

Практическая работа 1. Отладка кода в Visual Studio

В практической работе рассматриваются возможности отладчика Visual Studio. *Отладка приложения* обычно означает запуск и выполнение приложения с подключенным отладчиком. При этом в отладчике доступно множество способов наблюдения за выполнением кода. Вы можете пошагово перемещаться по коду и просматривать значения, хранящиеся в переменных, задавать контрольные значения для переменных, чтобы отслеживать изменение значений, изучать путь выполнения кода, просматривать выполнение ветви кода и т. д. Несмотря на то, что демонстрационное приложение написано на C#, большинство функций применимы к C++, Visual Basic, F#, Python, JavaScript и другим языкам, поддерживаемым Visual Studio (F# не поддерживает возможность "Изменить и продолжить"). F# и JavaScript не поддерживают окно **Видимые**). На снимках экрана представлены примеры на C#.

Рассмотрены следующие задачи:

- Запуск отладчика и попадание в точки останова.
- Использование команд для пошагового выполнения кода в отладчике.
- Проверка переменных в подсказках к данным и окнах отладчика.
- Просмотр стека вызовов

Создание проекта

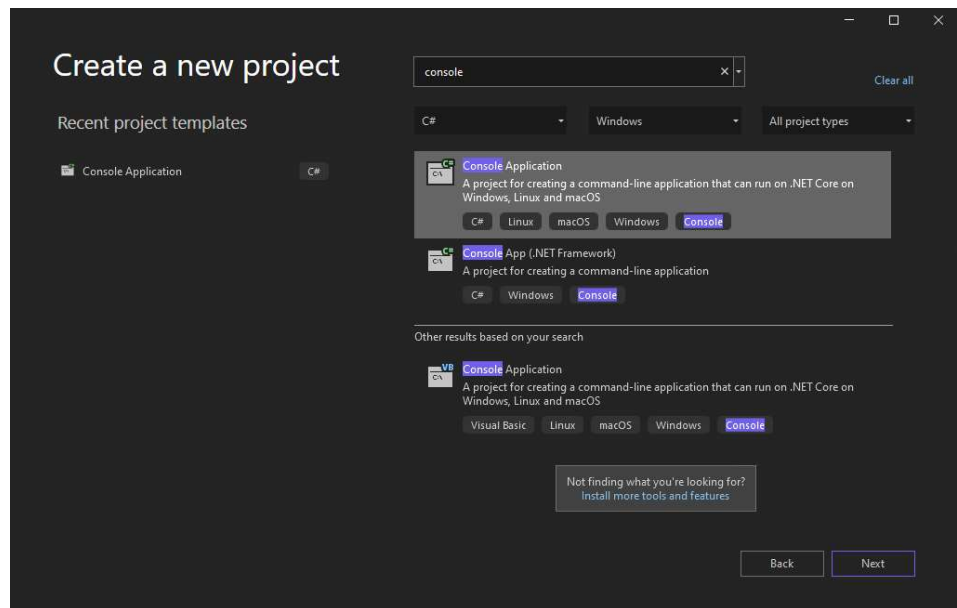
Сначала вы создадите проект консольного приложения .NET Core. Для этого типа проекта уже имеются все нужные файлы шаблонов, что избавляет вас от лишней работы.

1. Запустите Visual Studio.

Если окно запуска не открыто, выберите **Файл > Окно запуска**.

2. На начальном экране выберите **Создать проект**.
3. В поле поиска окна **Создание проекта** введите *консоль*. Затем выберите **C#** в списке языков и **Windows** в списке платформ.

Применив фильтры языка и платформы, выберите шаблон **Консольное приложение** и нажмите кнопку **Далее**.



4. В поле **Имя проекта** окна **Настроить новый проект** введите *GetStartedDebugging*. Нажмите кнопку **Далее**.

Новый проект открывается в Visual Studio.

