

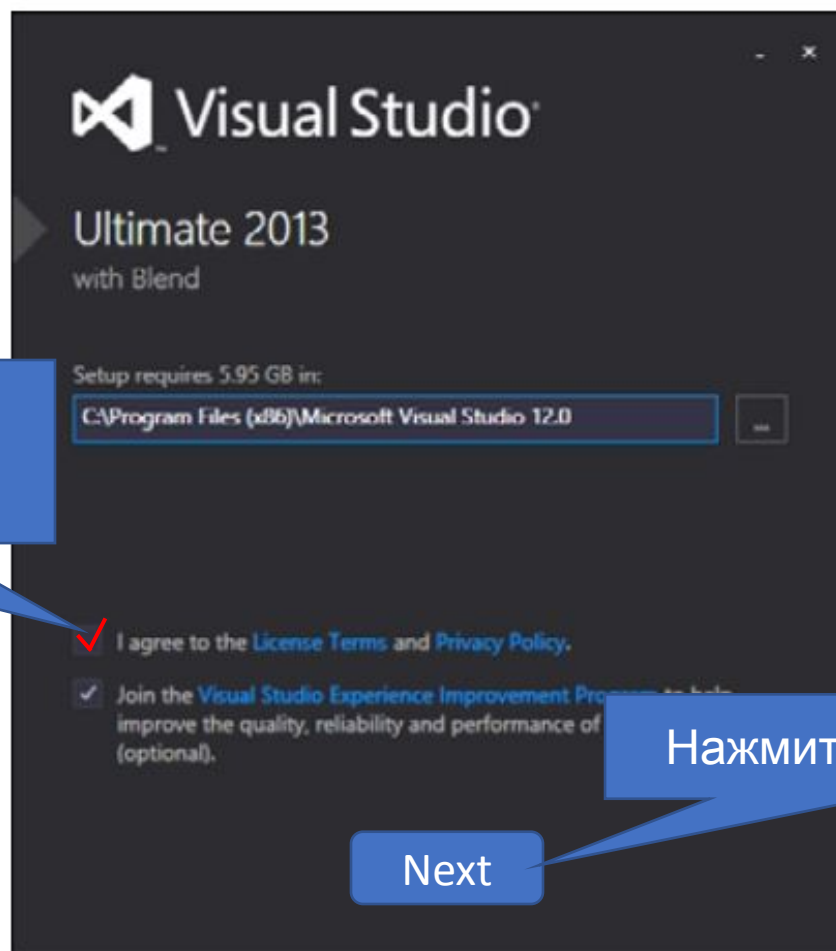
Введение в программирование

Три версии Visual Studio

- **Visual Studio Community 2015** — бесплатная версия доступная для скачивания всем разработчикам. Она имеет богатые возможности. Её можно использовать в любых целях, как коммерческих, так и некоммерческих. Это хороший выбор для того, чтобы начать знакомство с данным продуктом.
- **Visual Studio Professional 2015** — платная версия продукта. Содержит большое количество улучшений по сравнению с бесплатной версией.
- **Visual Studio Enterprise 2015** — полная платная версия продукта. В ней есть мощная функциональность, которая необходима при разработке самых сложных корпоративных приложений.

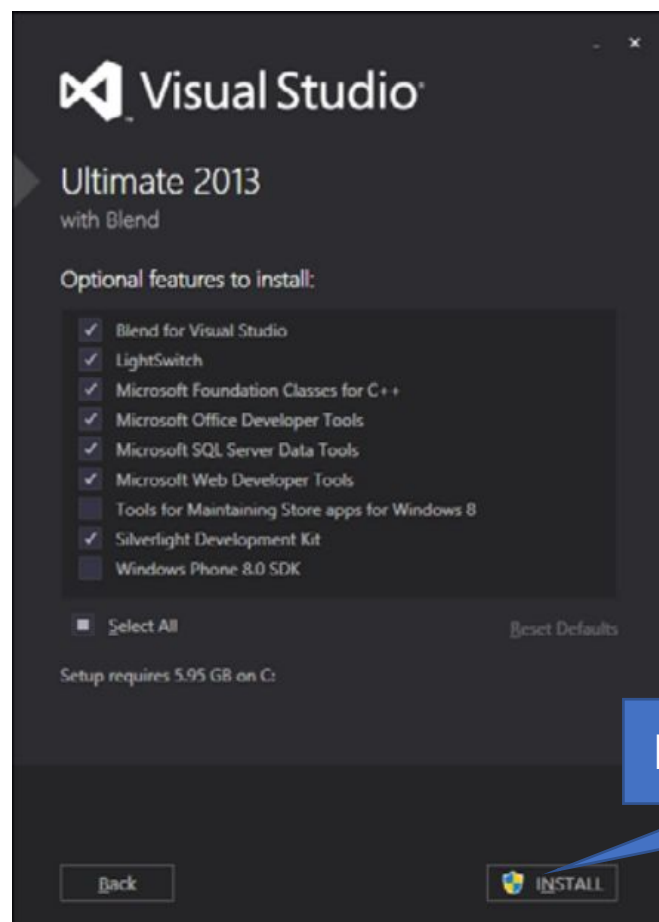
Установка Visual Studio

Отметьте пункт «I agree to the License Terms and Privacy Policy» для начала установки.



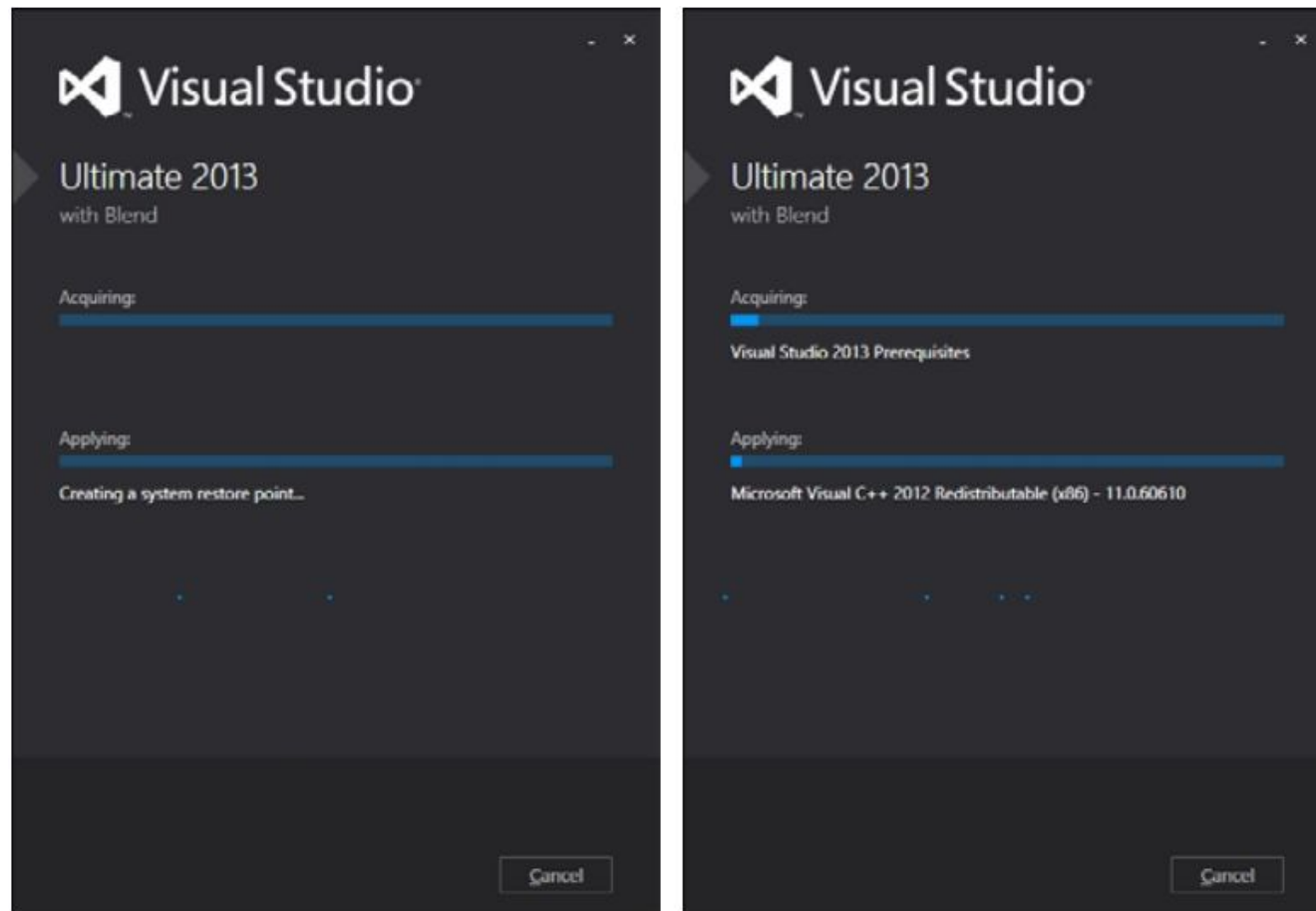
Нажмите кнопку «Next»

