Белорусский государственный технологический университет

**Реферат**

на тему «Прототипы функций»

Работу выполнил:

Студент ИСиТ 1-1

**Палазник А.В.**

Минск 2021

**Задача:** узнать, что такое прототип функции, для чего он нужен, и в дальнейшем сделать о нем определенные выводы

**Алгоритм решения:**

**1)**Определение прототипа функции

**2)**Для чего нужен прототип функции

**3)**Описать простейшую структуру прототипа функции и показать ее работоспособность

**4)**Прототип функции с параметрами

**5)**Вывод

**Определение прототипа функции**

Прототипом функции в языке C или C++ называется объявление функции, не содержащее тела функции, но указывающее имя функции, арность, типы аргументов и возвращаемый тип данных.

В то время как определение функции описывает, что именно делает функция, прототип функции может восприниматься как описание её интерфейса.

**Для чего нужен прототип функции**

Бывают ситуации, когда в «функции1» нам нужно вызвать «функцию2», однако, у нас не получится это сделать, если «функция2» была создана после «функции1». Конечно, можно просто поменять местами объявления «функции1» и «функции2», однако, если в нашей программе много функций, с этим могут возникнуть проблемы. Для таких случаев и существуют прототипы функции. Они как бы «указывают», на наличие в коде нашей функции

**Описать простейшую структуру прототипа функции и показать ее работоспособность**

Простейшая структура прототипа функции выглядит следующим образом:

<тип данных><имя функции>;

Покажем пример кода с использованием прототипа функции и без его использования

Пример без прототипов:

#include <iostream>

using namespace std;

int main() {

fun1(); // функция fun1 не была найдена

}

void fun1()

{

cout << "I'm fun1()\n";

fun2(); // функция fun2 не была найдена

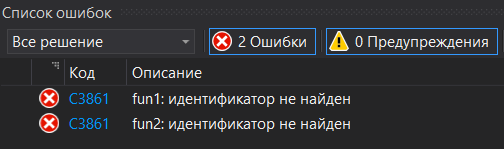
}

void fun2() {

cout << "I'm fun2()\n";

}

Результат кода:



Пример с прототипами:

#include <iostream>

using namespace std;

void fun1();

void fun2();

int main() {

fun1();

}

void fun1()

{

cout << "I'm fun1()\n";

fun2();

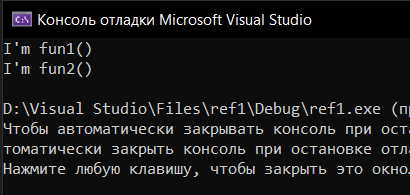
}

void fun2() {

cout << "I'm fun2()\n";

}

Результат кода:



**Прототип функции с параметрами**

Существует 2 вида прототипов с функциями:

1)Полноценный

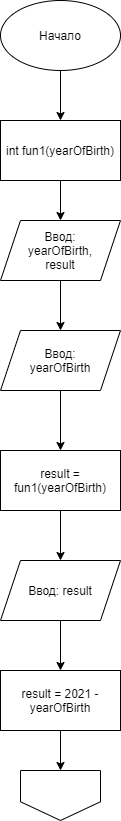
2)Краткий

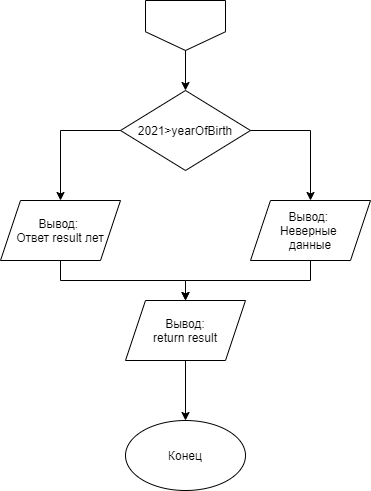
Полноценный прототип функции состоит из типа возврата функции, ее имени и параметров(тип + имя параметра).

В кратком прототипе отсутствуют имена параметров функции.

Задача: Вычислить количество полных лет, зная год рождения человека

Решение: Сперва, сделаем блок-схему задания





Далее по нашей блок-схеме напишем код:

#include <iostream>

using namespace std;

int fun1(int yearOfBirth);

int main() {

setlocale(LC\_ALL, "Russian");

int yearOfBirth;

int result;

cout << "Введите год рождения: \n";

cin >> yearOfBirth;

result = fun1(yearOfBirth);

}

int fun1(int yearOfBirth) {

int result;

result = 2021 - yearOfBirth;

if (2021 >= yearOfBirth) {

cout << "Ответ: " << result << " лет";

}

else {

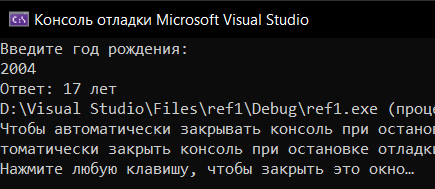
cout << "Неверные данные";

}

return result;

}

Результат кода:



**Вывод**

Прототип функции – это прекрасный инструмент, с помощью которого мы можем не беспокоится о порядке наших объявленных функций, просто вписав прототипы в начале кода