**Практическая работа №1**

**Разработка и отладка модуля обработки элементов массива. Отладка кода C# с помощью инструментов Visual Studio.**

**14 Вариант**

Отладка приложения обычно означает запуск и выполнение приложения с подключенным отладчиком. При этом в отладчике доступно множество способов наблюдения за выполнением кода. Можно 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.

Цель:

изучить

* 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

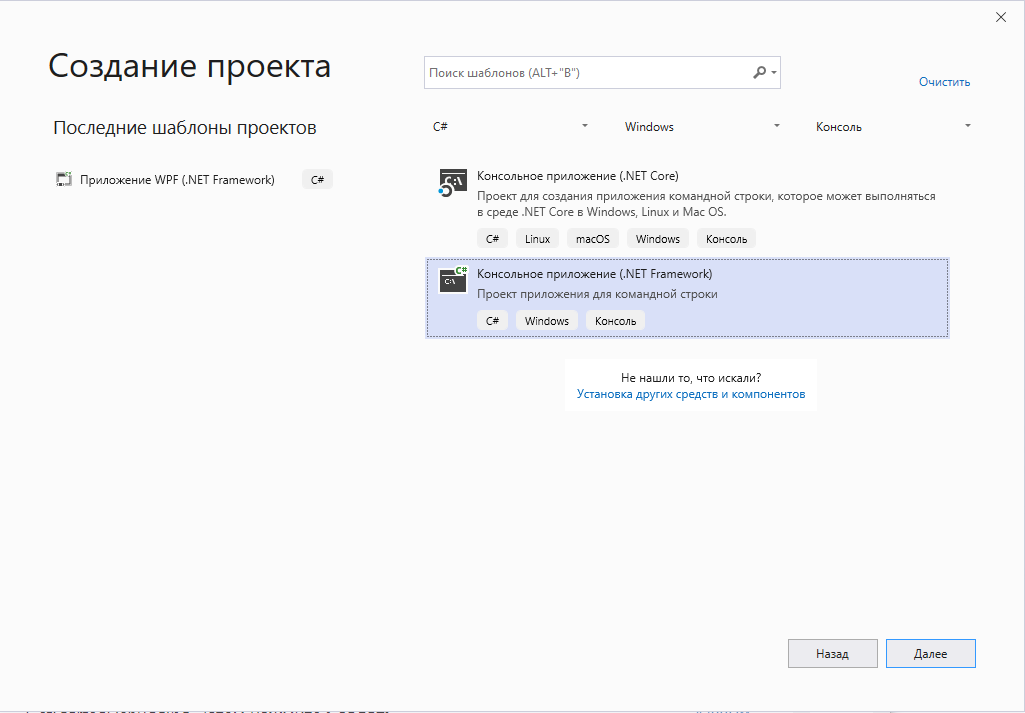
# Создание проекта

Создать проект консольного приложения .NET Framework.  Запустить Visual Studio 2022.

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

1. На начальном экране выберите **Создать проект**.
2. 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 (.NET Framework)** и нажмите кнопку **Далее**.

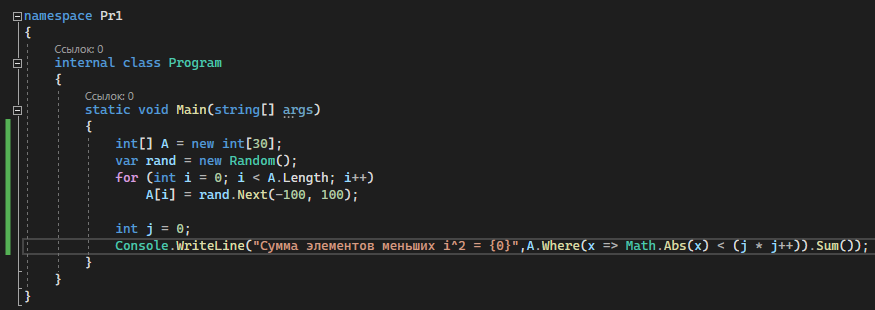


1. В поле **Имя проекта** окна **Настроить новый проект** введите *GetStartedDebugging*, выберите расположение проекта. Затем нажмите **Создать**.

Новый проект открывается в Visual Studio.

# Создание приложения

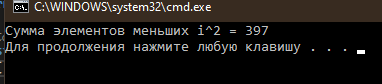
1. Откройте файл Program.cs и замените все его содержимое по умолчанию следующим кодом:



# Запуск отладчика

1. Нажмите клавишу **F5** (**Отладка > Начать отладку**) или кнопку **Начать отладку**Начать отладку на панели инструментов отладки.