Создание приложения

В файле *Program.cs* замените все его содержимое по умолчанию следующим кодом:

```
using System;
```

```
class ArrayExample
```

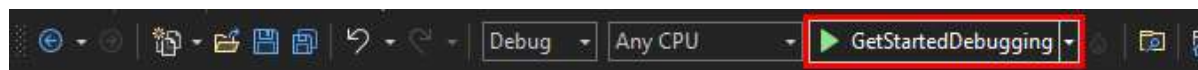
```
{  
    static void Main()  
    {  
        char[] letters = { 'f', 'r', 'e', 'd', ' ', 's', 'm', 'i', 't', 'h'};  
        string name = "";  
        int[] a = new int[10];  
        for (int i = 0; i < letters.Length; i++)  
        {  
            name += letters[i];  
            a[i] = i + 1;  
            SendMessage(name, a[i]);  
        }  
        Console.ReadKey();  
    }  
}
```

```
static void SendMessage(string name, int msg)
{
    Console.WriteLine("Hello, " + name + "! Count to " + msg);
}
}
```

Запуск отладчика

В основном мы используем сочетания клавиш, поскольку так быстрее выполнять команды отладчика. Кроме того, указаны эквивалентные команды, такие как команды панели инструментов или меню.

1. Чтобы запустить отладчик, нажмите клавишу **F5** или кнопку **Целевой объект отладки** на стандартной панели инструментов, или же нажмите кнопку **Начать отладку** на панели инструментов "Отладка" или выберите **Отладка > Начать отладку** в строке меню.



Клавиша **F5** запускает приложение с прикрепленным к процессу приложения отладчиком. Так как специально для изучения кода ничего сделано не было, приложение будет выполняться до завершения и появятся выходные данные консоли.

```
cmdКопировать
Hello, f! Count to 1
Hello, fr! Count to 2
Hello, fre! Count to 3
Hello, fred! Count to 4
Hello, fred ! Count to 5
Hello, fred s! Count to 6
Hello, fred sm! Count to 7
Hello, fred smi! Count to 8
Hello, fred smit! Count to 9
Hello, fred smith! Count to 10
```

2. Чтобы отключить отладчик, нажмите сочетание клавиш **SHIFT+F5** или кнопку **Остановить отладку** на панели инструментов "Отладка" или выберите **Отладка > Остановить отладку** в строке меню.



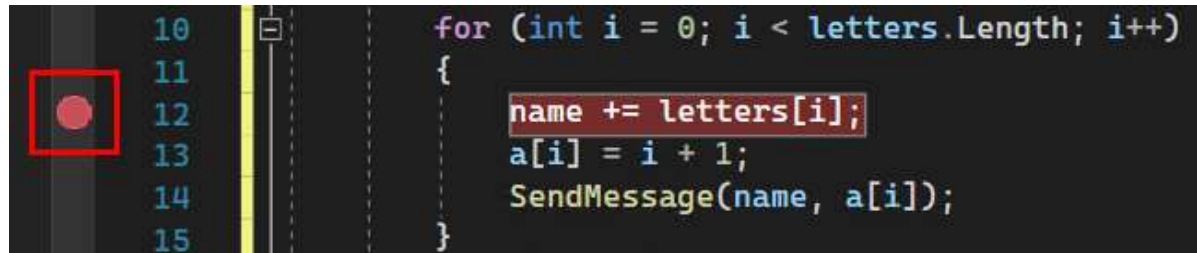
3. В окне консоли нажмите любую клавишу, чтобы закрыть его.

Установка точки останова и запуск отладчика

1. В цикле for функции Main установите точку останова, щелкнув левое поле следующей строки кода:

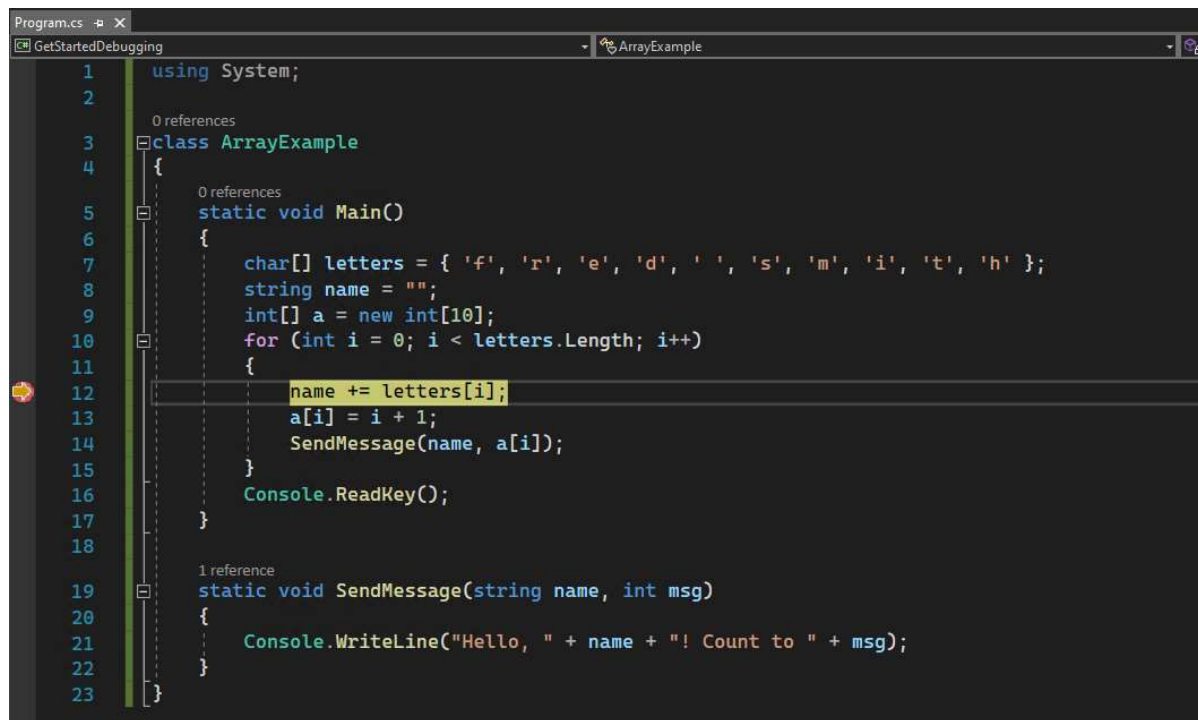
```
name += letters[i];
```

В месте установки точки останова появится красный круг.



Точки останова — это важные компоненты надежной отладки. Точки останова можно установить там, где Visual Studio следует приостановить выполнение кода, чтобы вы могли проверить значения переменных или поведение памяти либо выполнение ветви кода.

2. Чтобы запустить отладку, нажмите клавишу **F5** или кнопку **Целевой объект отладки** на стандартной панели инструментов, или же нажмите кнопку **Начать отладку** на панели инструментов "Отладка" или выберите **Отладка > Начать отладку** в строке меню. Будет запущено приложение, и отладчик перейдет к строке кода, где задана точка останова.



```
1 using System;
2
3 class ArrayExample
4 {
5     static void Main()
6     {
7         char[] letters = { 'f', 'r', 'e', 'd', ' ', 's', 'm', 'i', 't', 'h' };
8         string name = "";
9         int[] a = new int[10];
10        for (int i = 0; i < letters.Length; i++)
11        {
12            name += letters[i];
13            a[i] = i + 1;
14            SendMessage(name, a[i]);
15        }
16        Console.ReadKey();
17    }
18
19    static void SendMessage(string name, int msg)
20    {
21        Console.WriteLine("Hello, " + name + "! Count to " + msg);
22    }
23 }
```

Желтая стрелка указывает на оператора, на котором приостановлен отладчик. Выполнение приложения приостановлено в той же точке, но оператор еще не выполнен.

Если приложение не запущено, нажмите клавишу **F5** и запустите отладчик, который будет выполнять приложение, пока не достигнет первой точки останова. Если приложение приостановлено в точке останова, нажав клавишу **F5**, можно продолжить выполнение приложения, пока оно не достигнет следующей точки останова.

Точки останова будут полезны, если вам известны строка или раздел кода, поэтому изучите их подробно.

