

Практическая работа 6

«Проектирование программ с использованием функций»

1 Цель работы:

- 1.1 Формирование умения разрабатывать алгоритмы для конкретных задач.
- 1.2 Формирование умения разрабатывать подпрограммы.

2 Литература:

2.1 Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1735805>. – Режим доступа: по подписке

3 Подготовка к работе:

- 3.1 Подготовить отчет на C:\Temp\KSK-31\;
- 3.2 Повторить изученный материал.

4 Основное оборудование:

- 4.1 Персональный компьютер

5 Задание:

- 5.1 Внимательно изучите приложение к лабораторной работе и выполните задания из пункта 6. Оформите в виде отчета.

6 Порядок выполнения работы:

- 6.1 Внимательно изучить приложение.

6.2 Разработать функцию для поиска степени числа. Функция принимает два аргумента — само число и степень. Для возведения числа в какую-то степень необходимо его умножить на самого себя столько раз, сколько указано в степени т.е. для возведения 2 в степень 4 нужно $2 * 2 * 2 * 2$. При реализации функции **не использовать** функцию pow, использовать цикл.

6.3 Разработать функцию для поиска минимального из двух чисел. Найти минимальное из трех чисел с помощью этой функции. Объяснение: функция после выполнения вернет наименьшее из двух чисел. Можно сначала сравнить с помощью вашей функции первую пару чисел, а потом вторую пару чисел.

6.4 Разработать процедуру, меняющую цвет консоли на выбранный пользователем. Пользователю на выбор предоставляется три цвета. Например, при выборе пункта 1 — сменить цвет на красный. При выборе некорректного цвета — сменить цвет консоли на черный.

6.5 Разработать **алгоритм** поиска минимального из двух чисел с помощью программы Visio.

7 Содержание отчета:

- 7.1 Цель работы;
- 7.2 Содержание действий по выполнению данной работы;
- 7.3 Ответы на контрольные вопросы;
- 7.4 Вывод.

8 Контрольные вопросы

1. В чем разница между функцией и числом.
2. Каков синтаксис объявления функции.
3. Что такое синтаксис?
4. Как выглядит элемент «подпрограмма».

9 Приложение

В вашей программе **уже существует одна функция — int main()** или `int t_main()`. Это основная функция, точка входа в программу, в ней начинается выполнение кода и заканчивается. Если писать весь необходимый код внутри нее, можно сильно запутаться. Так же есть код, который очень часто повторяется. Поэтому для выноса часто повторяющихся блоков кода или просто для оптимизации кода могут использоваться функции.

Функция - подпрограмма, выполняющая какие-либо операции и возвращающая значение.

Процедура - подпрограмма, которая только выполняет операции, без возврата значения

Синтаксис (правило) объявления функции:

```
тип имя_функции(параметры)
{
    инструкции
}
```

Первая строка представляет заголовок функции. Вначале указывается **возвращаемый тип** функции. Если функция **не возвращает** никакого значения, то используется **тип void**.

Затем идет **имя функции**, которое представляет произвольный идентификатор. К именованию функции применяются те же правила, что и к именованию переменных.

После имени функции в скобках идет перечисление **параметров**. Функция может не иметь параметров, в этом случае указываются пустые скобки.

После заголовка функции в фигурных скобках идет **тело функции**, которое содержит выполняемые инструкции.

Для возвращения результата функция применяет **оператор return**. Если функция имеет в качестве возвращаемого типа **любой тип, кроме void**, то она должна обязательно с помощью оператора return возвращать какое-либо значение.

Например, определение функции main, которая должна быть в любой программе на языке C++ и с которой начинается ее выполнение:

```
int main()
{
    return 0;
}
```

Возвращаемым типом функции является тип int, поэтому функция должна использовать оператор return и возвращать какое-либо значение, которое соответствует типу int. Возвращаемое значение ставится после оператора return.

Другие функции, коорые вы будете создавать, **НУЖНО ПИСАТЬ ДО main и ПОСЛЕ БИБЛИОТЕК**

Если функция имеет тип void, то ей не надо ничего возвращать. Например, мы могли бы определить следующую функцию, которая просто выводит некоторый текст на консоль:

```
void hello()
{
    cout << "hello" << endl;
}
```

Выполнение функции

Когда запускается программа на языке C++, то запускается функция **main**. Никакие другие функции, определенные в программе, автоматически не выполняются. Для выполнения функции ее необходимо вызвать. Вызов функции осуществляется в форме:

```
имя_функции(аргументы);
```

После имени функции указываются скобки, в которых перечисляются аргументы - значения для параметров функции.

Например, определим и выполним простейшую функцию:

```
#include <iostream>
#include <string>
using namespace std;
void hello()
{
    cout << "hello" << endl;
}
```

```
int main()
{
hello();
hello();
}
```

Здесь определена функция `hello`, которая вызывается в функции `main` два раза. В этом и заключается преимущество функций: мы можем вынести некоторые общие действия в отдельную функцию и затем вызывать многократно в различных местах программы. В итоге программа два раза выведет строку "hello".

```
hello
hello
```

Функция, принимающая аргументы

Пример функции, приветствующей пользователя по имени:

```
#include <iostream>
#include <string>
using namespace std;
//наша функция, принимает в качестве параметра
//строку
void hello(string name)
{
    //выводит приветствие
    cout << "hello, " << name<<endl;
    //так как функция имеет тип void, return не нужен
}
int main()
{
    //вызов функции
    string name;
    cout << "Enter name: ";
    cin >> name;
    //вызов функции, в качестве параметра переменная
    hello(name);
    //вызов функции, в качестве параметра строка
    hello("Polina");
}
```

Пример функции, складывающей два числа:

```
#include <iostream>
#include <string>
using namespace std;
```

```

//функция приветствия
void hello(string name)
{
    //выводит приветствие
    cout << "hello, " << name<<endl;
    //так как функция имеет тип void, return не нужен
}
//функция сложения чисел
int sum(int a, int b) {
    //тип возвращаемого значения int, поэтому
    //есть return
    return a + b;
}
int main()
{
    cout << "Введите числа";
    int a, b;
    cin >> a;
    cin >> b;
    //результат выполнения возвращается,
    //но не выводится, с ним можно выполнять всякие вычисления
    //но мы просто выведем в консоль
    cout << sum(a, b); //в кач параметра для вызова два числа
}

```

Изменение цвета фона окна

Можно менять цвет фона окна и символов без дополнительных библиотек, простой функцией `system()`. Достаточно написать, например, `system("color F0")` и будет установлен белый фон (F) и чёрный текст (0). Можно писать и наоборот, например: `system("color 3F");` установит голубой фон и белый текст.

Список цветов:

- 0 — черный
- 1 — синий
- 2 — зеленый
- 3 — голубой
- 4 — красный
- 5 — лиловый
- 6 — желтый
- 7 — белый
- 8 — серый
- 9 — светло-синий
- A — светло-зеленый
- B — светло-голубой
- C — светло-красный
- E — светло-желтый
- F — ярко-белый