При нажатии клавиши **F5** происходит запуск приложения с присоединенным отладчиком. Но пока мы не сделали ничего особенного, чтобы проанализировать код. Поэтому приложение будет просто загружено, и вы увидите выходные данные консоли.



1. Остановите отладчик, нажав красную кнопку остановки Остановить отладку (**SHIFT** + **F5**).
2. В окне консоли нажмите 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:

A[i] = rand.Next(-100, 100);

В месте установки точки останова появится красный круг Точка останова.

Точки останова — это один из самых простых и важных компонентов надежной отладки. Точка останова указывает, где Visual Studio следует приостановить выполнение кода, чтобы вы могли проверить значения переменных или поведение памяти либо выполнение ветви кода.

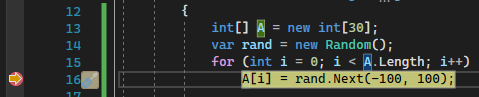
1. Нажмите клавишу **F5** или кнопку **Начать отладку**Начать отладку. Запустится приложение и отладчик перейдет к строке кода, где задана точка останова.



Желтая стрелка представляет оператор, на котором приостановлен отладчик. В этой же точке приостанавливается выполнение приложения (этот оператор пока не выполнен).

Если приложение еще не запущено, клавиша **F5** запускает отладчик и останавливается в первой точке останова. В противном случае **F5** продолжает выполнение приложения до следующей точки останова.

Точки останова полезны, если вам известны строка или раздел кода, которые вы хотите подробно изучить.



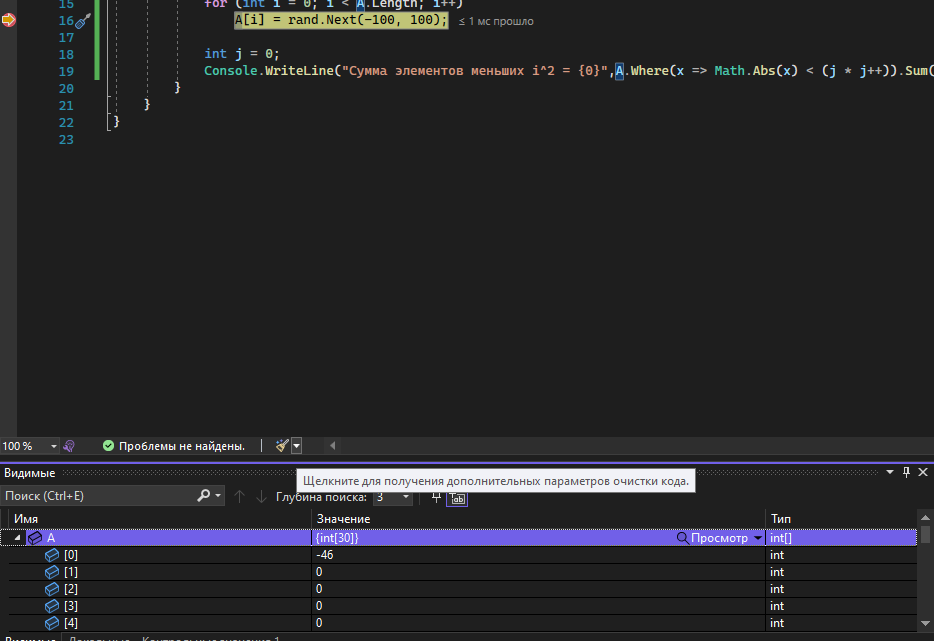
# Навигация по коду и проверка данных с помощью подсказок по данным

Здесь мы используем в основном сочетания клавиш, так как они позволяют быстро выполнять приложение в отладчике (эквивалентные команды, например команды меню, отображаются в круглых скобках).

1. При приостановке на операторе name += letters[i] наведите указатель мыши на переменную letters и увидите ее значение по умолчанию, а именно значение первого элемента в массиве — char[0].

Функции, позволяющие проверять переменные, являются самыми полезными возможностями отладчика. Реализовывать эту задачу можно разными способами. Часто при попытке выполнить отладку проблемы пользователь старается выяснить, хранятся ли в переменных значения, которые требуются ему в определенное время.

1. Разверните переменную letters, чтобы просмотреть ее свойства, включая все элементы, которые она содержит.



