



## **Введение в язык Си**

**Чернецов Андрей Михайлович,**

**К.т.н. доцент каф. ПМ**

# Список литературы по курсу

- 1. Князев А.В. Основы языка C++. Учебное пособие. М.: Издательство МЭИ, 2013 – 80 с. ISBN 978-5-7046-1425-8.
- 2. Князев А.В. Работа со сложными структурами данных на языке C++. Учебное пособие. М.: Издательство МЭИ, 2015 – 48 с. ISBN 978-5-7046-1658-0
- 3. Программирование. Сборник задач. Учебное пособие. Санкт-Петербург: Лань, 2019 – 140 с. ISBN 978-5-8114-3857-0

URL: <https://e.lanbook.com/book/121485>

# Структура программы на С

```
#include "stdafx.h"
```

В конце обычного оператора ставится;

```
int main ()
```

Операторные скобки { }

```
{
```

описание переменных

операторы..

```
return 0;
```

```
}
```

- Регистр букв важен:

А и а – это разные идентификаторы

Return и return – разные слова

# Описание переменных

Примеры имен переменных: `x`, `y`, `summa`, `s1`, `sred` и т.д.

Имена переменных не должны совпадать с ключевыми словами языка C.

Чтобы использовать в программе переменную, необходимо:

1. объявить переменную в начале программы, явно указав тип данных для переменной.

Пример:

```
double x; //вещественная переменная
```

```
int m; //целочисленная переменная
```

Если переменная не будет объявлена, но будет использоваться далее в программе, то программа не запустится, компилятор выдаст ошибку.

2. проинициализировать переменную, т.е. задать переменной значение.

Пример:

```
x=7.81; m=4; z=x+m;
```

Если переменная не будет проинициализирована, то компилятор не выдаст ошибки, но расчеты будут выполнены **неверно**.

# Стандартные типы данных C

Тип данных	Значение
int	целый тип, размер типа int не определяется стандартом, а зависит от компьютера и компилятора, Примеры значений типа int: 5, 0, -1, 100.
float	вещественный тип. В компьютерах величины типа float занимают 4 байта
double	вещественный тип с двойной точностью. Примеры значений типа double: 5.0, -0.00001, 2.9987.
char	символьный тип, под величину символьного типа отводится количество байт, достаточное для размещения любого символа из набора символов

# Ввод-вывод

Способ 1. Использование функций ввода-вывода

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    printf ("Enter positive num"); // Вывод
```

```
    scanf ("%d", &num); // Ввод
```

```
    printf (" Введено число %d\n", num);
```

```
    return 0
```

```
}
```

Спецификатор формата	Смысл
%f	Вещественные числа
%d	Целые числа
%s	текст

# Ввод-вывод

Способ 2. Использование операторов cin / cout

cout – оператор вывода данных на экран.

Пример использования:

```
cout << "abc"; //выведет на экран слово abc. Может вывести любой текст.
```

```
cout <<x; //выведет на экран число, хранящееся в переменной x
```

cin – оператор считывания с экрана.

Когда у пользователя запрашивают число, программа ждёт, пока пользователь не введёт число и нажмет ENTER. Тогда оператор cin записывает это значение в переменную x.

```
cin >>x; //присваивает переменной x значение, введённое с клавиатуры.
```

```
cin >>x>>y; //присваивает переменной x первое введённое с клавиатуры значение, переменной y – второе.
```

