

# Программирование на C++



| Минцифры  
РОССИИ

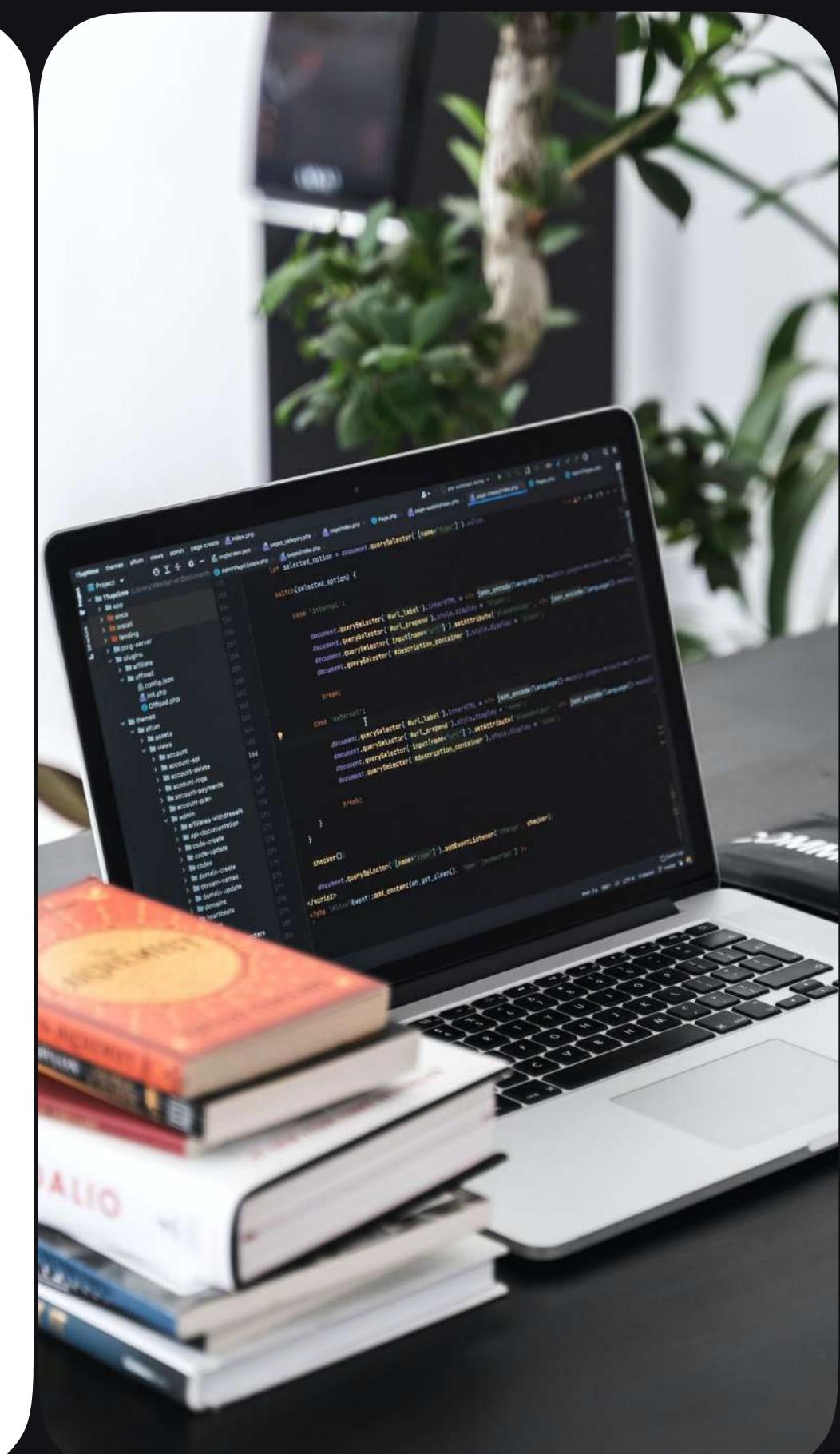
UCHI DOMA

**20.35**  
УНИВЕРСИТЕТ

Урок 2 Модуль 1

# Операторы ввода, вывода

Полезные материалы



## Цели урока

- ★ Изучить операторы ввода, вывода
- ★ отработать на практике написание программ с выводом данных на Си

# Операторы ввода, вывода



В базовом наборе языка С нет операторов ввода, вывода.