1. Затем наведите указатель мыши на переменную name, чтобы просмотреть ее текущее значение — пустую строку.
2. Нажмите клавишу **F10** (или выберите **Отладка > Шаг с обходом**) дважды, чтобы перейти к вызову метода SendMessage, а затем еще раз нажмите клавишу **F10**.

При нажатии клавиши F10 отладчик переходит к следующей инструкции без захода в функции или методы в коде приложения (код продолжает выполняться). Нажимая клавишу F10 в вызове метода SendMessage, мы пропускаем код реализации для SendMessage (который нас пока не интересует).

1. Несколько раз нажмите клавишу **F10** (или выберите **Отладка** > **Шаг с обходом**), чтобы выполнить несколько итераций по циклу for, каждый раз снова приостанавливая выполнение в точке останова и наводя указатель мыши на переменную name, чтобы просмотреть ее значение.

Значение переменной изменяется при каждой итерации цикла for — f, затем fr, fre и т. д. Для более быстрого прохода по циклу в этом сценарии можно нажать клавишу **F5** (или выбрать **Отладка** > **Продолжить**), чтобы перейти к точке останова, а не к следующей инструкции.

Часто при отладке требуется быстро проверить значения свойств в переменных, чтобы убедиться, что в них хранятся ожидаемые значения. Советы по данным — отличный способ это сделать.

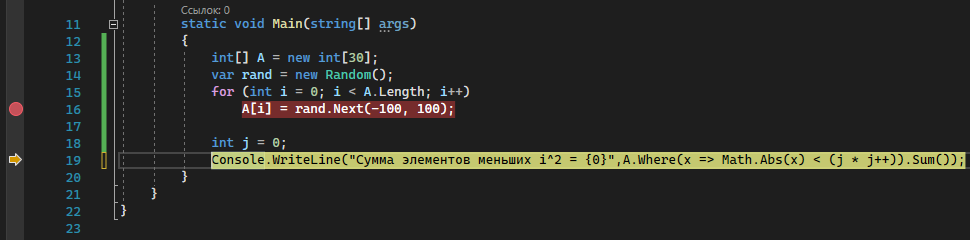
1. Во время приостановки в цикле for в методе Main нажмите клавишу **F11** (или выберите **Отладка > Шаг с заходом**) несколько раз, пока не перейдете в вызов метода SendMessage.

Вы должны находиться в следующей строке кода:

SendMessage(name, a[i]);

1. Еще раз нажмите клавишу **F11**, чтобы выполнить шаг с заходом в метод SendMessage.

Желтый указатель перемещается в метод SendMessage.



F11 — это команда **Шаг с заходом**, которая выполняет приложение с переходом к следующему оператору. Клавишу F11 удобно использовать для более детальной проверки потока выполнения. По умолчанию отладчик пропускает непользовательский код (дополнительные сведения см. в статье об [отладке в режиме "Только мой код"](https://docs.microsoft.com/ru-ru/visualstudio/debugger/just-my-code?view=vs-2019)).

Предположим, что вы закончили изучать метод SendMessage и хотите выйти из него, но остаться в отладчике. Это можно сделать с помощью команды **Шаг с выходом**.

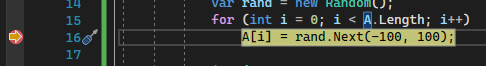
1. Нажмите сочетание клавиш **SHIFT** + **F11** (или **Отладка > Шаг с выходом**).

Эта команда возобновляет выполнение приложения (и работу отладчика) до возврата данных текущим методом или текущей функции.

Вы должны вернуться в цикл for в методе Main, приостановленный на 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 прокрутите вниз и наведите указатель мыши на метод Console.WriteLine в методе SendMessage, чтобы в левой части появилась зеленая кнопка **Выполнение до щелкнутого**https://docs.microsoft.com/ru-ru/visualstudio/debugger/media/dbg-tour-run-to-click.png?view=vs-2019. В подсказке для кнопки выводится "Выполнить до этого места".



1. Нажмите кнопку **Выполнение до щелкнутого**Выполнение до щелкнутого.

Отладчик перемещается к методу Console.WriteLine.

Использование этой кнопки аналогично установке временной точки останова. Функция **Выполнение до щелкнутого** удобна для быстрой работы в видимой области кода приложения (можно щелкнуть в любом открытом файле).