```
cout <<endl; //курсор перейдёт на новую строку.
```

# Математические функции в C

- `#include <math.h>`

Программная запись	Описание
<code>fabs(x)</code>	Модуль числа.
<code>sin(x)</code>	Синус числа, аргумент в радианах.
<code>cos(x)</code>	Косинус числа, аргумент в радианах.
<code>tan(x)</code>	Тангенс числа, аргумент в радианах.
<code>exp(x)</code>	Экспонента числа.
<code>log(x)</code>	Натуральный логарифм числа.
<code>log10(x)</code>	Десятичный логарифм числа.
<code>pow(x, y)</code>	x в степени y.
<code>pow10(x)</code>	Степень десяти.
<code>sqrt(x)</code>	Квадратный корень из числа.
<code>asin(x)</code>	Арксинус числа, в радианах.
<code>acos(x)</code>	Арккосинус числа, в радианах.
<code>atan(x)</code>	Арктангенс числа, в радианах.
<code>M_PI</code>	Число =3.141593

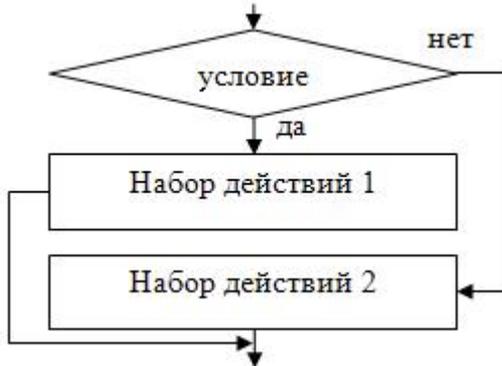
# Основные арифметические операции C

Операция в C	Описание	Пример в программе
=	присвоение	x=0.4;
+	сложение	x=8+3; y=x+7.9;
-	вычитание	x=8-3; y=x-7.9;
*	умножение	x=4*2.5; y=x*4;
/	деление.	x=7.5/3.2; y=x/2.6;
		z=1/3; //z=0
	Результатом деления <b>целых чисел</b> является целое число.	(округление до целого, т.к. делятся целые числа) z=1.0/3.0; //z=0.333(3)

# Логические операции C

Сравнение в C++	Описание	Пример в программе
>	больше чем	x>0
<	меньше чем	y<z
>=	больше или равно	y>=x
<=	меньше или равно	z<=8.51
==	проверка на равенство	x==0.3
!=	не равно	x!=y
&&	логическое И	x>0 && x<1 // двойное неравенство 0<x<1
	логическое ИЛИ	s<7    s>10

# Условный оператор