Установка Visual Studio



Нажмите кнопку «INSTALL»

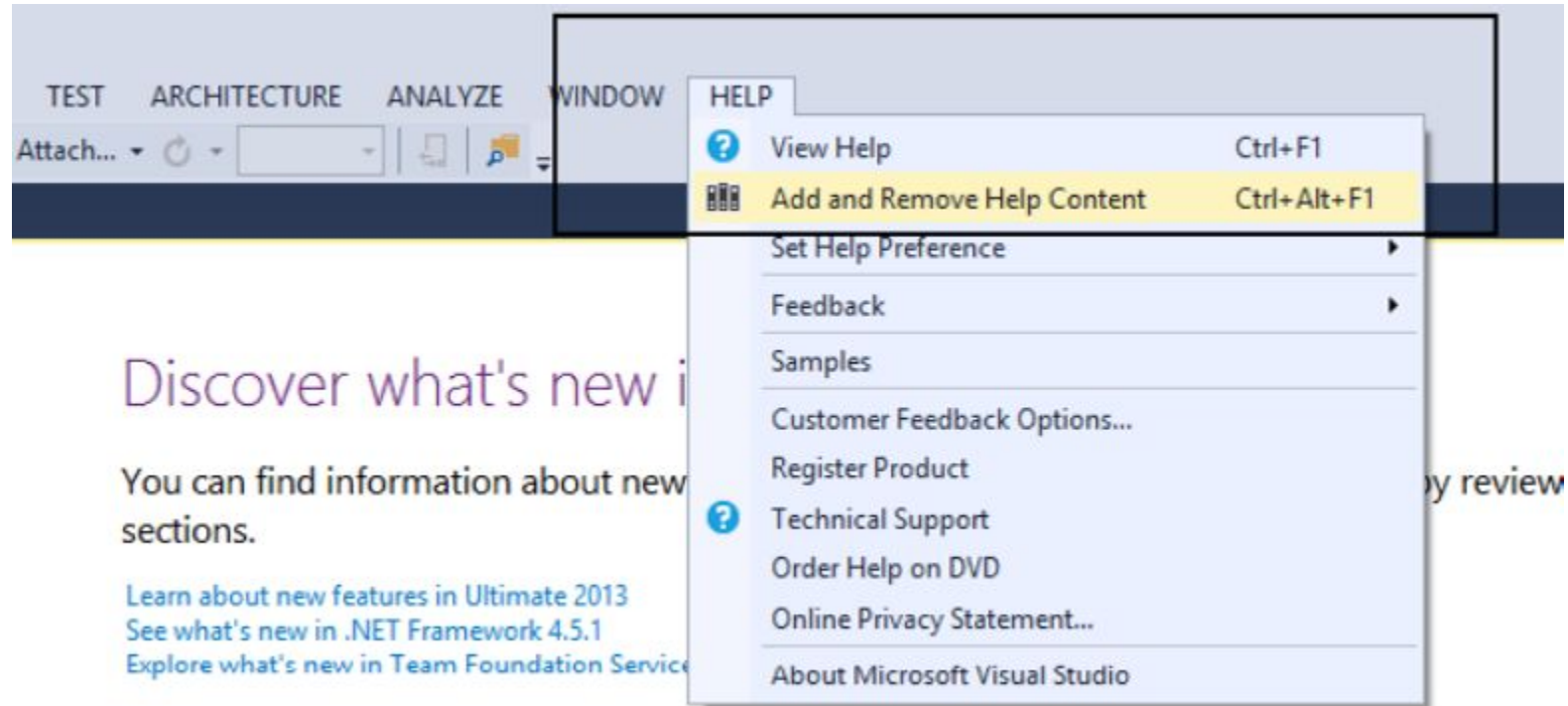
Установка Visual Studio



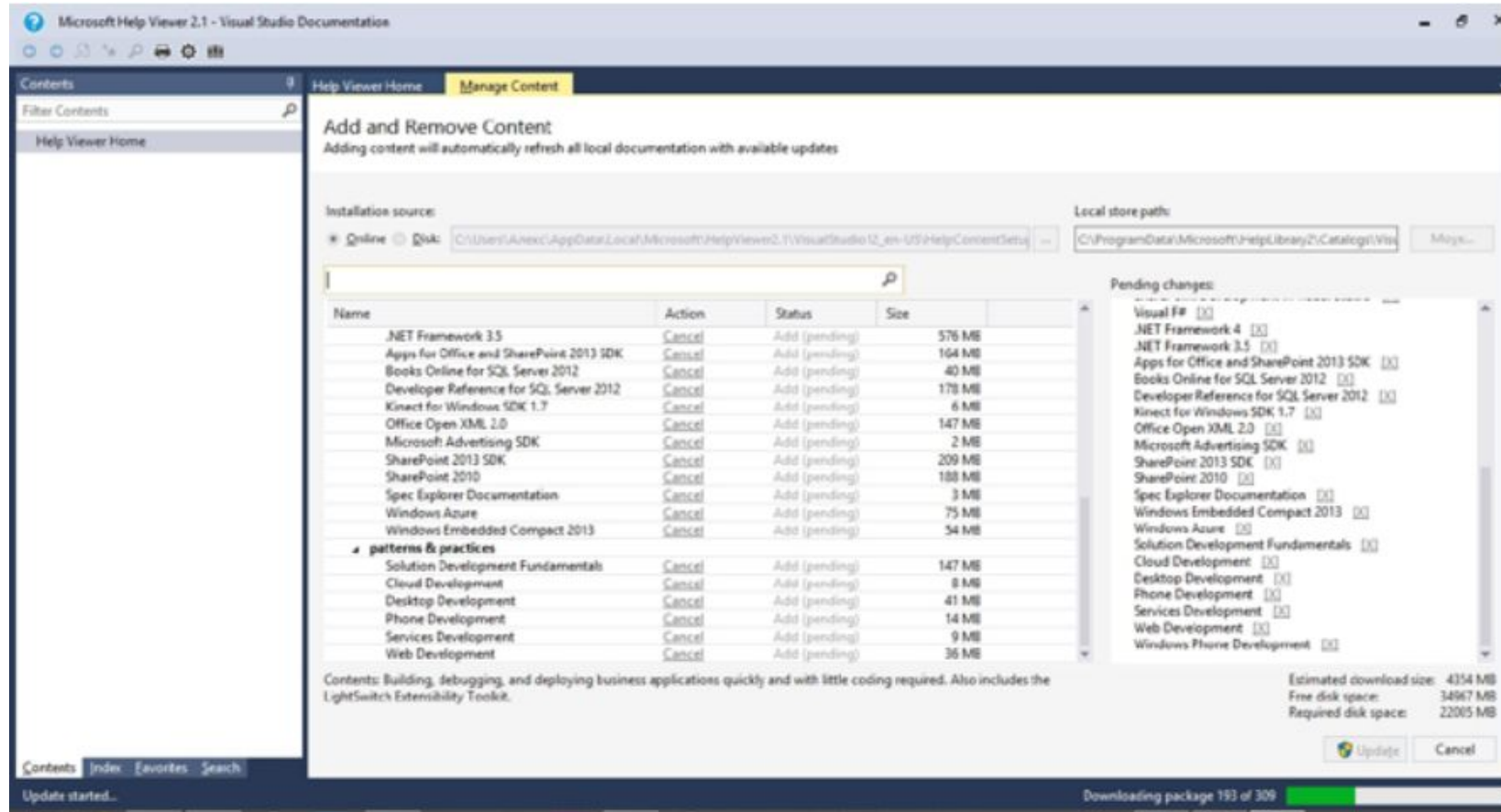
Получение справок

- При обучении и разработке приложений вам очень пригодится справочная информация (MSDN) от Microsoft.
- Для того, чтобы установить локальную копию справки нужно выбрать пункт меню Help->Add and Remove Help Content или же нажать Ctrl+Alt+F1.