Ввод и вывод информации осуществляется через функции стандартной библиотеки `stdio.h`. Эта библиотека содержит функции:

- ❖ `printf( )` – для вывода информации
- ❖ `scanf( )` – для ввода информации

Для использования функций библиотеки, ее необходимо подключить:

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

# Структура программы

```
1 #include <stdio.h>
2 int main ( )
3 {
4     return 0;
5 }
```

# Оператор вывода

Функция `printf( )` предназначена для форматированного вывода. Она переводит данные в символьное представление и выводит полученные изображения символов на экран. При этом у программиста имеется возможность форматировать данные, то есть влиять на их представление на экране.

Общая форма записи функции `printf( )`:

```
printf("СтрокаФорматов", объект1, объект2, ..., объектn);
```

# Оператор вывода

```
printf("СтрокаФорматов", объект1, объект2, ..., объектn);
```

СтрокаФорматов состоит из следующих элементов:



управляющих символов



текста, представленного для непосредственного вывода



форматов, предназначенных для вывода  
значений переменных различных типов



Объекты могут отсутствовать.

# Оператор вывода

Управляющие символы не выводятся на экран, а управляют расположением выводимых символов. Отличительной чертой управляющего символа является наличие обратного слэша \ перед ним.

Основные управляющие символы:



\n – перевод строки



\t – горизонтальная табуляция



\v – вертикальная табуляция



\b – возврат на символ



\r – возврат на начало строки



\a – звуковой сигнал

# Оператор вывода

Форматы нужны для того, чтобы указывать вид, в котором информация будет выведена на экран. Отличительной чертой формата является наличие символа процент `%` перед ним:

- ★ `%d` – целое число типа `int` со знаком в десятичной системе счисления
- ★ `%u` – целое число типа `unsigned int`
- ★ `%x` – целое число типа `int` со знаком в шестнадцатеричной системе счисления
- ★ `%o` – целое число типа `int` со знаком в восьмеричной системе счисления
- ★ `%hd` – целое число типа `short` со знаком в десятичной системе счисления
- ★ `%hu` – целое число типа `unsigned short`
- ★ `%hx` – целое число типа `short` со знаком в шестнадцатеричной системе счисления

# Оператор вывода

Форматы нужны для того, чтобы указывать вид, в котором информация будет выведена на экран. Отличительной чертой формата является наличие символа процент `%` перед ним:

- ★ `%ld` – целое число типа `long int` со знаком в десятичной системе счисления
- ★ `%lu` – целое число типа `unsigned long int`
- ★ `%lx` – целое число типа `long int` со знаком в шестнадцатеричной системе счисления
- ★ `%f` – вещественный формат (числа с плавающей точкой типа `float`)
- ★ `%lf` – вещественный формат двойной точности (числа с плавающей точкой типа `double`)
- ★ `%e` – вещественный формат в экспоненциальной форме (числа с плавающей точкой типа `float` в экспоненциальной форме)
- ★ `%c` – символьный формат
- ★ `%s` – строковый формат

# Оператор вывода

Строка форматов содержит форматы для вывода значений. Каждый формат вывода начинается с символа %. После строки форматов через запятую указываются имена переменных, которые необходимо вывести.

Количество символов % в строке формата должно совпадать с количеством переменных для вывода.

Тип каждого формата должен совпадать с типом переменной, которая будет выводиться на это место. Замещение форматов вывода значениями переменных происходит в порядке их следования.