Навигация по коду и проверка данных с помощью подсказок по данным

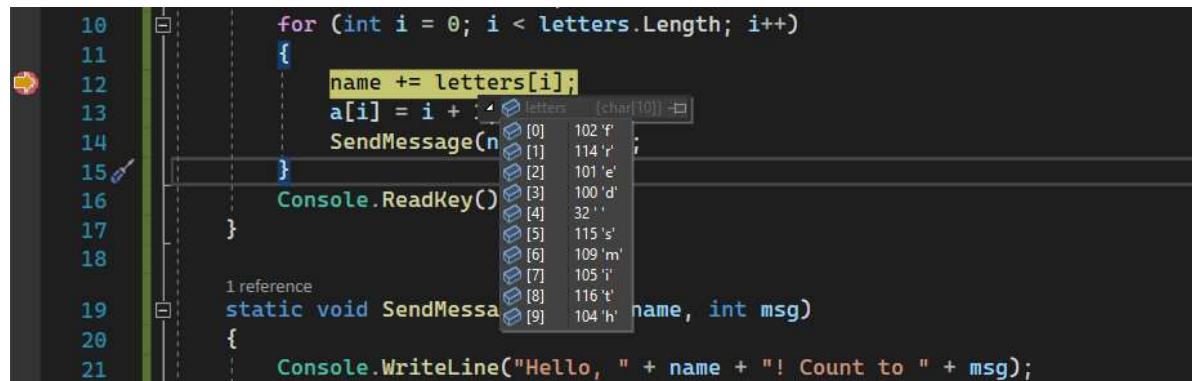
1. Во время приостановки выполнения оператора `name += letters[i]` наведите указатель мыши на переменную `letters`, чтобы увидеть подсказку по данным, в которой указан размер массива и тип элемента `char[10]`.

Примечание

Одной из наиболее полезных возможностей отладчика является его способность проверять переменную. Часто при попытке выполнить отладку после возникновения проблемы пользователь старается выяснить, содержат ли переменные значения, которые

требуются ему в определенное время. Хорошим способом проверить это является просмотр подсказок по данным.

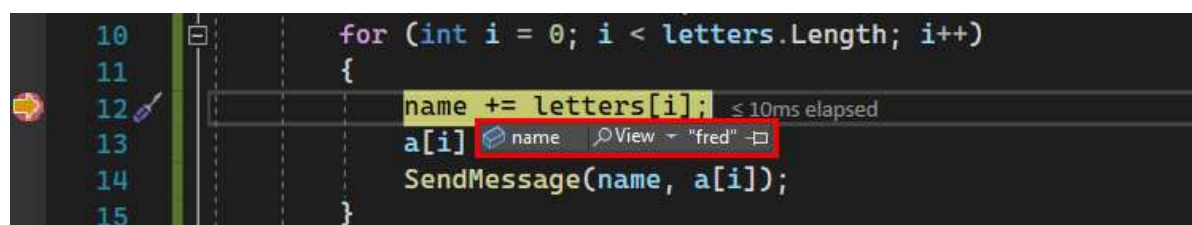
2. Разверните переменную `letters`, чтобы просмотреть все ее элементы массива и их значения.



3. Наведите указатель мыши на переменную `name`, чтобы просмотреть ее текущее значение — пустую строку.
4. Чтобы переместить отладчик к следующему оператору, нажмите клавишу **F10** или кнопку **Шаг с обходом** на панели инструментов "Отладка" или выберите **Отладка > Шаг с обходом** в строке меню. Дважды нажмите клавишу **F10**, чтобы перейти за пределы вызова метода `SendMessage`.

При нажатии клавиши **F10** отладчик выполняется без захода в функции или методы в коде приложения, хотя код продолжает выполняться. Таким образом мы пропустили отладку кода в методе `SendMessage`, который сейчас нас не интересует.

5. Чтобы несколько раз выполнить итерацию по циклу `for`, повторно нажимайте клавишу **F10**. Во время каждой итерации цикла сделайте приостановку в точке останова, а затем наведите указатель мыши на переменную `name`, чтобы проверить ее значение в подсказке по данным.



Значение переменной изменяется при каждой итерации цикла `for` — `f`, затем `fr`, `fre` и т. д. Чтобы ускорить работу отладчика при прохождении цикла, взамен нажмите клавишу **F5** и перейдите к вашей точке останова вместо следующего оператора.