Получение справок



Получение справок



Вы увидите окно с выбором справочного материала. Рекомендуем везде нажать ссылку «Add».

После выбора материалов нажмите кнопку «Update» для их локальной установки.

Программирование

На языке Си

Вид программы в машинных кодах

```
00000000 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
00000010 02 00 3e 00 01 00 00 00 00 04 40 00 00 00 00 00
00000020 40 00 00 00 00 00 00 00 70 11 00 00 00 00 00 00
00000030 00 00 00 00 40 00 38 00 09 00 40 00 1e 00 1b 00
00000040 06 00 00 00 05 00 00 00 40 00 00 00 00 00 00 00
00000050 40 00 40 00 00 00 00 00 40 00 40 00 00 00 00 00
00000060 f8 01 00 00 00 00 00 00 f8 01 00 00 00 00 00 00
00000070 08 00 00 00 00 00 00 00 03 00 00 00 04 00 00 00
00000080 38 02 00 00 00 00 00 00 38 02 40 00 00 00 00 00
00000090 38 02 40 00 00 00 00 00 1c 00 00 00 00 00 00 00
000000a0 1c 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00
000000b0 01 00 00 00 05 00 00 00 00 00 00 00 00 00 00 00
000000c0 00 00 40 00 00 00 00 00 00 00 40 00 00 00 00 00
000000d0 ac 06 00 00 00 00 00 00 ac 06 00 00 00 00 00 00
000000e0 00 00 20 00 00 00 00 00 01 00 00 00 06 00 00 00
000000f0 10 0e 00 00 00 00 00 00 10 0e 60 00 00 00 00 00
00000100 10 0e 60 00 00 00 00 00 28 02 00 00 00 00 00 00
00000110 30 02 00 00 00 00 00 00 00 00 20 00 00 00 00 00
00000120 02 00 00 00 06 00 00 00 28 0e 00 00 00 00 00 00
00000130 28 0e 60 00 00 00 00 00 28 0e 60 00 00 00 00 00
00000140 d0 01 00 00 00 00 00 00 d0 01 00 00 00 00 00 00
00000150 08 00 00 00 00 00 00 00 04 00 00 00 04 00 00 00
00000160 54 02 00 00 00 00 00 00 54 02 40 00 00 00 00 00
00000170 54 02 40 00 00 00 00 00 44 00 00 00 00 00 00 00
00000180 44 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00
00000190 50 e5 74 64 04 00 00 00 a4 05 00 00 00 00 00 00
000001a0 a4 05 40 00 00 00 00 00 a4 05 40 00 00 00 00 00
000001b0 34 00 00 00 00 00 00 00 34 00 00 00 00 00 00 00
000001c0 04 00 00 00 00 00 00 00 51 e5 74 64 06 00 00 00
000001d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

- Все компьютеры не понимают человеческий язык, они понимают исключительно свой, **машинный язык**.
- Человек может разобраться в машинном языке, но это сложно, утомительно и чревато большим количеством ошибок.

Вид программы на ассемблере

```
intel_processor:
    mov     intel_proc, 1
    mov     [intel-1], ' '           ; add a space so the Genuine Intel
                                    ; message prints out.

cpuid_data:
    mov     eax, 1
    cpuid
    mov     saved_cpuid, eax        ; save for future use
    and     eax, 0F00H             ; mask everything but family
    shr     eax, 8
    mov     cpu_type, al           ; set cpu_type with family

    mov     eax, saved_cpuid        ; restore data
    mov     stepping, al
    and     stepping, 0FH         ; isolate stepping info