```
if (условие)
{
    Набор_действий_1;
}
else
{
    Набор_действий_2;
}
```



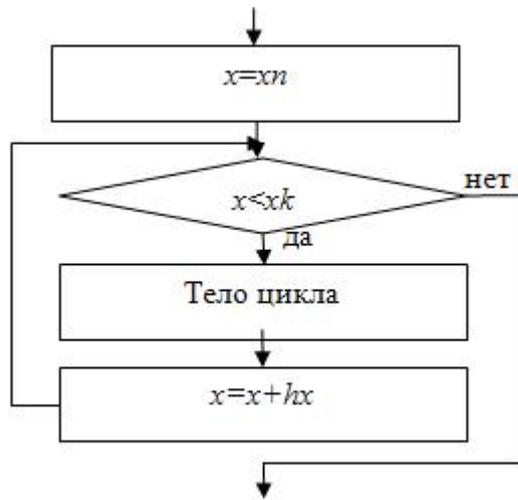
```
if (условие) {
    Набор_действий_1;
}
```

# Циклы while и for

Оператор while:

$x=xn$ ;

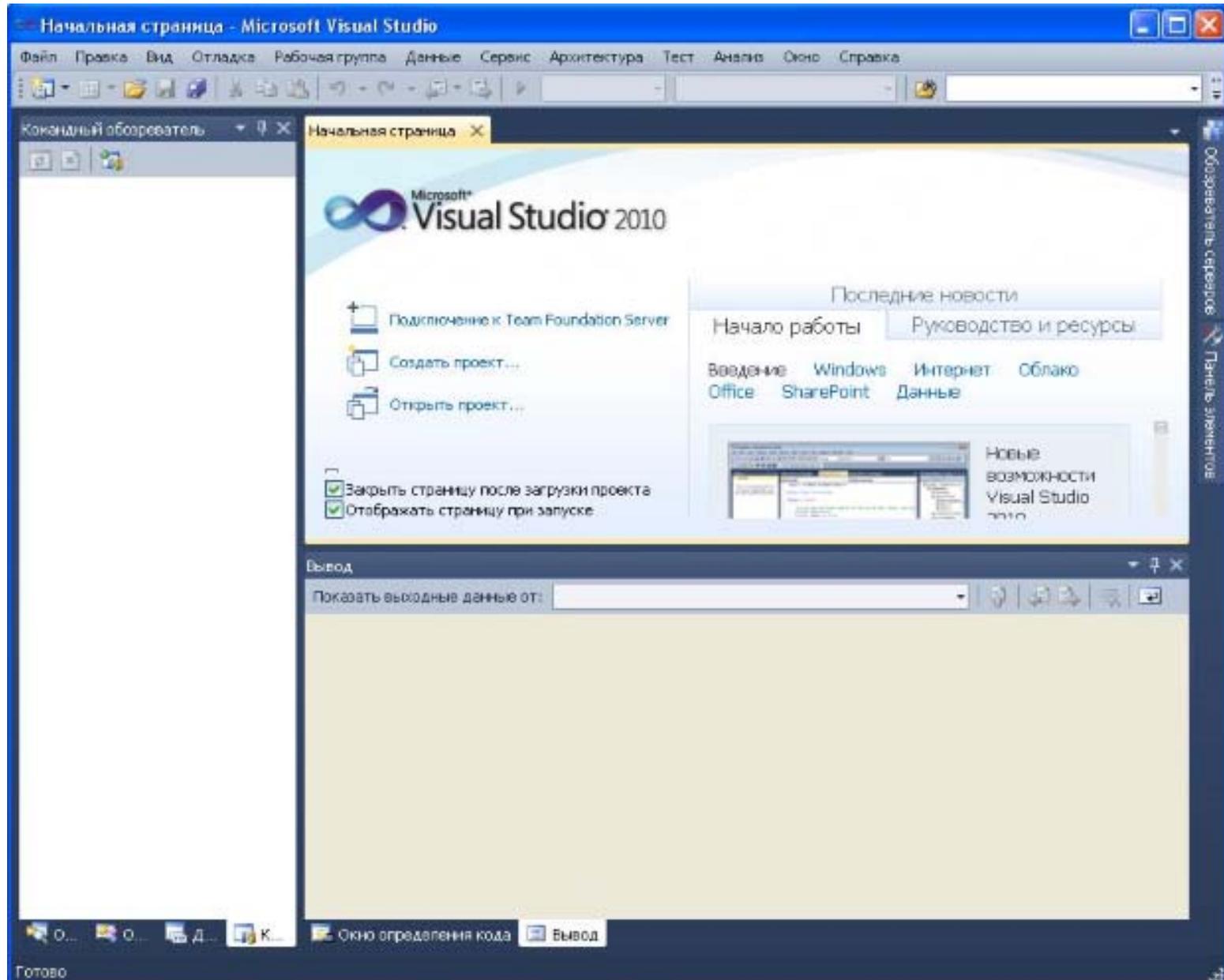
```
while( $x \leq xk$ ){  
    Тело_цикла;  
     $x=x+hx$ ; }
```



Оператор for:

```
for( $x=xn$ ;  $x \leq xk$ ;  $x=x+hx$ )  
    { Тело_цикла; }
```

# Интерфейс MS Visual Studio



# Создать проект



Последние шаблоны

.NET Framework 4

Сортировать по: По умолчанию

Установленн

## Установленные шаблоны

- Visual C++
  - ATL
  - CLR
  - Общие
  - MFC
  - Тест
  - Win32
- Другие языки
- Другие типы проектов
- Базы данных
  - Проекты моделирования
- Тестовые проекты

	Консольное приложение Win32	Visual C++
	Проект Win32	Visual C++

Тип: Visual C++  
Проект по созданию консольного приложения Win32

## Шаблоны в Интернете

Имя:

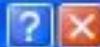
Расположение:

Имя решения:

Обзор...

- Создать каталог для решения
- Добавить в систему управления версиями

OK Отмена



## Добро пожаловать в мастер приложения Win32

Обзор

Параметры приложения

Текущие параметры проекта:

- Консольное приложение

Для подтверждения текущих параметров нажмите кнопку **Готово** из любого окна.

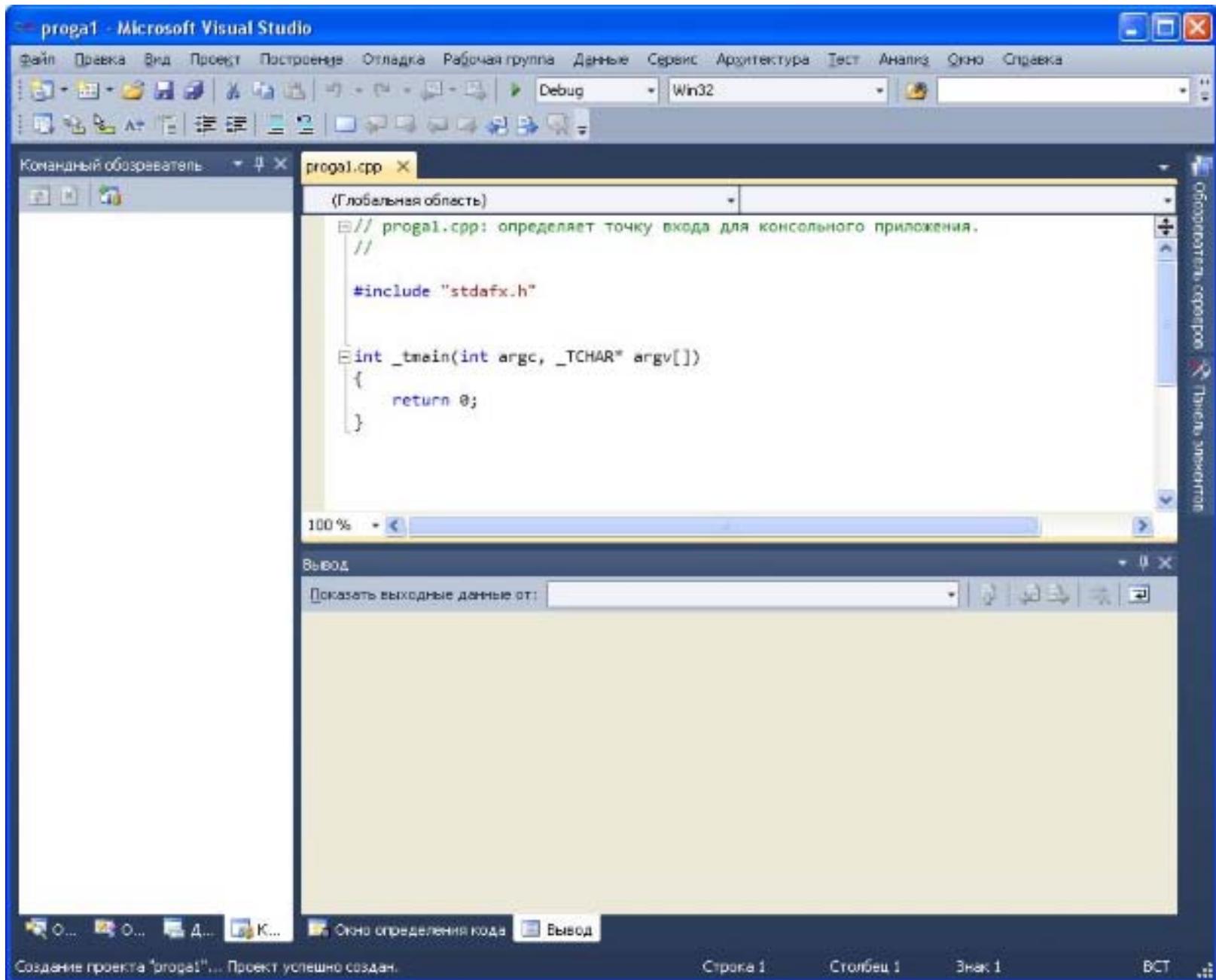
После создания проекта просмотрите файл `readme.txt` данного проекта для получения сведений о проекте и создаваемых файлах.

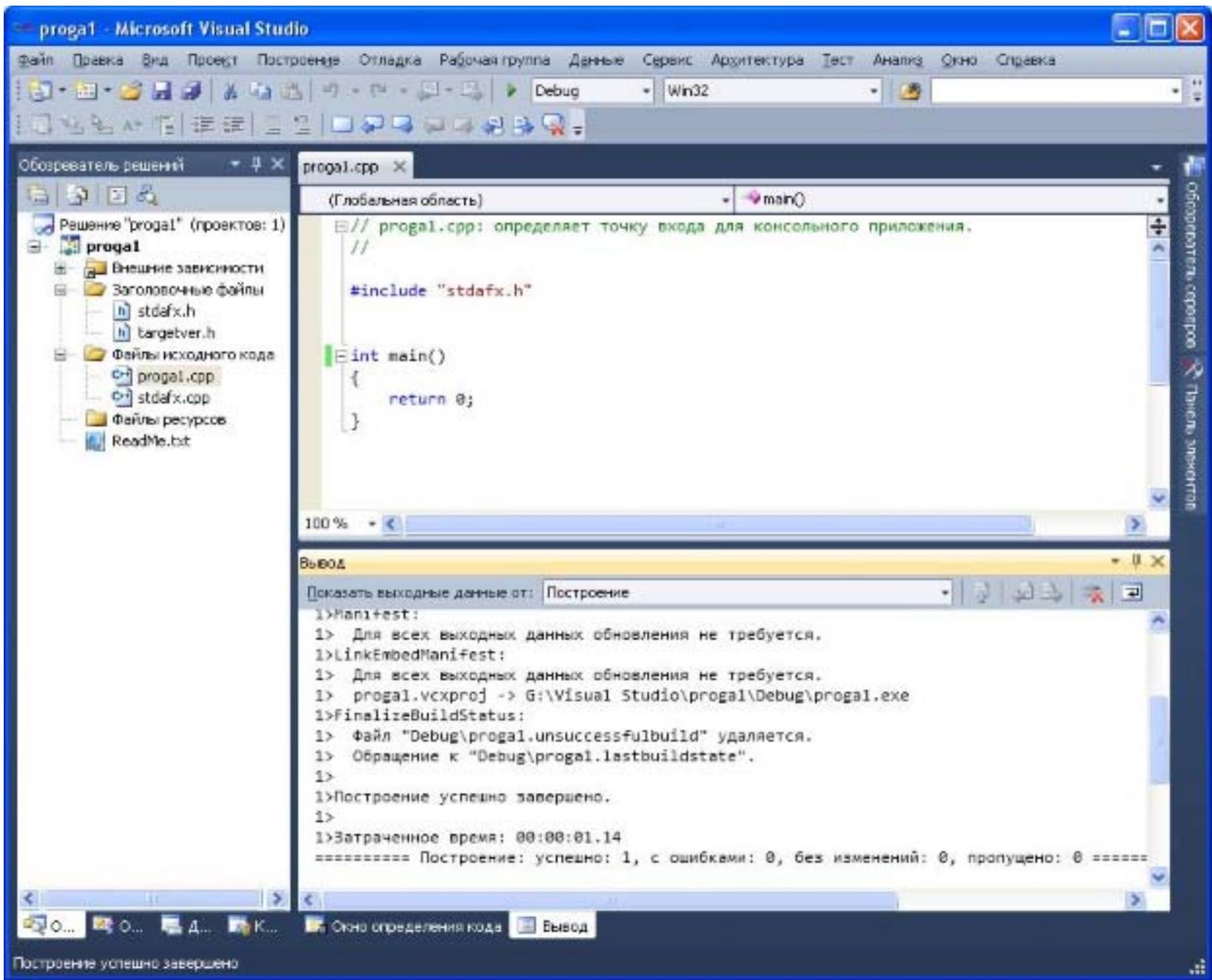
< Назад

Далее >

Готово

Отмена





Вывод

Показать выходные данные от: Построение

