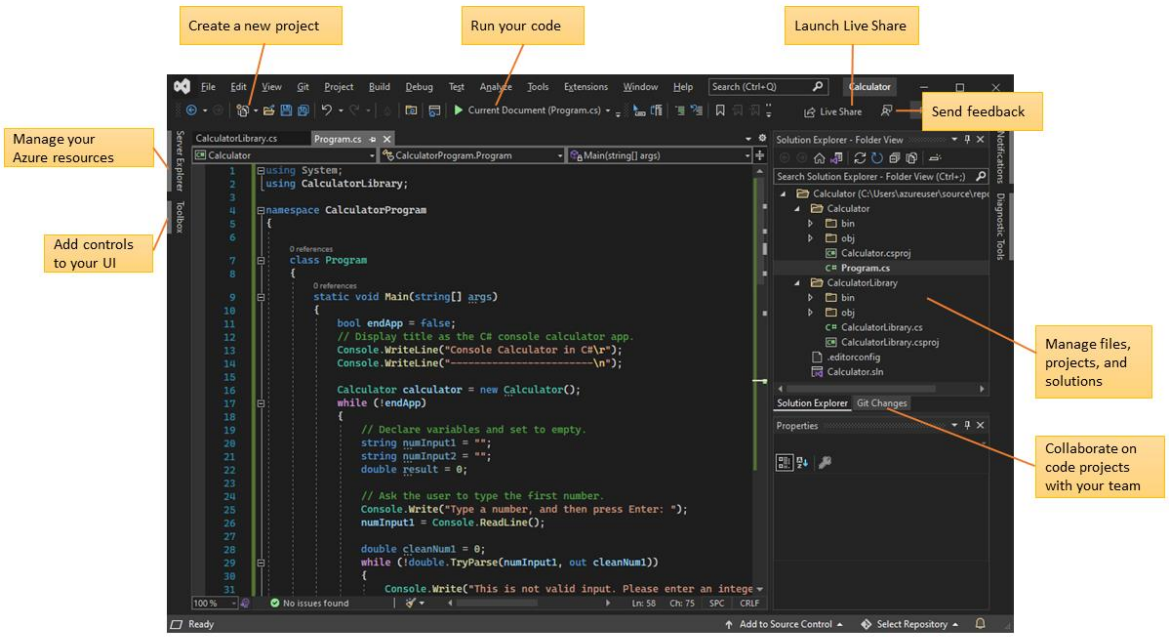
Введение в интегрированную среду разработки Visual Studio (C#)

Интегрированная среда разработки (IDE) — это многофункциональная программа, которая поддерживает многие аспекты разработки программного обеспечения. Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки.



На рисунке выше представлена среда Visual Studio с открытым проектом и подсказки по основным окнам и функциональных возможностях.

- Справа в верхнем углу окна Обозревателя решений вы можете просматривать файлы кода, перемещаться по ним и управлять ими. **Обозреватель решений** позволяет упорядочить код путем объединения файлов в решения и проекты.
- В центральном окне редактора, с которым вы, вероятно, будете работать дольше всего, отображается содержимое файла. В окне редактора вы можете вносить изменения в код или разрабатывать пользовательский интерфейс, например окно с кнопками или текстовые поля.
- Окно Изменения Git в нижнем углу справа позволяет отслеживать рабочие элементы и предоставлять общий доступ к коду, используя Git, GitHub или другие технологии управления версиями.

Выпуски

Служба Visual Studio доступна для Windows и Mac. Функции Visual Studio для Mac во многом аналогичны возможностям Visual Studio для Windows и оптимизированы для разработки кросс-платформенных и мобильных приложений. Эта статья посвящена версии Visual Studio для Windows.

Существует три выпуска Visual Studio: Community, Professional и Enterprise. Сведения о функциях, поддерживаемых в каждом выпуске, см. на странице Сравнение выпусков Visual Studio.