# Оператор вывода

## Пример программы

```
1 #include <stdio.h>
2 int main ()
3 {
4     int a = 5;
5     float x = 2.78;
6     printf("a=%d\n", a);
7     printf("x=%f\n", x);
8     return 0;
9 }
```

## Результат работы программы

```
a=5
x=2.780000
```

# Оператор вывода

Тот же самый код может быть представлен с использованием одного вызова `printf`:

Пример программы

```
1 #include <stdio.h>
2 int main ()
3 {
4     int a = 5;
5     float x = 2.78;
6     printf("a=%d\nx=%f\n", a);
7     return 0;
8 }
```

Результат работы программы

```
a=5
x=2.780000
```

## Оператор вывода

## Табличный вывод

При указании формата можно явным образом указать общее количество знакомест и количество знакомест, занимаемых дробной частью:

### Пример программы

```
1 #include <stdio.h>
2 int main ()
3 {
4     float x = 2.78;
5     printf("x=%10.5f\n", x);
6     return 0;
7 }
```

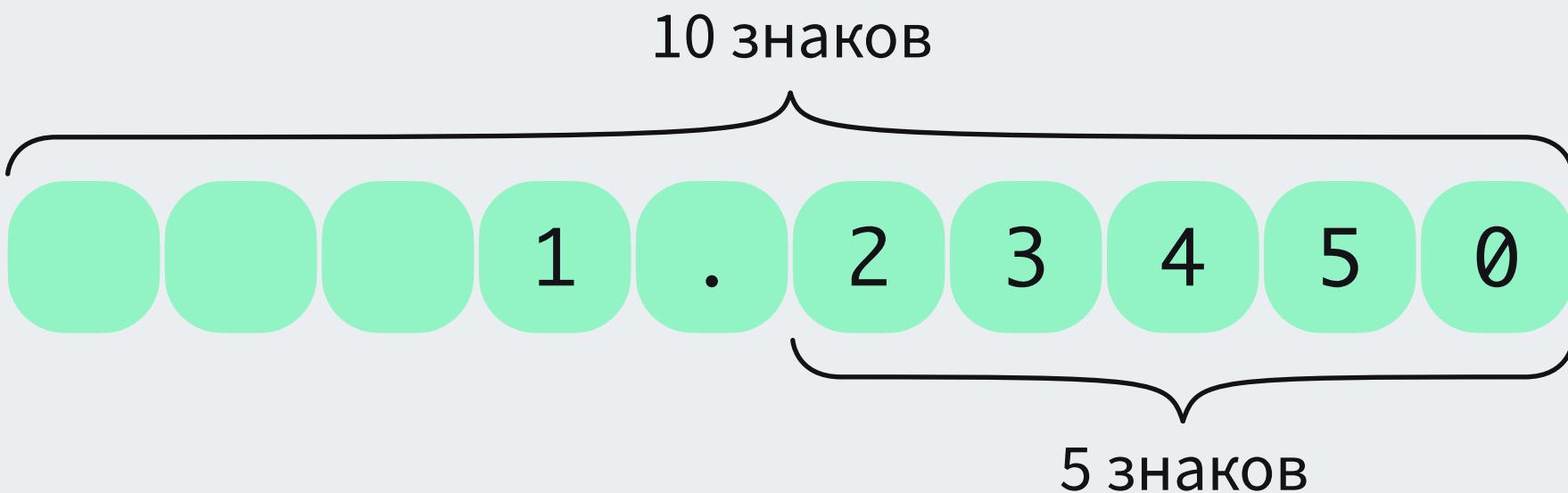
### Результат работы программы

```
x= 1.23450
```

# Оператор вывода

```
printf("x=%10.5f\n", x); x= 1.23450
```

В приведенном примере 10 – общее количество знакомест, отводимое под значение переменной; 5 – количество позиций после разделителя целой и дробной части (после десятичной точки). В указанном примере количество знакомест в выводимом числе меньше 10, поэтому свободные знакоместа слева от числа заполняются пробелами. Такой способ форматирования часто используется для построения таблиц.