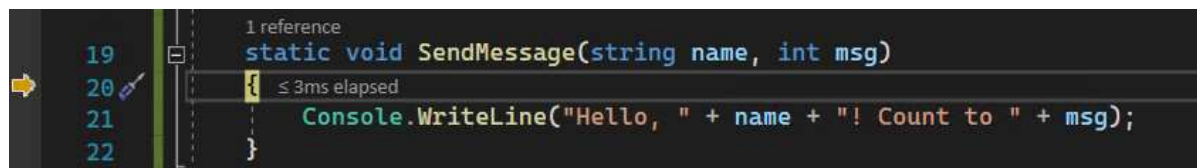
6. Во время приостановки в цикле for метода Main нажмите клавишу **F11** или кнопку **Шаг с заходом** на панели инструментов "Отладка" или выберите **Отладка > Шаг с заходом** в строке меню, пока не будет достигнут вызов метода SendMessage.

Отладчик должен быть приостановлен в следующей строке кода:

```
SendMessage(name, a[i]);
```

7. Чтобы выполнить шаг с заходом в метод SendMessage, снова нажмите клавишу **F11**.

Желтый указатель перемещается в метод SendMessage.



Нажав клавишу **F11**, вы сможете более подробно изучить поток выполнения кода. Чтобы выполнить шаг с заходом в метод из вызова метода, нажмите клавишу **F11**. По умолчанию отладчик пропускает шаг с заходом в непользовательские методы.

После завершения отладки метода SendMessage можно вернуться к циклу for метода main.

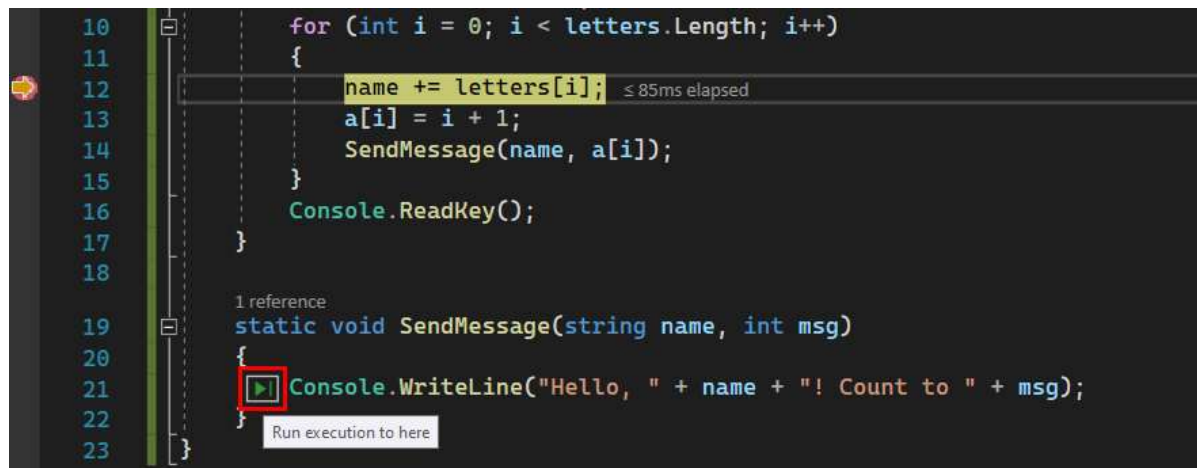
8. Чтобы выйти из метода SendMessage, нажмите сочетание клавиш **SHIFT+F11** или кнопку **Шаг с выходом** на панели инструментов отладки или выберите **Отладка > Шаг с выходом** в строке меню.

Шаг с выходом возобновляет выполнение приложения и работу отладчика до возврата текущего метода или функции.

Желтый указатель будет отображаться в цикле for метода Main, приостанавливаясь в вызове метода SendMessage.

Переход по коду с помощью команды "Выполнение до щелкнутого"

1. Нажмите клавишу **F5**, чтобы снова перейти к точке останова.
2. В редакторе кода наведите указатель мыши на метод Console.WriteLine в методе SendMessage, чтобы в левой части появилась кнопка **Выполнение до щелчка**. В подсказке для кнопки выводится "Выполнить до этого места".



3. Нажмите на кнопку **Выполнение до щелчка**. Кроме того, вы можете навести курсор на оператор `Console.WriteLine` и нажать сочетание клавиш **Ctrl+F10**. Другим способом щелкните вызов метода `Console.WriteLine` правой кнопкой мыши и выберите в контекстном меню пункт **Выполнение до текущей позиции**.

