фактические параметры функции.
3. Каково соотношение между ними.
4. Что такое сигнатура параметров, и какие требования к ней предъявляются.

Упражнение 2. Описание функции и обращение к ней

В заголовочном файле опишите функцию, которая возвращает площадь треугольника по заданным значениям длин сторон. Если треугольник не существует, функция возвращает 0.

Протестируйте функцию, обратившись к ней трижды, со значениями длин сторон (3.,4.,5.), затем (1.,1.,3.), затем (4.,4.,4.).

3. Функция типа void. Оператор-функция. Фактические параметры при обращении к функции

Функции типа `void` встречаются редко, чаще всего, используются для интерфейсных функций, или когда значения возвращаются через параметры. Обращение к ним, это оператор-функция, по форме выглядит как обычный оператор. Как пример рассмотрим вычисление и вывод таблицы значений функции.

Входными данными являются.

1. Диапазон изменения аргумента, количество точек на этом диапазоне.
2. Сама функция, для которой строится таблица.

Упражнение 3. Обращение к функции

Обратитесь к функции `Table` со значениями $x_0 = 0.$, $x_n = \pi.$, количество точек = 12.

Упражнение 4. Описание функции

Пусть требуется вычислить функцию $Y(x) = A \cdot \sin(\varphi \cdot x)$.

Напишите новую функцию `Table_A`, так, чтобы ее описание соответствовало новому условию.

Обратитесь в цикле со значениями параметра $A = 1, 1.5, 2., 2.5$.

Бонус

Опишите функцию, которая может вычислять таблицу любой зависимости $Y(x)$. Для этого нужно научиться передавать функцию в функцию.