# Оператор вывода

## Пример вывода текста

```
1 #include <stdio.h>
2 int main ()
3 {
4     printf("Привет \n");
5     printf("Вася\n");
6     return 0;
7 }
```

## Результат работы программы

Привет  
Вася

# Оператор вывода

## Пример вывода текста

```
1 #include <stdio.h>
2 int main ()
3 {
4     printf("Привет \nВася \n");
5     return 0;
6 }
```

## Результат работы программы

Привет  
Вася

# Оператор ввода

Функция форматированного ввода данных с клавиатуры `scanf( )` выполняет чтение данных, вводимых с клавиатуры, преобразует их во внутренний формат и передает вызывающей функции. При этом программист задает правила интерпретации входных данных с помощью спецификаций форматной строки. Общая форма записи функции `scanf( )`:

```
scanf("СтрокаФорматов", адрес1, адрес2, ...);
```

Строка форматов аналогична функции `printf( )`. Для формирования адреса переменной используется символ амперсанд `&`:

```
адрес = &объект
```



Строка форматов и список аргументов для функции обязательны.

# Оператор ввода

## Пример программы

```
1 #include <stdio.h>
2 int main ( )
3 {
4     float y;
5     printf("Введите у: "); // выводим сообщение
6     scanf("%f", &y);      // вводим значения переменной у
7     printf("Значение переменной у=%f", y); // выводим значение переменной у
8     return 0;
9 }
```

## Результат работы программы

Введите у: 5  
Значение переменной у=5.000000

# Оператор ввода

## Пример

Пользователь вводит с клавиатуры 2 целых числа, программа должна вывести их сумму

```
1 #include <stdio.h>
2 int main ( )
3 {
4     int a,b;
5     printf("Введите первое число: "); // выводим сообщение
6     scanf("%d", &a);      // вводим значения переменной a
7     printf("Введите второе число: ", y); // выводим сообщение
8     scanf("%d", &b);      // вводим значения переменной b
9     printf("Сумма чисел=%d", a+b); // выводим значение суммы
10    return 0;
11 }
```

Пример работы  
программы

Введите первое число: 1  
Введите второе число: 2  
Сумма чисел=3

# Среда разработки

**Интегрированная среда разработки, ИСР (англ. Integrated development environment – IDE)** – комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Среда разработки включает в себя:

✓ текстовый редактор

✓ транслятор (компилятор и/или интерпретатор)

✓ средства автоматизации сборки

✓ отладчик

# Среда разработки

Популярные среды разработки на C++:



Microsoft Visual Studio



Eclipse CDT



Sublime Text



NetBeans



Qt Creator



CLion



CodeLite



Code::Blocks

# Среды разработки

## Интегрированная среда разработки Visual Studio (IDE)

Помимо изменения кода, Visual Studio IDE объединяет графические конструкторы, компиляторы, средства завершения кода, системы управления версиями, расширения и многие другие функции в одном месте.