# Быстрый перезапуск приложения

Нажмите кнопку **Перезапустить** Перезапустить приложение на панели инструментов отладки (**CTRL** + **SHIFT** + **F5**).

Кнопка **Перезапустить** позволяет сэкономить время, затрачиваемое на остановку приложения и перезапуск отладчика. Отладчик приостанавливается в первой точке останова, достигнутой при выполнении кода.

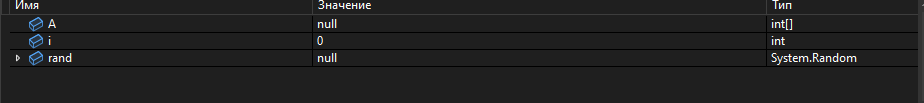
Отладчик еще раз останавливается в точке останова, ранее заданной вами в цикле for.

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

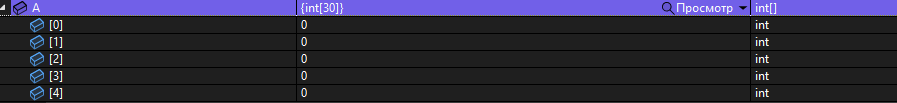
1. Взгляните на окно **Видимые** в нижней части редактора кода.

Если оно закрыто, откройте его во время приостановки в отладчике, выбрав **Отладка** > **Окна** > **Видимые**.

В окне **Видимые** отображаются переменные и их текущие значения. В окне **Видимые** отображаются все переменные, используемые в текущей или предыдущей строке (сведения о зависящем от языка поведении см. в соответствующей документации).



1. Затем посмотрите на окно **Локальные** на вкладке рядом с окном **Видимые**.
2. Разверните переменную letters, чтобы отобразить элементы, которые она содержит.



# Установка контрольного значения

1. В основном окне редактора кода щелкните правой кнопкой мыши переменную name и выберите команду **Добавить контрольное значение**.

В нижней части редактора кода откроется окно **Контрольное значение**. В окне **Контрольное значение** можно указать переменную (или выражение), которую необходимо отслеживать.

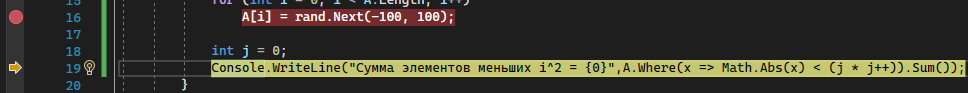
Теперь у вас есть контрольное значение, заданное для переменной name, и по мере перемещения по отладчику вы можете наблюдать за изменением его значения. В отличие от других окон переменных, в окне **Контрольное значение** всегда отображаются просматриваемые вами переменные (они выделяются серым цветом, когда находятся вне области действия).