```
1>Manifest:
1> Для всех выходных данных обновления не требуется.
1>LinkEmbedManifest:
1> Для всех выходных данных обновления не требуется.
1> progal.vcxproj -> G:\Visual Studio\progal\Debug\progal.exe
1>FinalizeBuildStatus:
1> Файл "Debug\progal.unsuccessfulbuild" удалится.
1> Обращение к "Debug\progal.lastbuildstate".
1>
1>Построение успешно завершено.
1>
1>Затраченное время: 00:00:01.14
===== Построение: успешно: 1, с ошибками: 0, без изменений: 0, пропущено: 0 =====
```

Построение успешно завершено

# Пример

Задано целое, положительное значение переменной  $n$  и вещественное значение переменной  $x$ .

Подсчитать значение  $S$ , получаемое по формуле:

$$S = \sum_{i=1}^n \frac{x+i}{i^2} = \frac{x+1}{1} + \frac{x+2}{4} + \dots + \frac{x+n}{n^2}$$

<u>Исходные данные</u>				
n	Число шагов	целый	простая переменная	n>0
x	Значение аргумента	вещественный	простая переменная	

<u>Выходные данные</u>				
S	Значение суммы	вещественный	простая переменная	

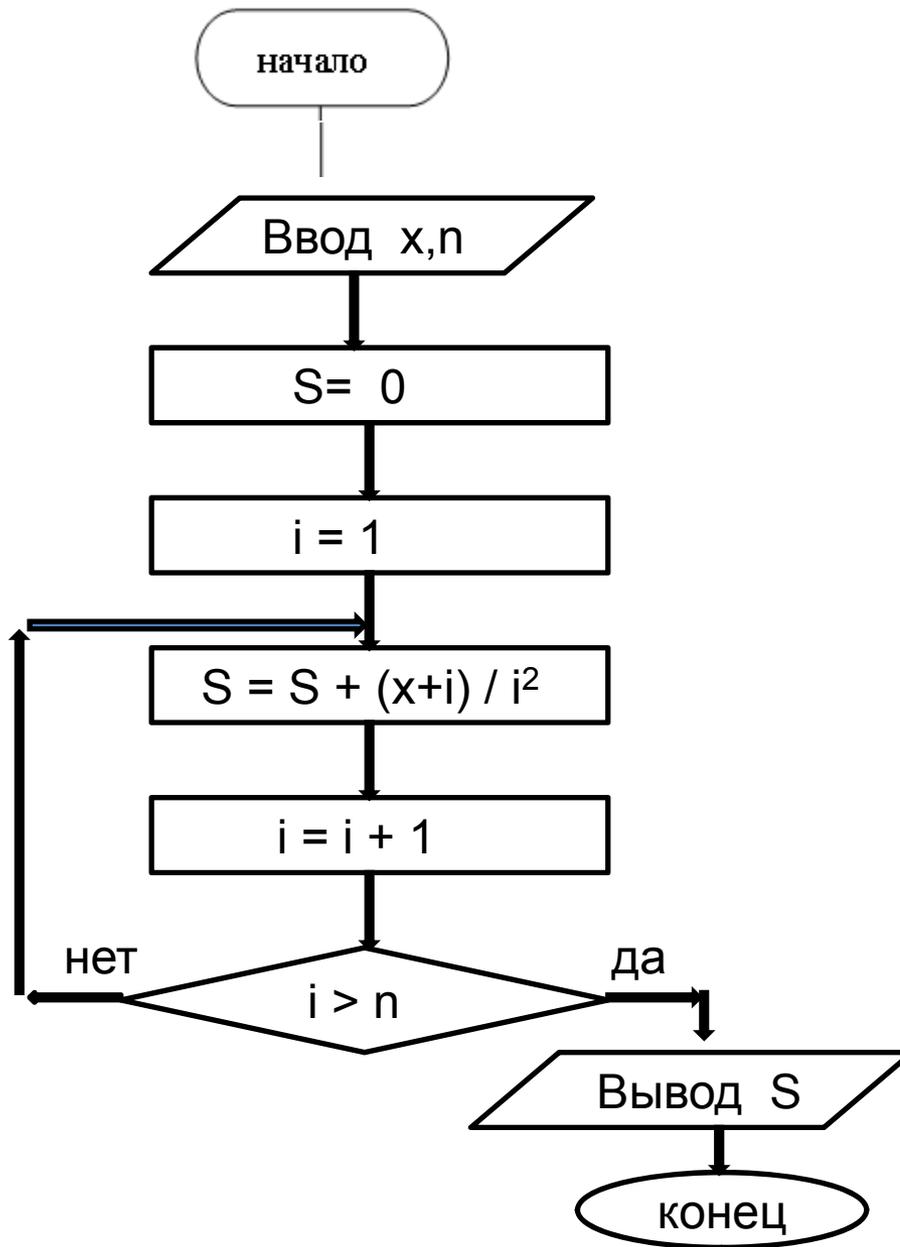
<u>Промежуточные данные</u>				
i	Порядковый номер текущего шага	целый	простая переменная	

Необходимо составить алгоритм решения этой задачи.

Решим задачу методом накопления.

Это значит сначала положим начальное значение переменной  $S$  равным нулю.

Потом к значению  $S$  прибавим первое слагаемое, затем второе, третье и так далее, до тех пор пока не просуммируем все  $n$  слагаемых.



# Код программы

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    float x, S;
```

```
    int n, i;
```

```
    printf ("Enter x \n");
```

```
    scanf ("%f", &x);
```

```
    printf ("Enter n \n");
```

```
    scanf ("%d", &n);
```

```
    S=0.0;
```

```
    for (i=1; i<N; i++) {
```

```
        S = S +(x+i)/(i*i);
```

```
    }
```

```
    printf ("S= ", S);
```

```
    return 0;
```

```
}
```