Популярные средства повышения производительности

Вот несколько популярных возможностей Visual Studio, которые повышают производительность при разработке программного обеспечения:

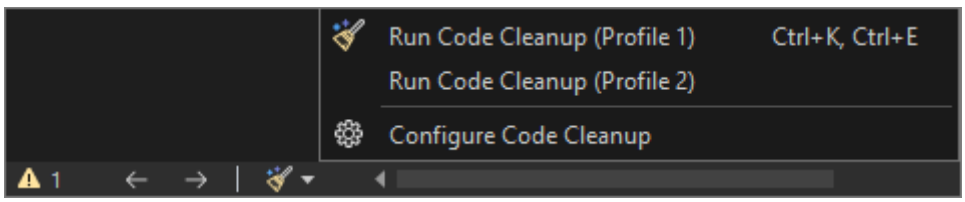
- Волнистые линии и быстрые действия

  Волнистые линии обозначают ошибки или потенциальные проблемы кода прямо во время ввода. Эти визуальные подсказки помогают немедленно устранить проблемы, не дожидаясь появления ошибок во время сборки или выполнения. Если навести указатель мыши на волнистую линию, на экран будут выведены дополнительные сведения об ошибке. Также в поле слева может отображаться лампочка, указывающая на наличие сведений о *быстрых действиях* для устранения ошибки.

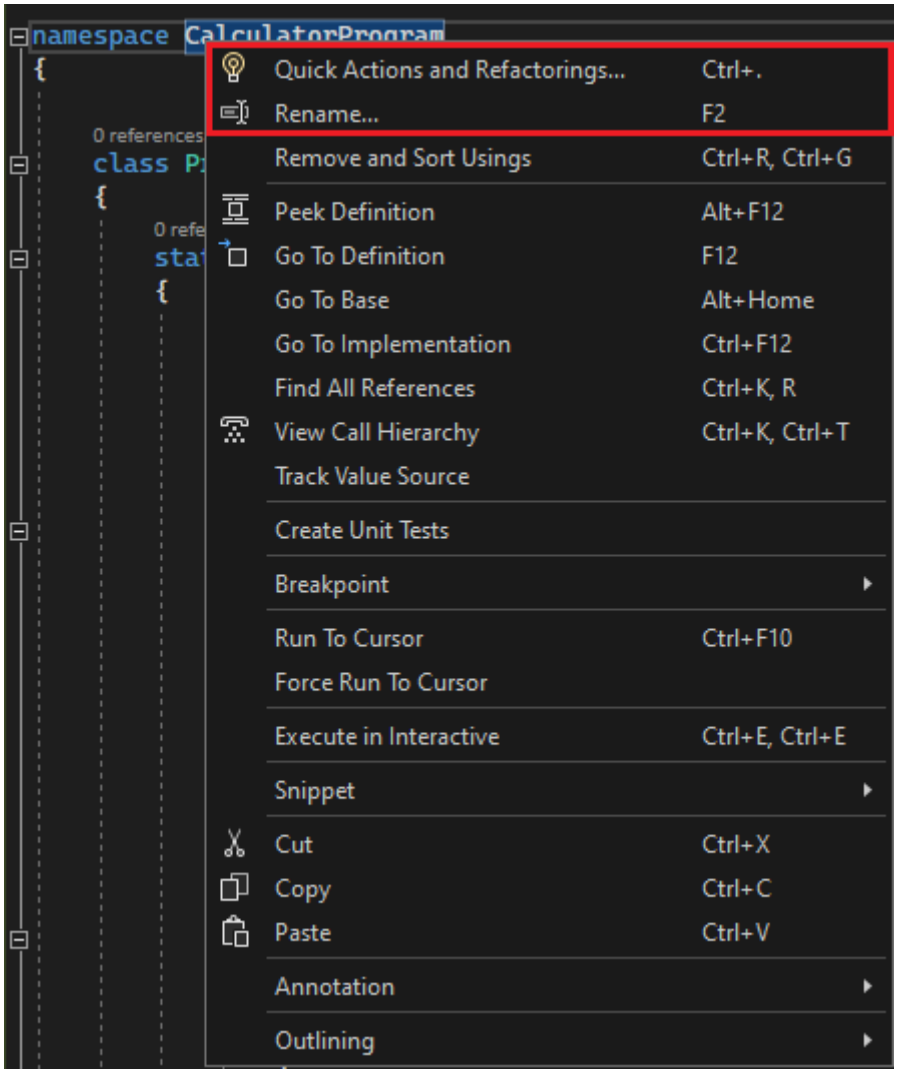
- Очистка кода

  Вы можете одним нажатием кнопки отформатировать код и применить к нему исправления, предложенные параметрами стиля кода, соглашениями в файле .editorconfig и (или) анализаторами Roslyn. **Очистка кода**, которая сейчас доступна только для кода C#, помогает устранять проблемы в коде перед переходом к его проверке.