Отладчик перемещается к вызову метода `Console.WriteLine`.

Использование кнопки **Выполнение до щелчка** аналогично настройке временной точки останова и удобно для быстрого обхода в видимой области кода приложения в открытом файле.

Быстрый перезапуск приложения

Чтобы повторно запустить приложение с самого начала в отладчике, нажмите сочетание клавиш **CTRL+SHIFT+F5** или кнопку **Перезапустить** на панели инструментов "Отладка" или выберите **Отладка>Перезапустить** в строке меню.



Кнопка **Перезапустить** останавливает отладчик, а затем сразу перезапускает его. При перезапуске отладчик запустится до первой точки останова, заданной ранее в цикле `for`, а затем приостановится.

Проверка переменных с помощью окон "Видимые" и "Локальные"

В окнах **Видимые** и **Локальные** отображаются значения переменных во время отладки. Окна доступны только во время сеанса отладки. В окне **Видимые** отображаются переменные, используемые в текущей строке, в которой находится отладчик, и в предыдущей строке. В окне **Локальные** отображаются переменные, определенные в локальной области, которая обычно является текущей функцией или методом.

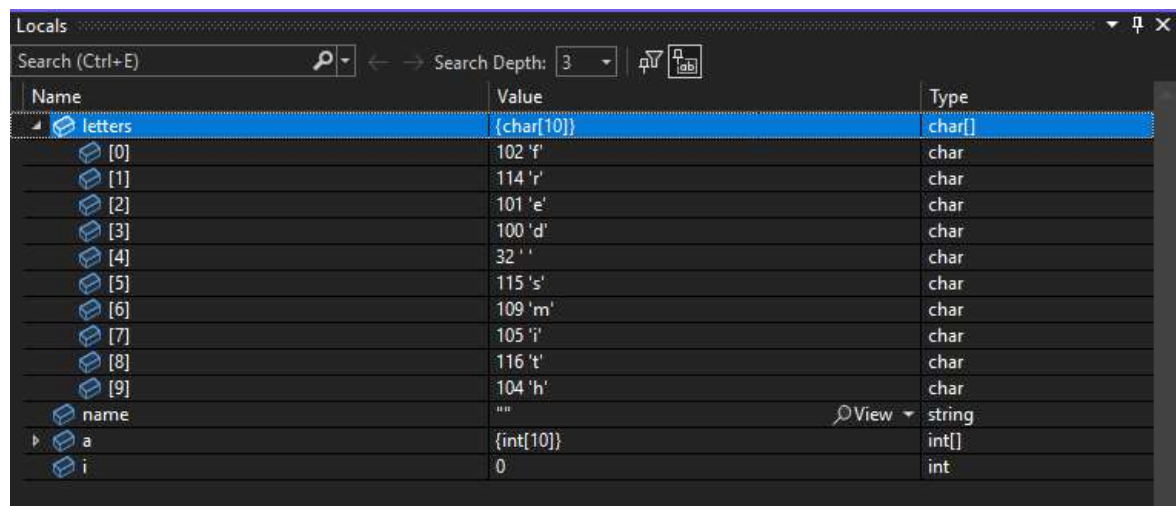
1. Во время приостановки отладчика просмотрите окно **Видимые** в нижней части редактора кода.

Если окно **Видимые** закрыто, нажмите клавиши **Ctrl+D, A** или в строке меню выберите **Отладка > Windows > Видимые**.

2. Если отладчик по-прежнему приостановлен, просмотрите окно **Локальные**, расположенное во вкладке рядом с окном **Видимые**.

Если окно **Локальные** закрыто, нажмите сочетание клавиш **Ctrl+D, L** или выберите **Отладка > Windows > Локальные**.

3. В окне **Локальные** разверните переменную `letters`, чтобы просмотреть все ее элементы массива и их значения.



Установка контрольного значения

Вы можете указать переменную или выражение, которое необходимо отслеживать во время поэтапного выполнения кода —, добавив его в окно **контрольных значений**.

1. Когда отладчик будет приостановлен, щелкните правой кнопкой мыши переменную `name` и выберите **Добавить контрольное значение**.

В нижней части редактора кода по умолчанию откроется окно **контрольных значений**.