# Пример

Вычислить значение  $K$ -го члена ряда Фибоначчи.

Числовой ряд, в котором каждое последующее число равно сумме двух предыдущих, носит название чисел Фибоначчи.

Первое и второе число ряда равны единице.

# Метод решения

Исходное данное  $K$  – номер искомого члена ряда. По смыслу задачи значение  $K$  должно быть больше двух.

Обозначим через  $X$  искомое значение, а через  $X_1$  и  $X_2$  – значения двух предыдущих членов ряда.

Тогда  $X = X_1 + X_2$  .

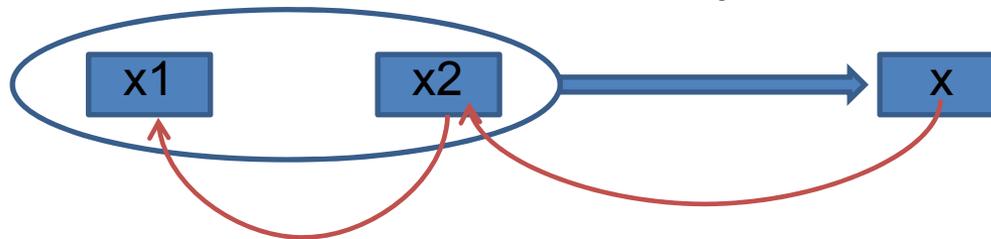
Обозначим через  $i$  номер подсчитанного значения  $X$ .

Решение начинается со ввода значения  $K$ .

После этого надо проверить, является ли полученное значение допустимым, это называется проверкой аномалий.

Если аномалии нет, можно решать задачу.