    mov     eax, saved_cpuid
    mov     themodel, al
    and     themodel, 0F0H        ; isolate model info
    shr     themodel, 4
```

- Для большего удобства людей, были разработаны **ассемблерные языки**. Ассемблерный код несколько более понятен, чем машинный код.
- Но всё же и с ассемблерными языками работать человеку очень сложно. Кроме того, ассемблерные языки сильно привязаны к платформе.

Языки высокого уровня.

- Машинные коды и **ассемблерные** языки – это языки низкого уровня, сильно связанные к конкретным процессором, не совместимые друг с другом и малопонятны человеку.
- Для большей платформенной независимости и большего удобства для человека разработаны языки **высокого уровня**.
- Языки высокого уровня более универсальны и их синтаксис приближен к естественному человеческому языку.

Два вида языков программирования

- ИНТЕРПРЕТИРУЕМЫЕ

- КОМПИЛИРУЕМЫЕ

- Деление языков на эти две группы условное, так как для любого языка может быть создан **компилятор** или **интерпретатор**.

Два вида языков программирования

- **Компилируемый язык программирования** — это язык, исходный код которого преобразуется компилятором в машинный код и записывается в файл с особым заголовком и/или расширением для последующей идентификации этого файла, как исполняемого операционной системой.
- **Интерпретируемый язык программирования** — это язык, в котором исходный код не преобразовывается в машинный код для непосредственного выполнения процессором, а исполняется с помощью специальной программы - интерпретатора.

Два вида языков программирования

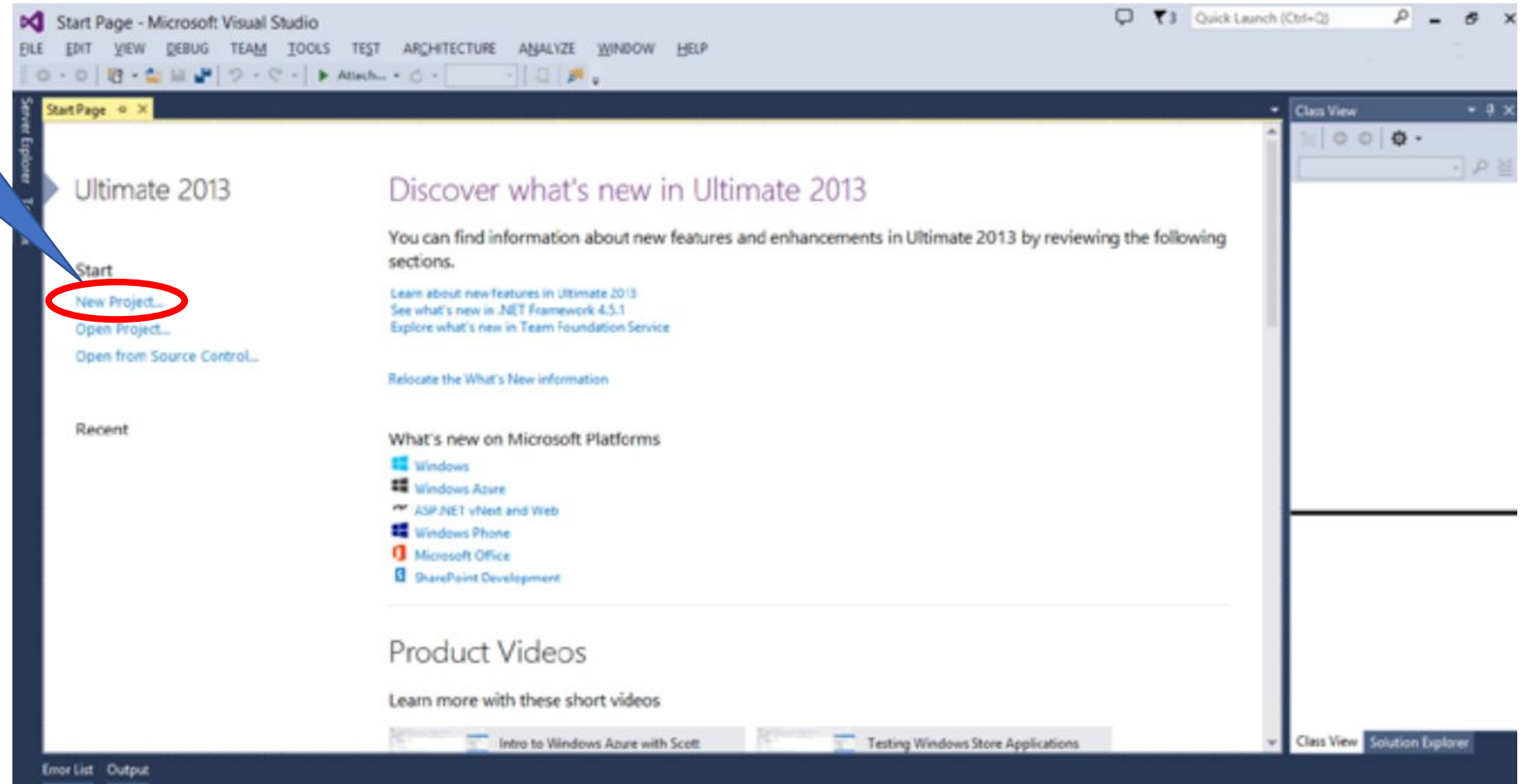
- Язык **Си** является **КОМПИЛИРУЕМЫМ** языком программирования.

Два вида языков программирования

- Язык **Си** является **КОМПИЛИРУЕМЫМ** языком программирования.

Первая программа

Нажмите
New Project



Первая программа

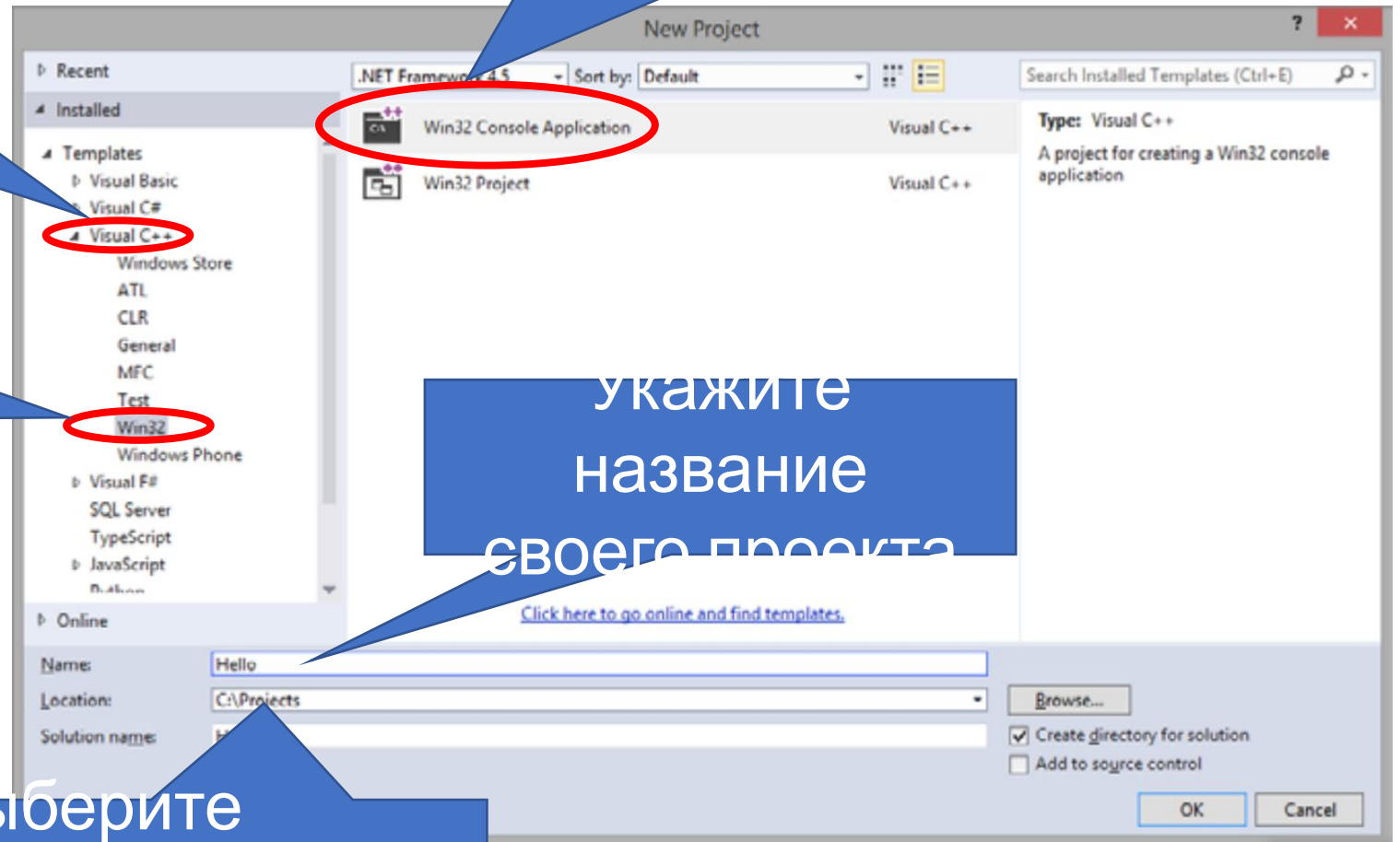
Выберите
Visual C++

Выберите
Win32

Выберите
Win32 Console Application

укажите
название
своего проекта

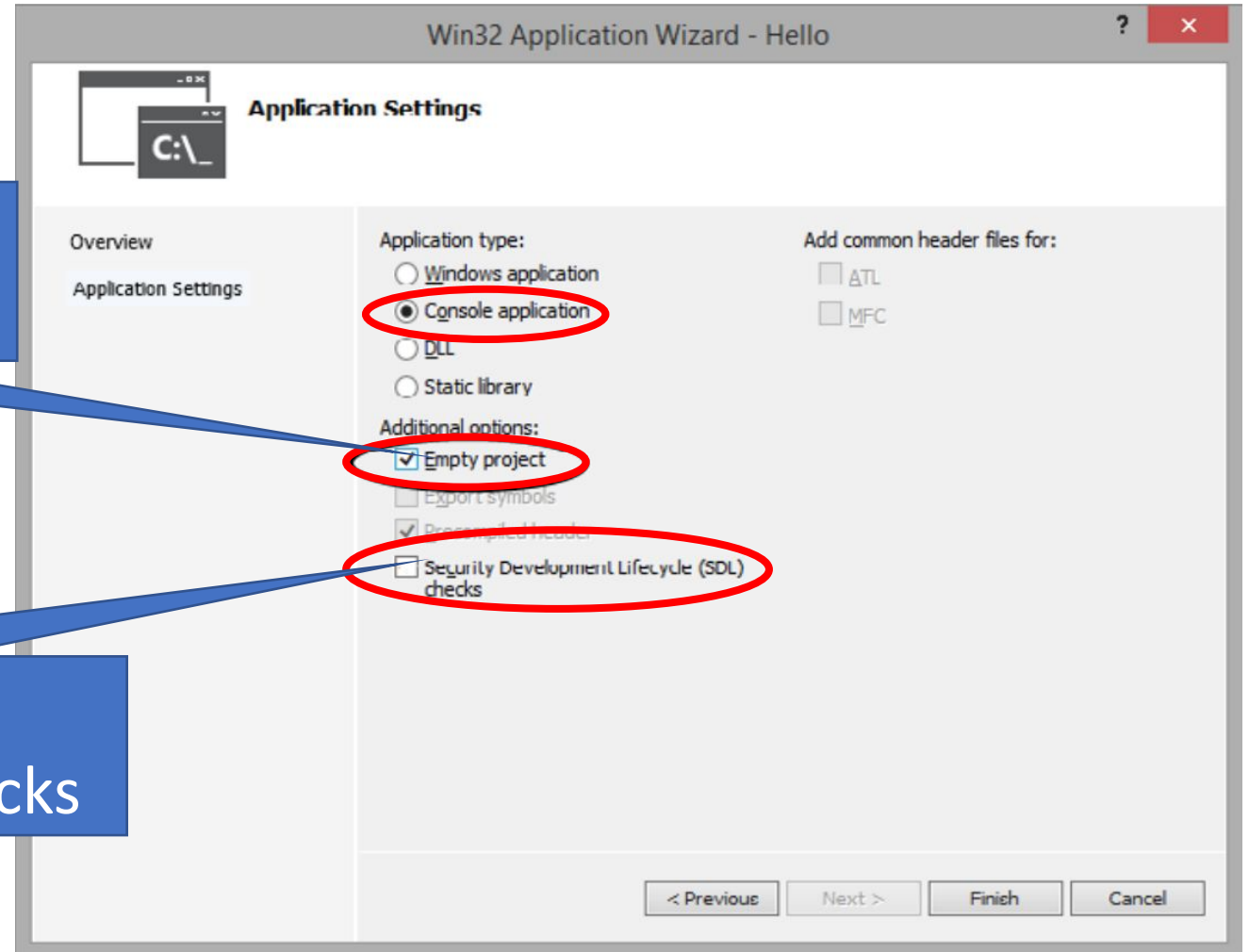
Выберите
папку для своего
проекта



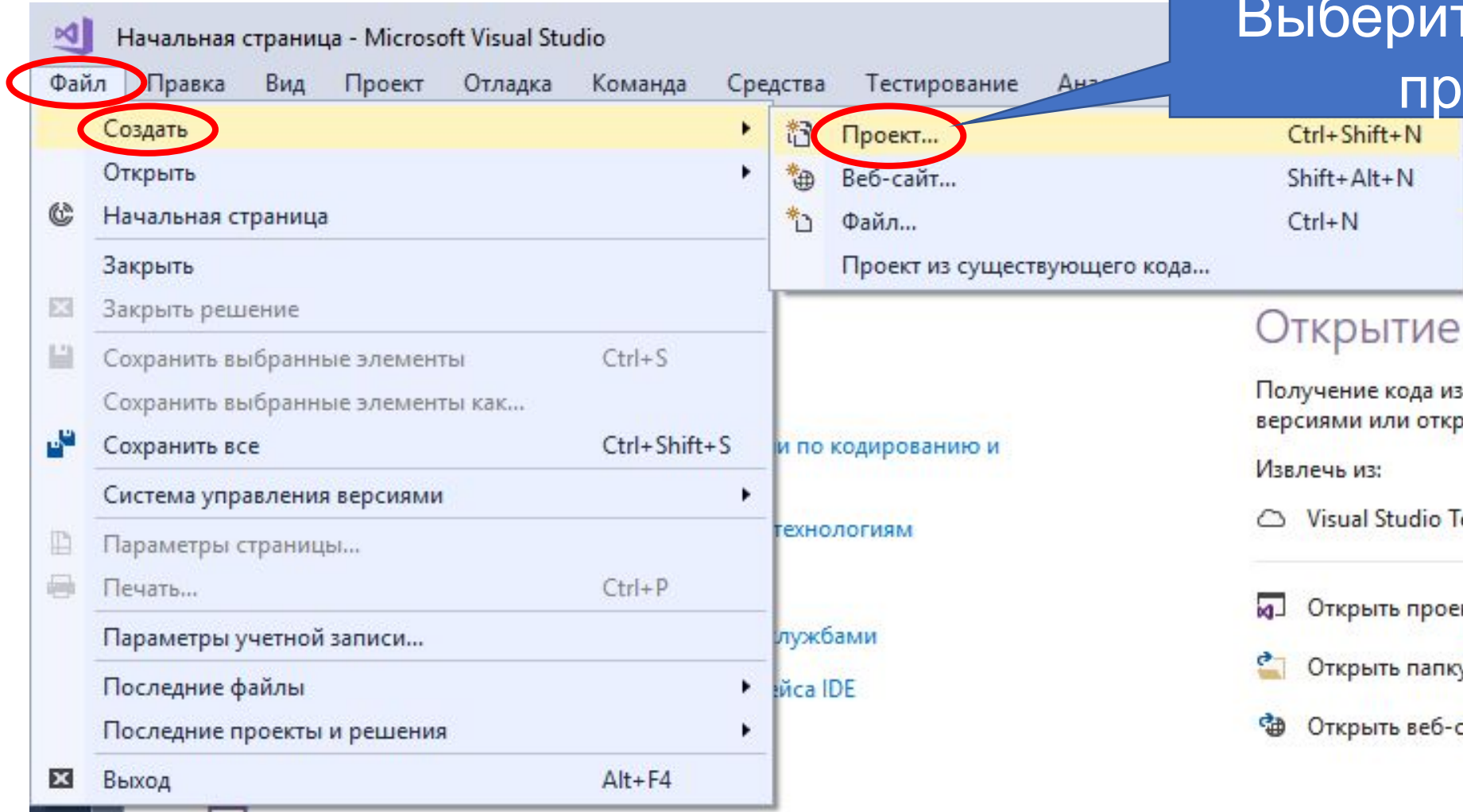
Первая программа

укажите что создаете пустой проект (установите «галочку»)

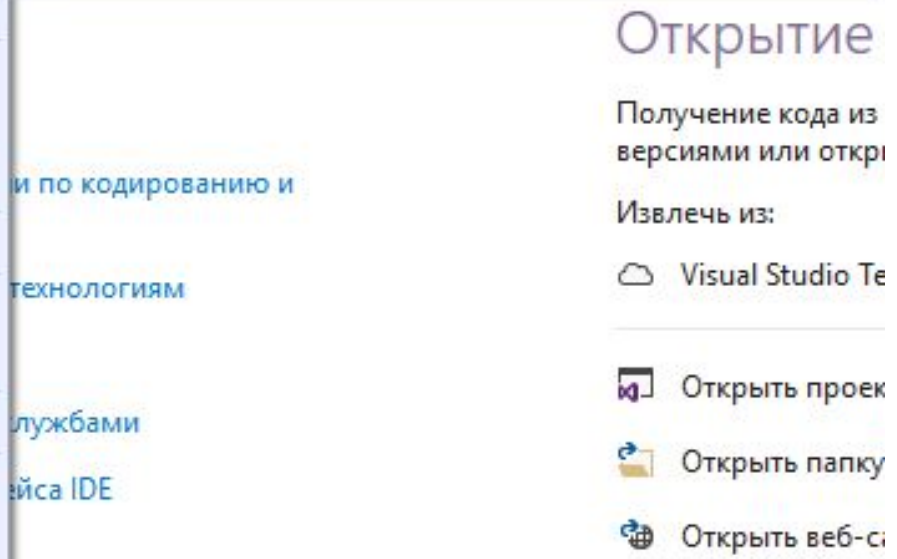
СНИМИТЕ галочку Security Development Lifecycle (SDL) checks



Первая программа в Visual Studio 2017



Выберите создание проекта

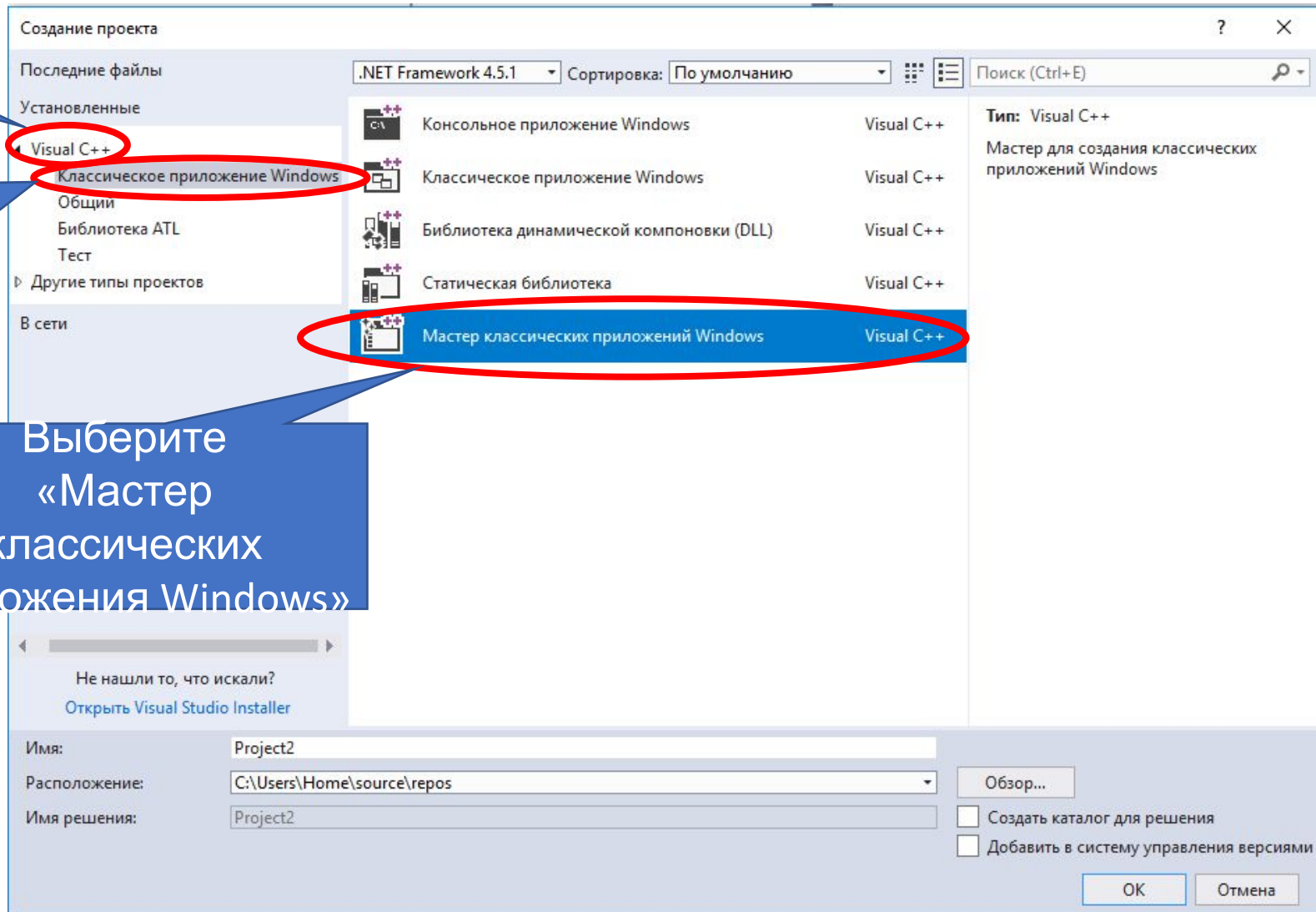


Первая программа в Visual Studio 2017

Выберите
«Visual C++»

Выберите
«Классическое приложение Windows»

Выберите
«Мастер классических приложений Windows»



Первая программа в Visual Studio 2017

The image shows a screenshot of the 'New Project' dialog box in Visual Studio 2017. The dialog is titled 'классического приложения Windc' and has a close button 'x' in the top right corner. It contains the following elements:

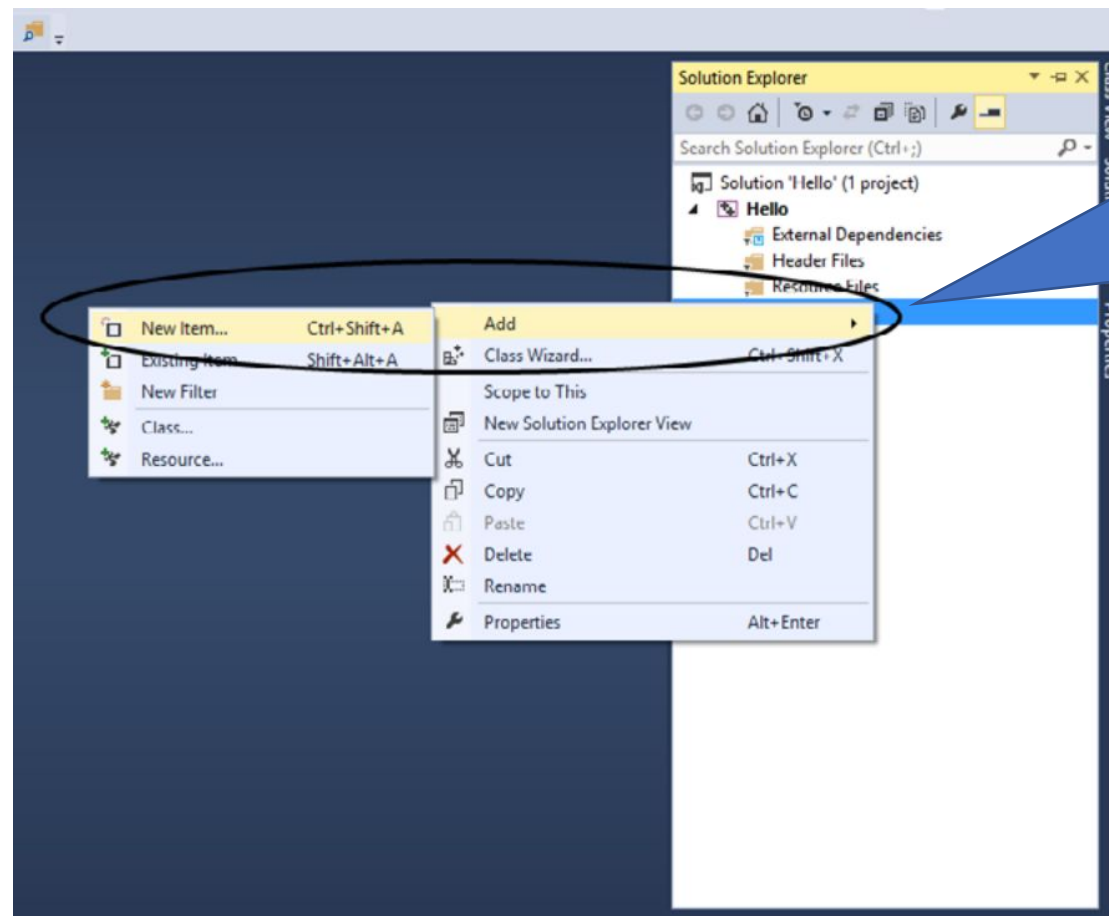
- Тип приложения:** A dropdown menu with 'Консольное приложение (.c' selected. This option is circled in red.
- Дополнительные параметры:** A list of checkboxes:
 - Пустой проект (This checkbox is circled in red.)
 - Экспорт символов
 - Предкомпилированный заголовок
 - Проверки в жизненном цикле разработки защищенных приложений (SDL)
- Добавить общие заголовки для:** Two checkboxes:
 - ATL
 - MFC
- Buttons:** 'OK' and 'Отмена' (Cancel) buttons at the bottom right.

Three blue callout boxes provide instructions:

- Top callout: 'Выберите «Консольное приложение»' (Select 'Console application').
- Middle callout: 'Установите галочку «Пустой проект»' (Set the 'Empty project' checkbox).
- Bottom callout: 'Снимите галочку «Пустой проект»' (Remove the 'Empty project' checkbox).

Добавляем чистый файл

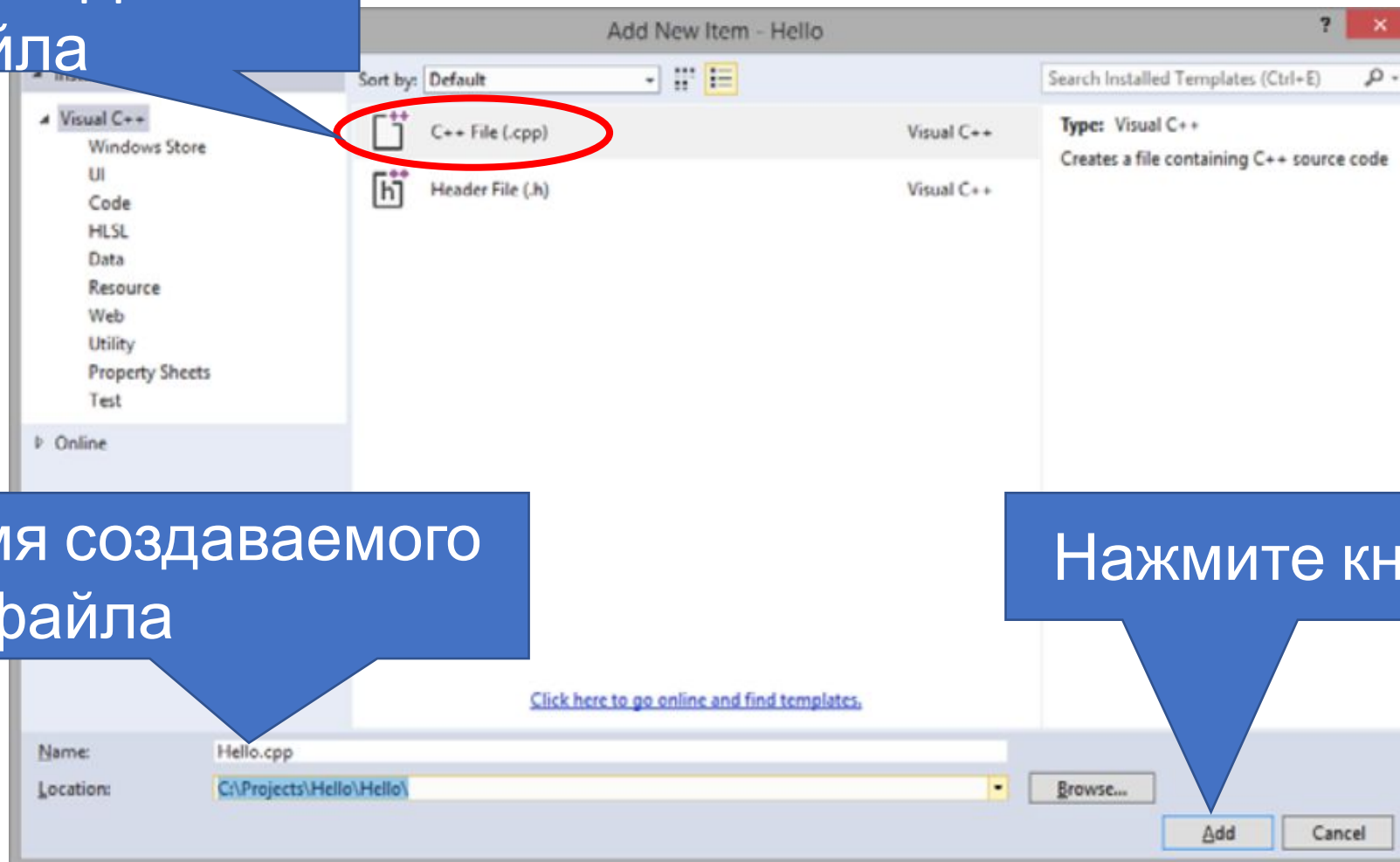
Для вызова окна Solution Explorer можно также использовать клавиатурную комбинацию **Ctrl+Alt+L**



Необходимо щелкнуть правой кнопкой на папке под названием **Source Files**

Добавляем чистый файл

Выберите .cpp тип файла



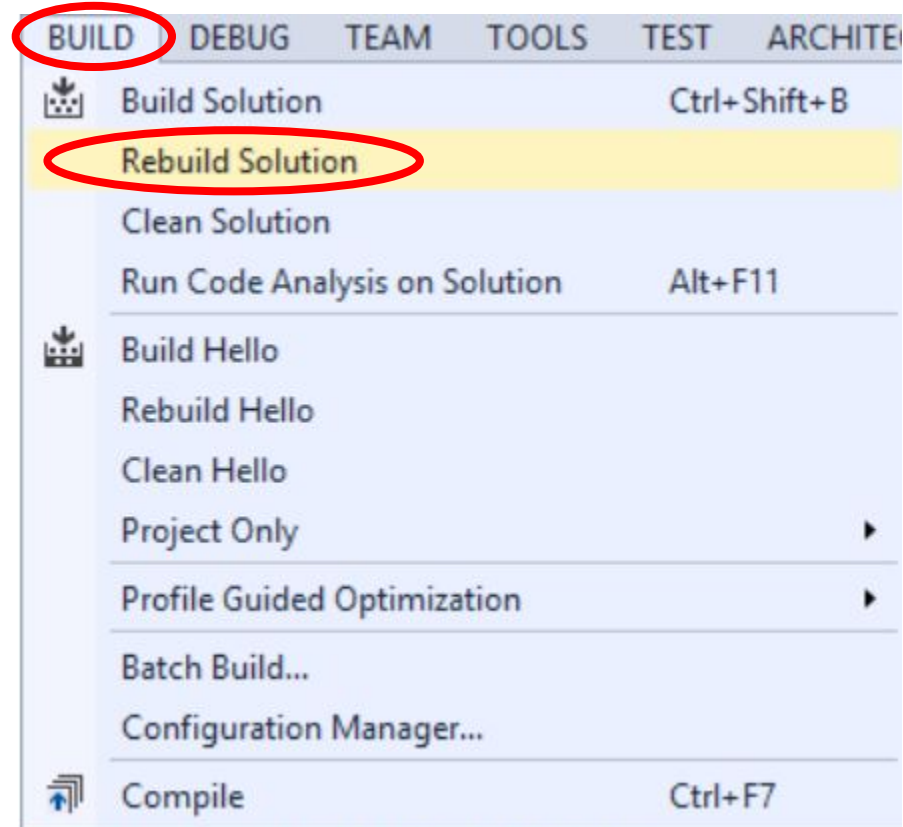
Введите имя создаваемого файла

Нажмите кнопку Add

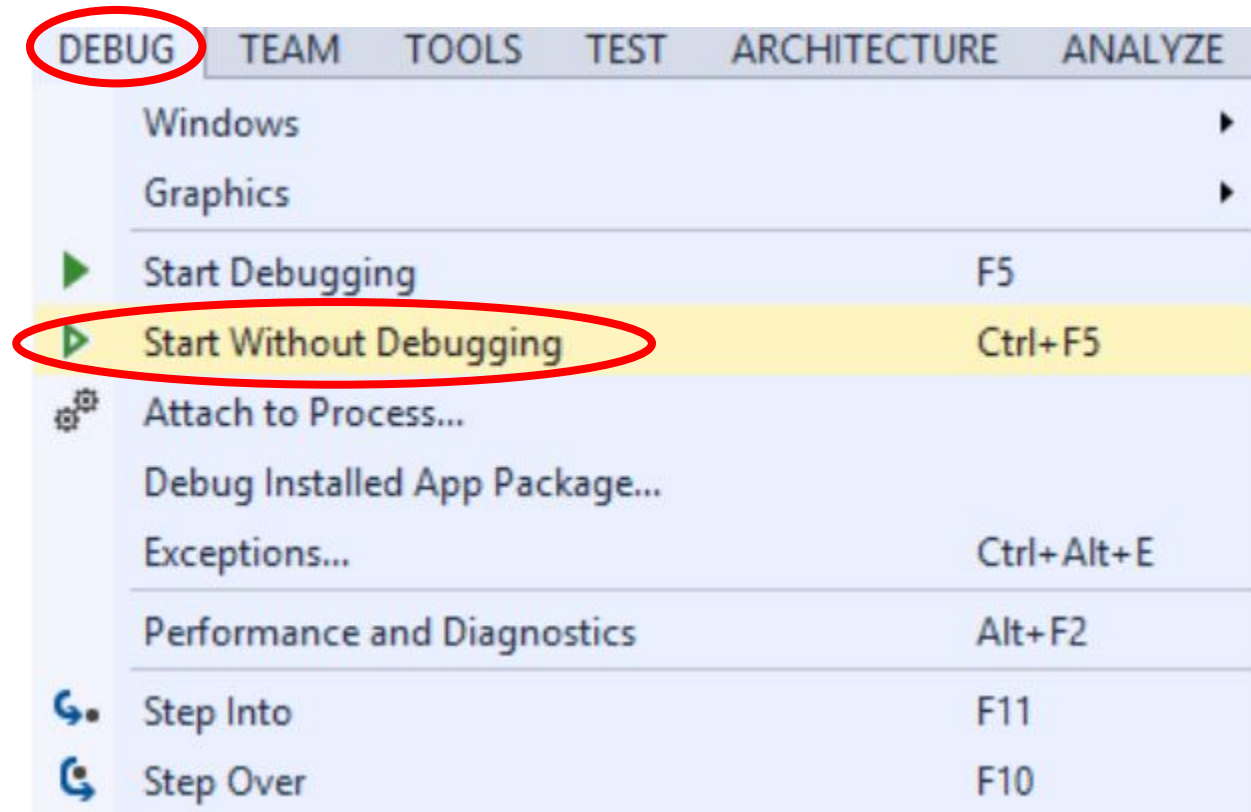
Первая программа

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello, World!\n";
    return 0;
}
```