2. Теперь, когда вы установили контрольное значение для переменной `name`, пошагово выполните код, чтобы увидеть, как

значение переменной `name` меняется с каждой итерацией цикла `for`.

В отличие от других окон переменных, в окне **контрольных значений** всегда отображаются просматриваемые вами переменные, но они выделяются серым цветом, когда находятся вне области действия.

Просмотр стека вызовов

Стек вызовов может помочь разобраться в потоке выполнения приложения, показывая порядок, в котором вызываются методы и функции.

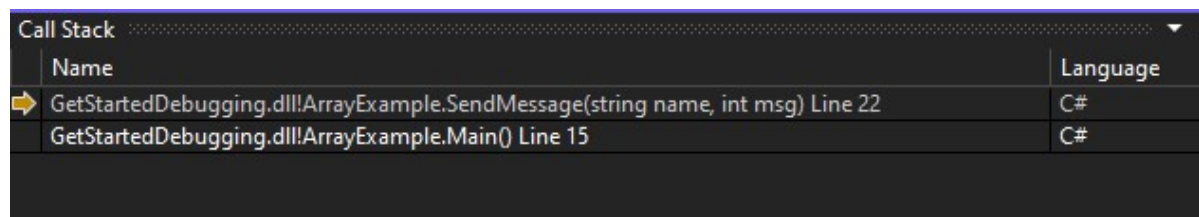
1. Пока отладчик приостановлен в цикле `for`, просмотрите окно **Стек вызовов**, которое по умолчанию открывается в нижней правой области редактора кода.

Если окно **Стек вызовов** закрыто, нажмите сочетание клавиш **Ctrl+D, A** или в строке меню выберите **Отладка >Windows >Стек вызовов**.

В окне **Стек вызовов** вы увидите желтый указатель в текущем методе `Main`.

2. Несколько раз нажмите клавишу **F11**, пока отладчик не приостановится в методе `SendMessage`.

В верхней строке окна **Стек вызовов** показана текущая функция – метод `SendMessage`. Во второй строке показано, что метод `SendMessage` был вызван из метода `Main`.



	Name	Language
➡	GetStartedDebugging.dll!ArrayExample.SendMessage(string name, int msg) Line 22	C#
	GetStartedDebugging.dll!ArrayExample.Main() Line 15	C#

Примечание

Окно **Стек вызовов** аналогично перспективе "Отладка" в некоторых наборах IDE, например Eclipse.

В окне **Стек вызовов** дважды щелкните строку кода, чтобы перейти к этому исходному коду. При этом изменится текущая область, проверяемая отладчиком. Это действие не перемещает отладчик.

Для выполнения других задач можно воспользоваться контекстными меню из окна **Стек вызовов**. Например, можно вставить точки останова в указанные функции, переместить отладчик с помощью функции **Выполнение до текущей позиции** или перейти к исходному коду.

Изменение потока выполнения

Вы можете переместить указатель выполнения, чтобы изменить поток приложения во время отладки.

1. Когда отладчик приостановится на вызове метода `SendMessage` в цикле `for`, три раза нажмите клавишу **F11**, чтобы выполнить шаг с заходом в метод `SendMessage` и перейти за пределы метода `Console.WriteLine` его выполнения.

Теперь отладчик приостанавливается на последней закрывающей фигурной скобке метода `SendMessage`.

2. Мышью захватите желтую стрелку или указатель выполнения (в поле слева), а затем перетащите указатель вверх на одну строку.

Теперь отладчик снова находится в операторе `Console.WriteLine`.

3. Нажмите клавишу **F11**.

Отладчик повторно выполнит метод `Console.WriteLine`, и вы увидите повторяющиеся строки в выходных данных окна консоли.

4. Чтобы продолжить выполнение приложения, нажмите клавишу **F5**.

Изменяя поток выполнения, можно решать множество задач, например тестировать различные пути выполнения кода или повторно выполнять код без перезапуска отладчика.

Предупреждение

Используйте эту возможность осторожно. В подсказке указателя выполнения появится предупреждение о возможных непредвиденных последствиях. Могут также появиться другие предупреждения. Переместив указатель выполнения, нельзя вернуть приложение в предыдущее состояние.