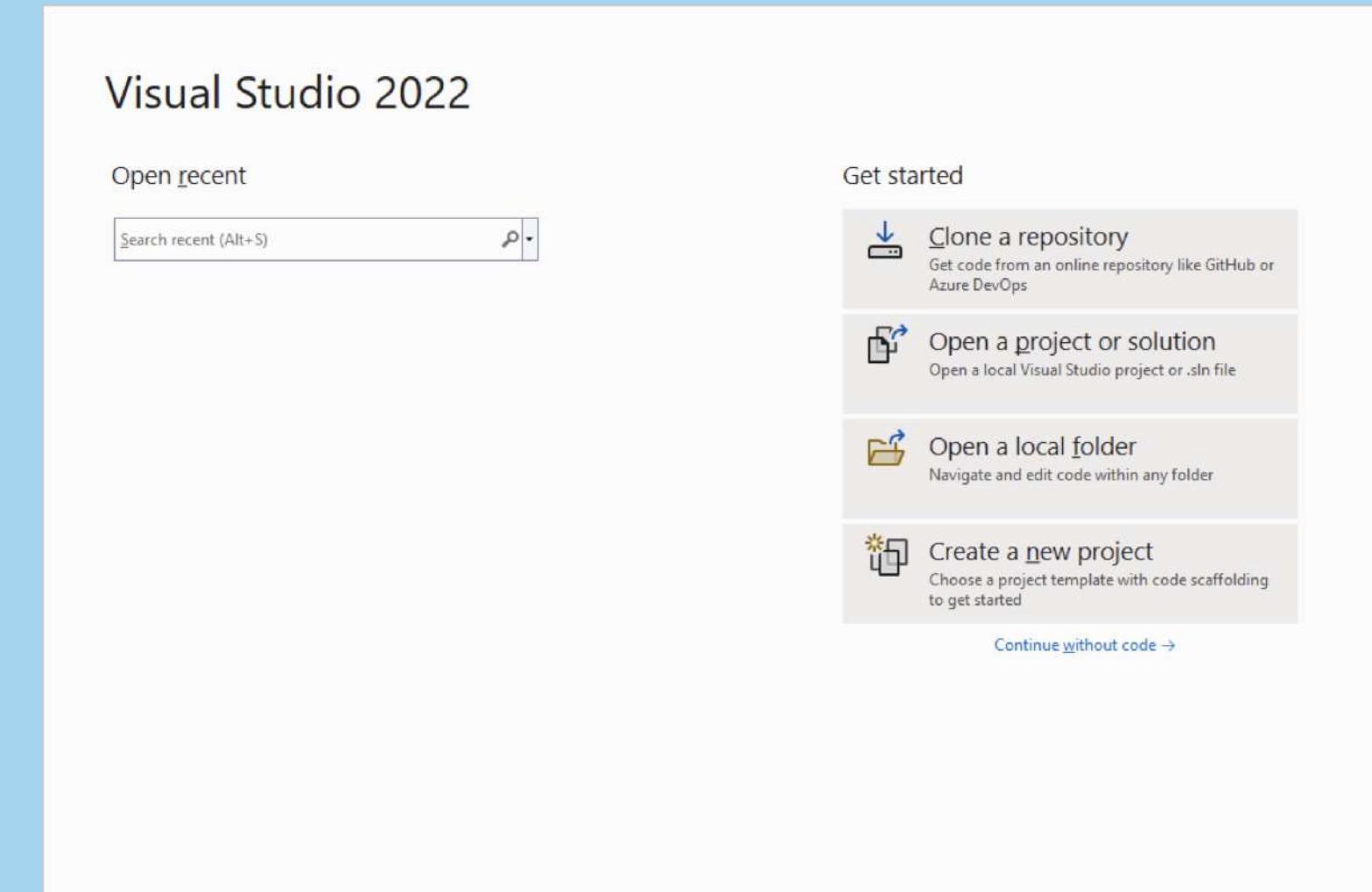
# Среда разработки

- 1 Установите и запустите на своем компьютере Visual Studio с рабочей нагрузкой Разработка классических приложений на C++.
- 2 Запустите Visual Studio, вы увидите диалоговое окно запуска. Выберите «Создать проект», чтобы приступить к работе.



Или в строке меню Visual Studio последовательно выберите Файл>Создать>Проект. Откроется окно «Создание проекта».

Visual Studio использует проекты, чтобы упорядочить код для приложения, и решения, чтобы упорядочить проекты. Проект содержит все параметры, конфигурации и правила, используемые для сборки приложения. Кроме того, он управляет связью между всеми файлами проекта и любыми внешними файлами. Чтобы создать приложение, сначала создайте проект и решение.



# Среда разработки

3

В списке шаблонов проектов выберите «Консольное приложение» и нажмите «Далее».

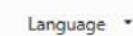
**ВАЖНО**



Убедитесь, что вы выбрали версию C++ для шаблона Консольное приложение. Этот шаблон содержит теги C++, Windows и Консоль, а в углу значка есть «++».

## Create a new project

Search for project templates



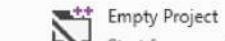
Language

Platform

Project type

### Recent project templates

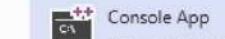
A list of your recently accessed templates will be displayed here.