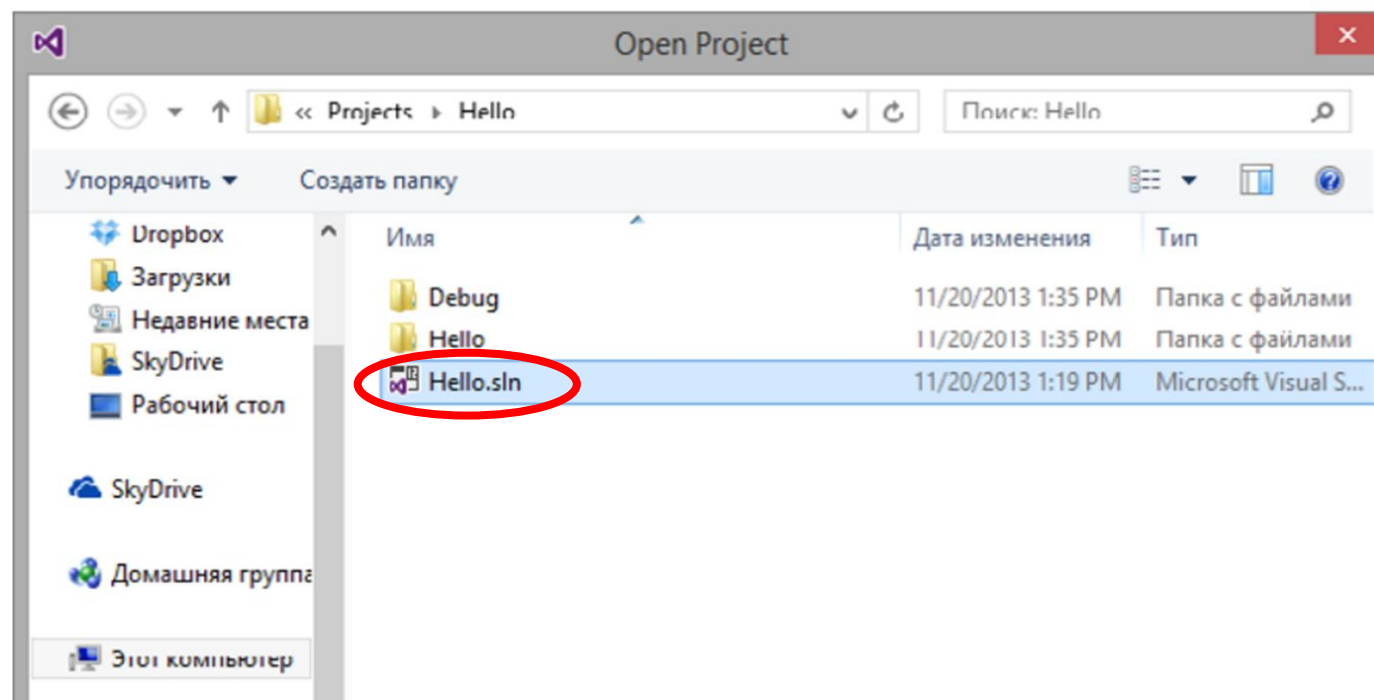
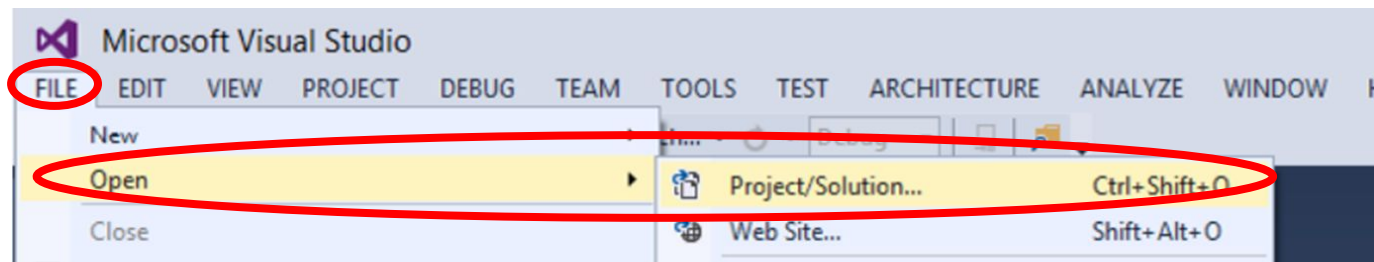
Первая программа