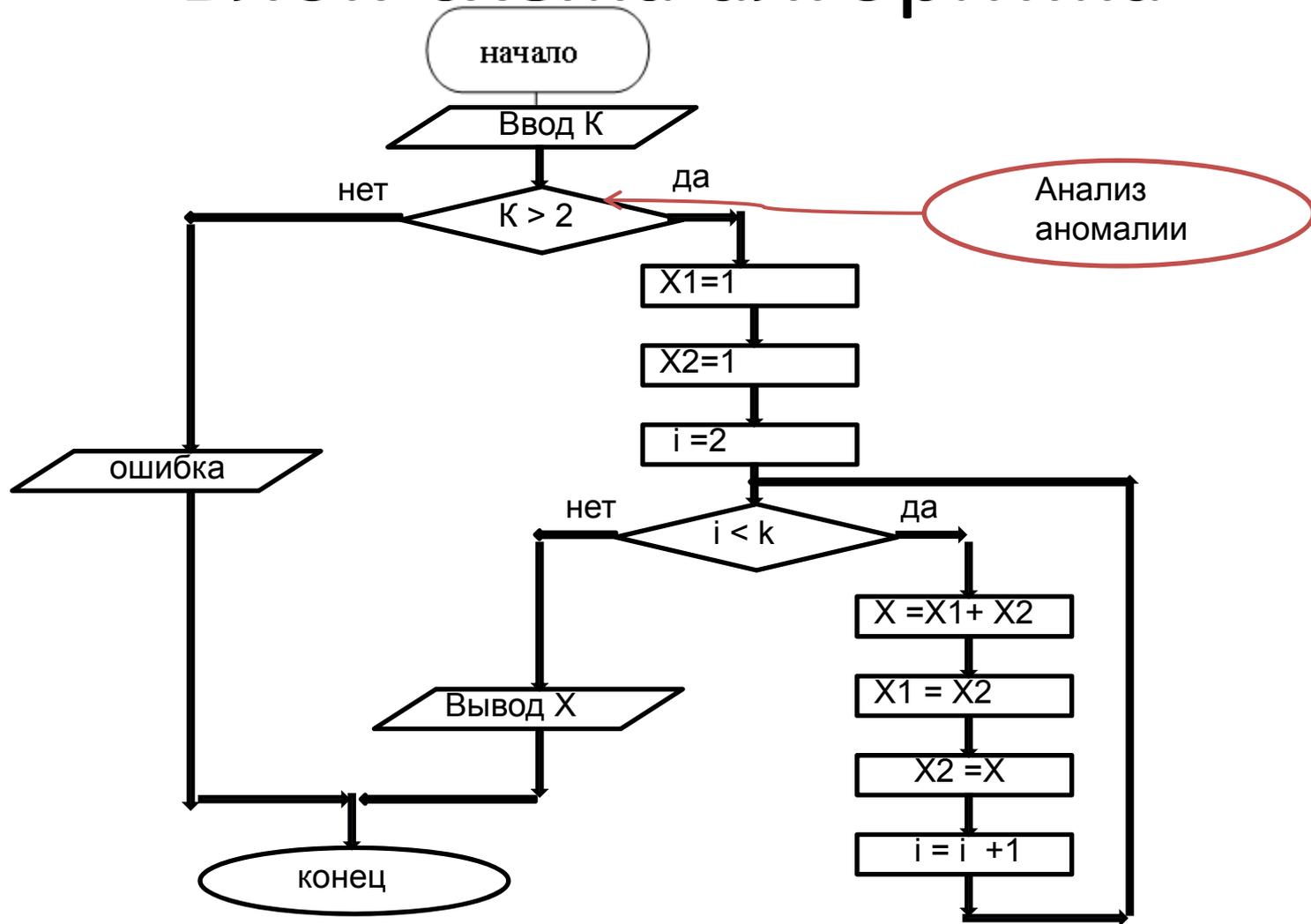
Решение начинается с присвоения переменным  $X_1$  и  $X_2$  начального значения, после этого можно рассчитать значение  $X$ .



При этом  $i$  получает следующее значение. Если  $i = K$ , то результат получен, и его надо напечатать (значение  $X$ ), в противном случае надо подготовить следующий шаг подсчёта  $X$ , для этого надо переписать значение  $X_2$  в  $X_1$  и значение  $X$  в  $X_2$ .

Таким образом, в  $X_1$  и  $X_2$  будут находиться два последних рассчитанных члена ряда, и можно получить следующий член  $X$ .

# Блок-схема алгоритма



```
#include <stdio.h>
```

```
int main ()
```

```
{  
    int K, i;           return 0;  
    int X1, X2;        }  
  
    printf ("Enter K \n");  
    scanf ("%d",&K);  
  
    if ( K >2) {  
        X1=1; X2=1;
```

```
        for (i=2; i<K; i++) {  
            X=X1+X2;  
            X1=X2;  
            X2 = X;
```

```
        } //for
```

```
        printf ("X= ",X);
```

```
    } //if
```

```
    else
```

```
{
```

```
    printf ("Error");
```

```
}
```