Empty Project

Start from scratch with C++ for Windows. Provides no starting files.

C++ Windows Console



Console App

Run code in a Windows terminal. Prints "Hello World" by default.

C++ Windows Console



Windows Desktop Wizard

Create your own Windows app using a wizard.

C++ Windows Desktop Console Library



Windows Desktop Application

A project for an application with a graphical user interface that runs on Windows.

C++ Windows Desktop

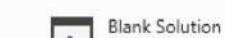


Shared Items Project

A Shared Items project is used for sharing files between multiple projects.

C++ Windows Android iOS Linux Desktop Console

Library UWP Games Mobile



Blank Solution

Create an empty solution containing no projects

Other

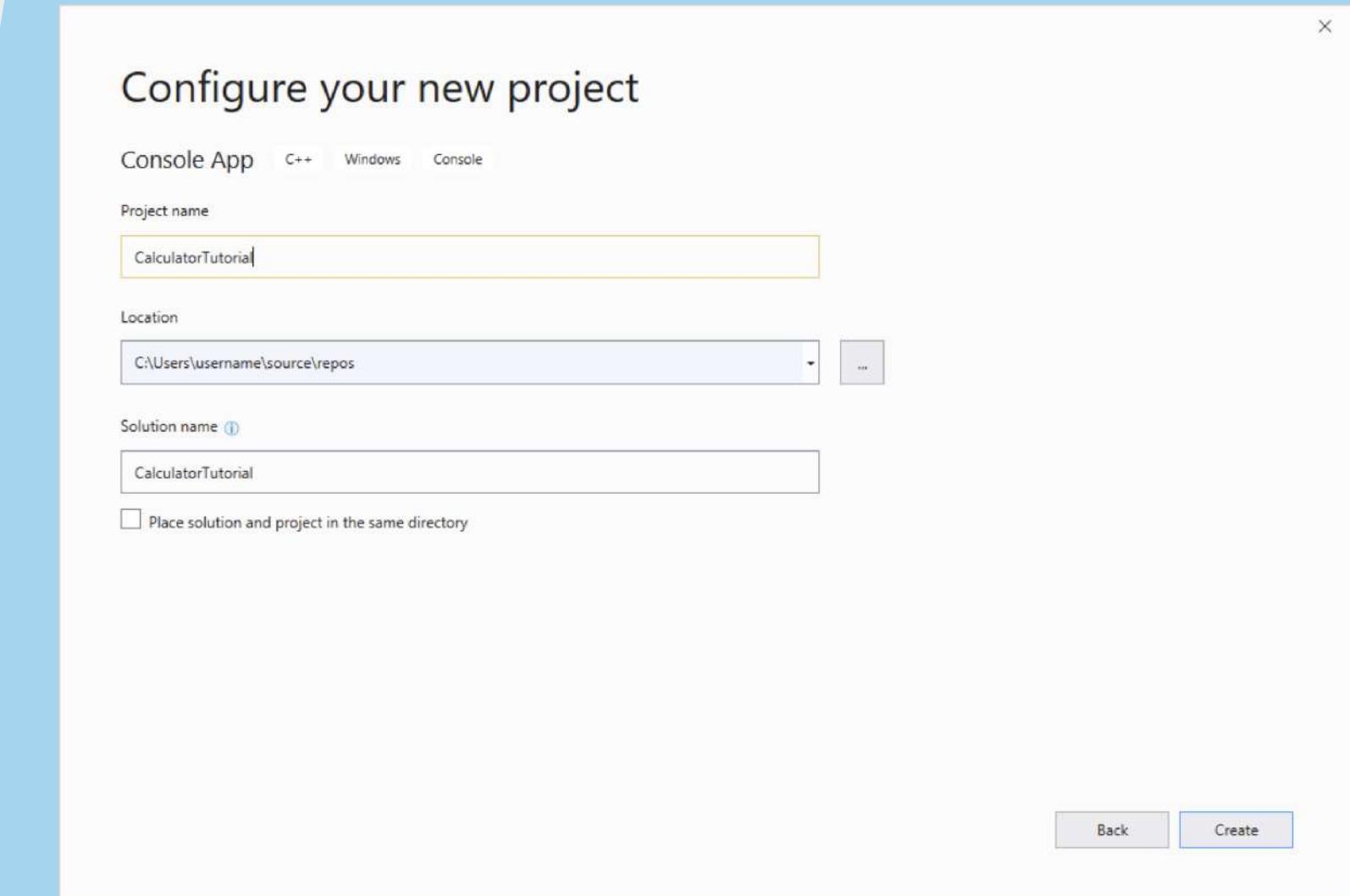
Back