Первая программа



Открытие сохранённой программы



Вывод данных

- Вывод данных осуществляется с помощью команды

`cout<<`

- В заголовке программы должна присутствовать строка:

`#include <iostream>`

Вывод данных

- Перед использованием команды необходимо подключить пространство имен, к которому принадлежит команда `cout`

using namespace std;

Вывод данных

- Строка, которую мы хотим вывести на экран используя `cout<<` обязательно должна быть заключённой в кавычки:

```
cout << "здесь пишем то, что хотим";
```

В языке Си всё делается в функциях

- Функция `main()` – это главная функция, с её выполнения начинается любая программа написанная на языке Си;
- Фигурными скобками обозначается логический блок кода программы.

Вывод данных

- Для оформления вывода строки используют специальные управляющие символы, представляющие собой комбинацию символа \ и символа, определяющего действие, которое необходимо произвести над строкой.
- Эти управляющие символы называются Escape-последовательностями.

Вывод данных

Escape-последовательность	Представляет
<code>\b</code>	Удаление предыдущего символа
<code>\n</code>	Новая строка
<code>\t</code>	Горизонтальная табуляция
<code>\'</code>	Одиночная кавычка
<code>\"</code>	Двойная кавычка
<code>\\</code>	Обратная косая черта
<code>\a</code>	Звуковой сигнал

Пример использования cout<<

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"\t\t\tEscape Sequences\n";
    cout<<"\n";
    cout<<"\t\t\b"<<"\tBackspace"<<endl;
    cout<<"\n";
    cout<<"\t\t\n"<<"\tNew line\n";
    cout<<endl;
```

Пример использования cout<<

```
cout<<"\t\t\t"<<"\tHorizontal tab\n";  
cout<<"\n";  
cout<<"\t\t\t\t\t"<<"\tBackslash \\\n";  
cout<<"\n";  
cout<<"\t\t\t""<<"\tDouble quotation mark \"\n";  
cout<<"\n";  
cout<<"\t\t\t'"<<"\tSingle quotation mark '\n";  
cout<<"\n";  
return 0;  
}
```

Смена локали

- `setlocale(LC_ALL, "RUSSIAN");` // Смена локали
- `setlocale(0, "");` // Смена локали

Спасибо за внимание.