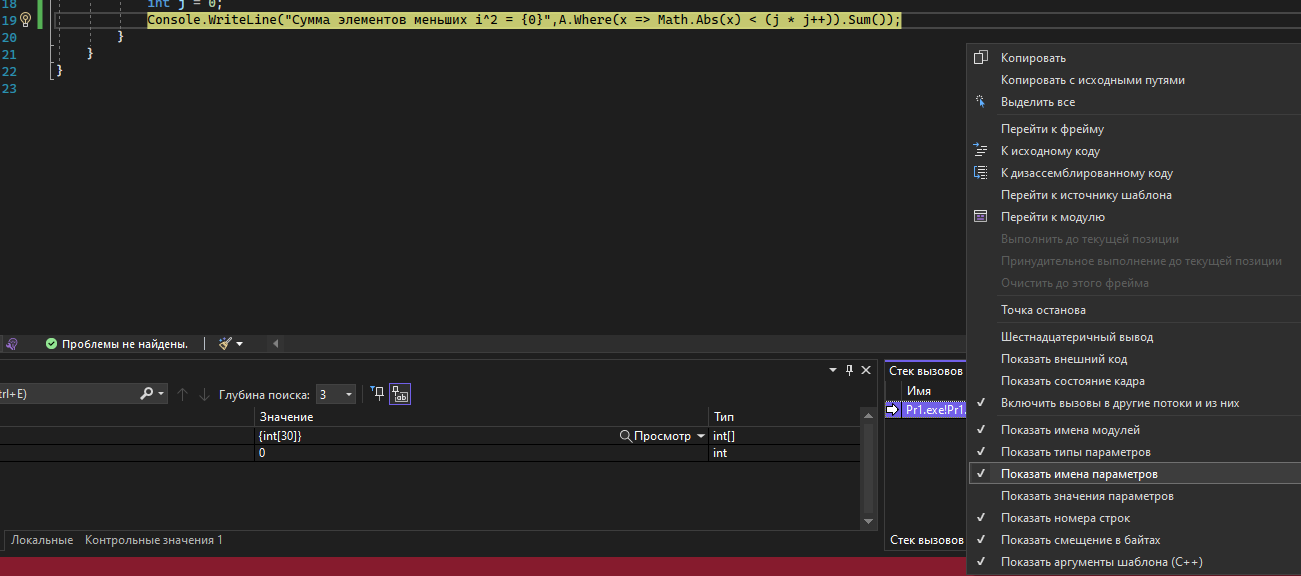
# Просмотр стека вызовов

1. Во время приостановки в цикле for щелкните окно **Стек вызовов**, которое по умолчанию открыто в нижней правой области. Если оно закрыто, откройте его во время приостановки в отладчике, выбрав **Отладка** > **Окна** > **Стек вызовов**.
2. Несколько раз нажмите клавишу **F11**, пока отладчик не приостановится в методе SendMessage. Взгляните на окно **Стек вызовов**.

В окне **Стек вызовов** показан порядок вызова методов и функций. В верхней строке приведена текущая функция (в данном приложении метод SendMessage). Во второй строке показано, что функция SendMessage была вызвана из метода Main и т. д.

Стек вызовов хорошо подходит для изучения и анализа потока выполнения приложения. 

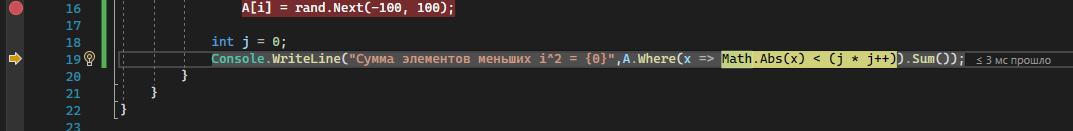
Для выполнения других задач можно воспользоваться контекстными меню из окна **Стек вызовов**. Например, можно вставлять точки останова в указанные функции, перемещать отладчик с помощью функции **Выполнение до текущей позиции** и изучать исходный код.



# Изменение потока выполнения

1. Дважды нажмите клавишу **F11**, чтобы запустить метод Console.WriteLine.
2. Приостановив отладчик в вызове метода SendMessage, с помощью мыши захватите желтую стрелку (указатель выполнения) в левой части и переместите ее вверх на одну строку — обратно в Console.WriteLine.
3. Нажмите клавишу **F11**.

Отладчик повторно выполнит метод Console.WriteLine (вы увидите это в выходных данных окна консоли).



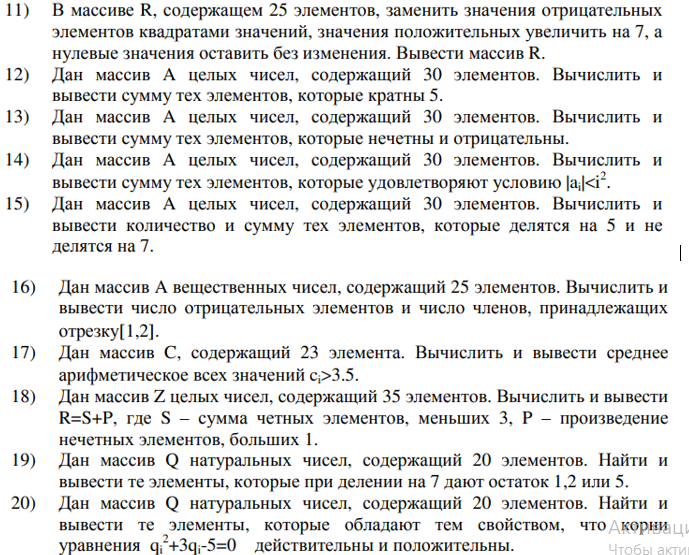
Изменяя поток выполнения, можно решать множество задач, например тестировать различные пути выполнения кода или повторно выполнять код без перезапуска отладчика.

**Предупреждение**

Как правило, при работе с этой функцией необходимо соблюдать осторожность — вы увидите соответствующее предупреждение во всплывающей подсказке. Могут отображаться и другие предупреждения. При перемещении указателя предыдущее состояние приложения не возвращается.

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

***Индивидуальное задание.***

******

***Сдать на проверку архив с приложением, отчет в формате .docx***