Next

# Среда разработки

- 4 В диалоговом окне «Настроить новый проект» выберите поле ввода «Имя проекта», назовите новый проект и нажмите «Создать».

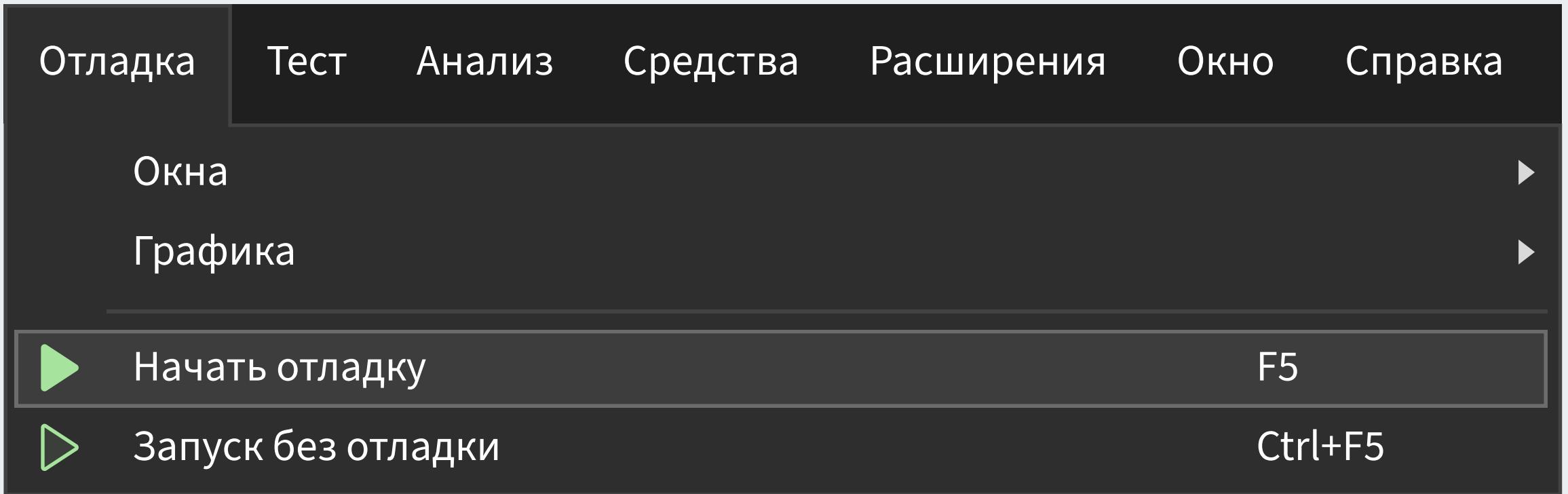
Будет создано пустое консольное приложение Windows на C++. Консольные приложения используют окно консоли Windows для отображения выходных данных и приема данных, вводимых пользователем. В Visual Studio откроется окно редактора с созданным кодом.



# Запуск программы

5

Чтобы запустить этот код, в строке меню выберите «Отладка» и «Начать отладку» или нажмите на клавиатуре кнопку «F5».



# Online компиляторы

[online-cpp.com](http://online-cpp.com)

[onlinegdb.com/  
online\\_c\\_compiler](http://onlinegdb.com/online_c_compiler)

[replit.com/  
languages/cpp](http://replit.com/languages/cpp)