- Рефакторинг

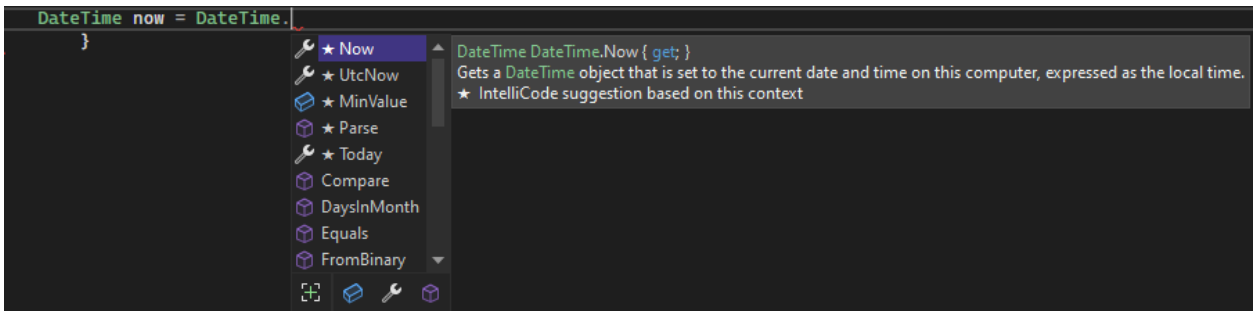
Рефакторинг включает в себя такие операции, как интеллектуальное переименование переменных, извлечение одной или нескольких строк кода в новый метод и изменение порядка параметров методов.



- IntelliSense

  IntelliSense — это набор возможностей, отображающих сведения о коде непосредственно в редакторе и в некоторых случаях автоматически создающих небольшие отрывки кода. По сути, это встроенная в редактор базовая документация, которая избавляет от необходимости искать информацию в других источниках.

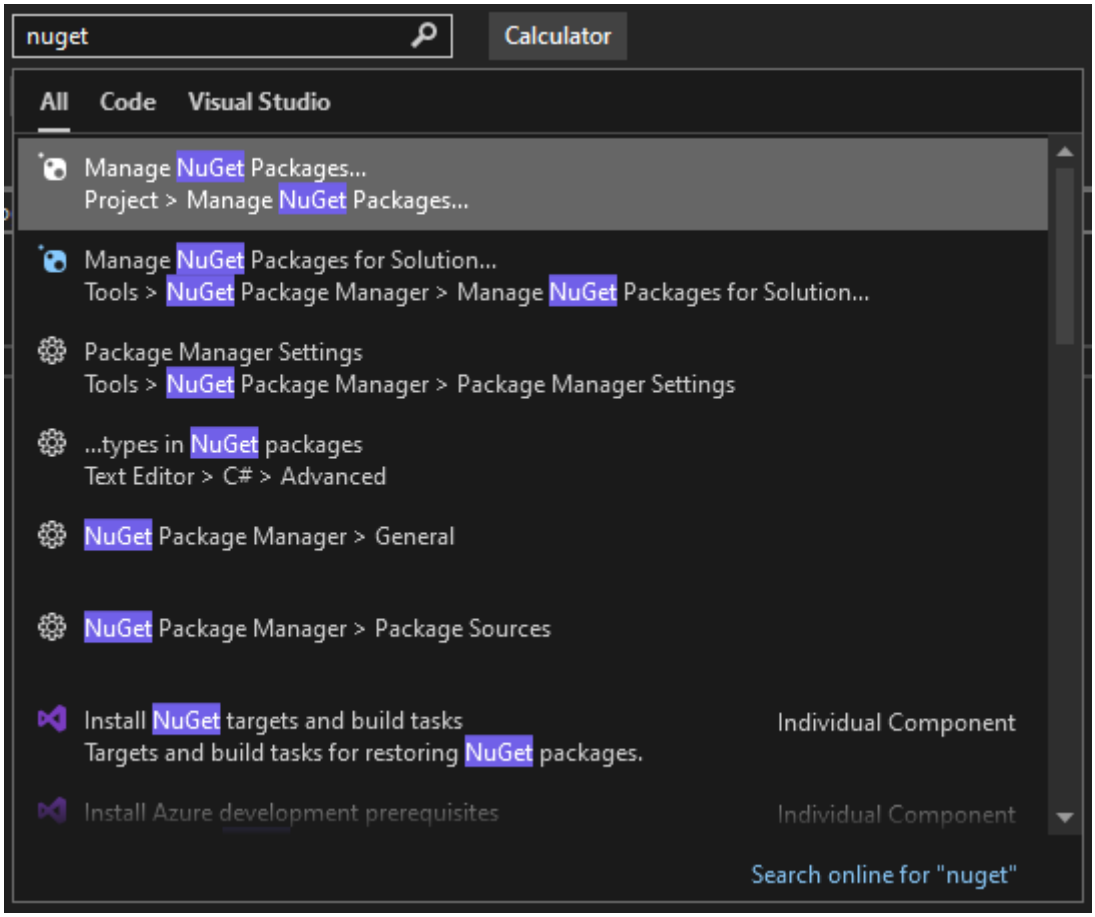
  На следующем рисунке показано, как IntelliSense отображает список членов типа:



Функции IntelliSense зависят от языка. Дополнительные сведения см. в руководствах по IntelliSense для C# , IntelliSense для Visual C++, IntelliSense для JavaScript и IntelliSense для Visual Basic.

- Поиск Visual Studio

  Иногда вам будет казаться, что в Visual Studio слишком много меню, действий и свойств. Чтобы быстро находить функции интегрированной среды разработки или элементы кода, в Visual Studio представлен единый компонент поиска (**CTRL**+**Q**).



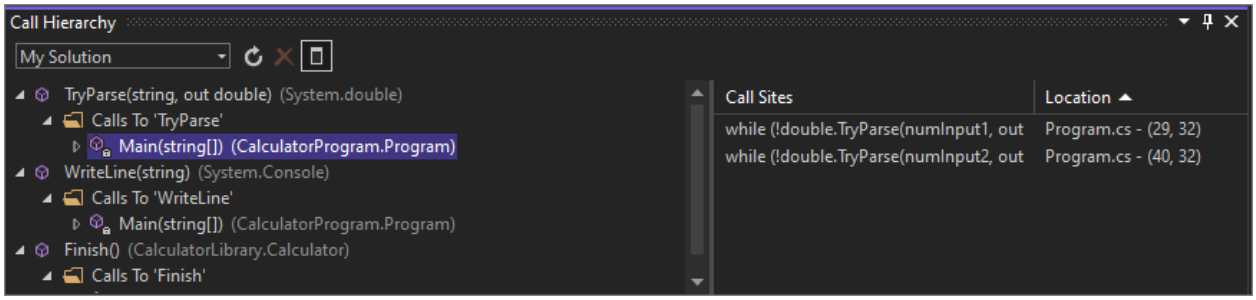
  Дополнительные сведения и советы по повышению производительности см. в разделе Практическое руководство. Поиск в Visual Studio.

- Live Share

Предоставляет возможности совместного редактирования и отладки в реальном времени независимо от типа приложения или языка. Вы можете мгновенно предоставлять общий доступ к своему проекту с поддержкой высокого уровня безопасности. Кроме того, вы можете предоставлять общий доступ к сеансам, экземплярам терминала, веб-приложениям на локальном компьютере, голосовым звонкам и т. п.

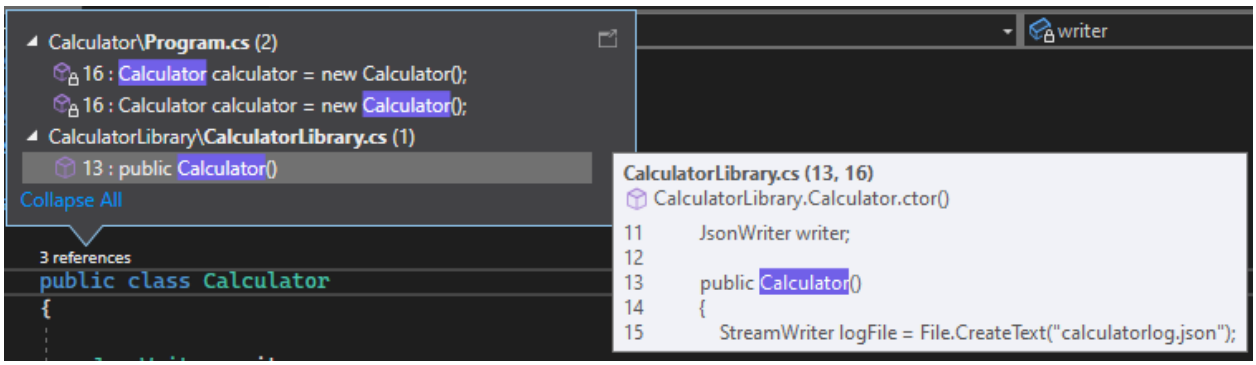
- Иерархия вызовов

  В окне **Иерархия вызовов** показаны методы, вызывающие выбранный метод. Это может быть полезно, если вы собираетесь изменить либо удалить метод или хотите отследить ошибку.



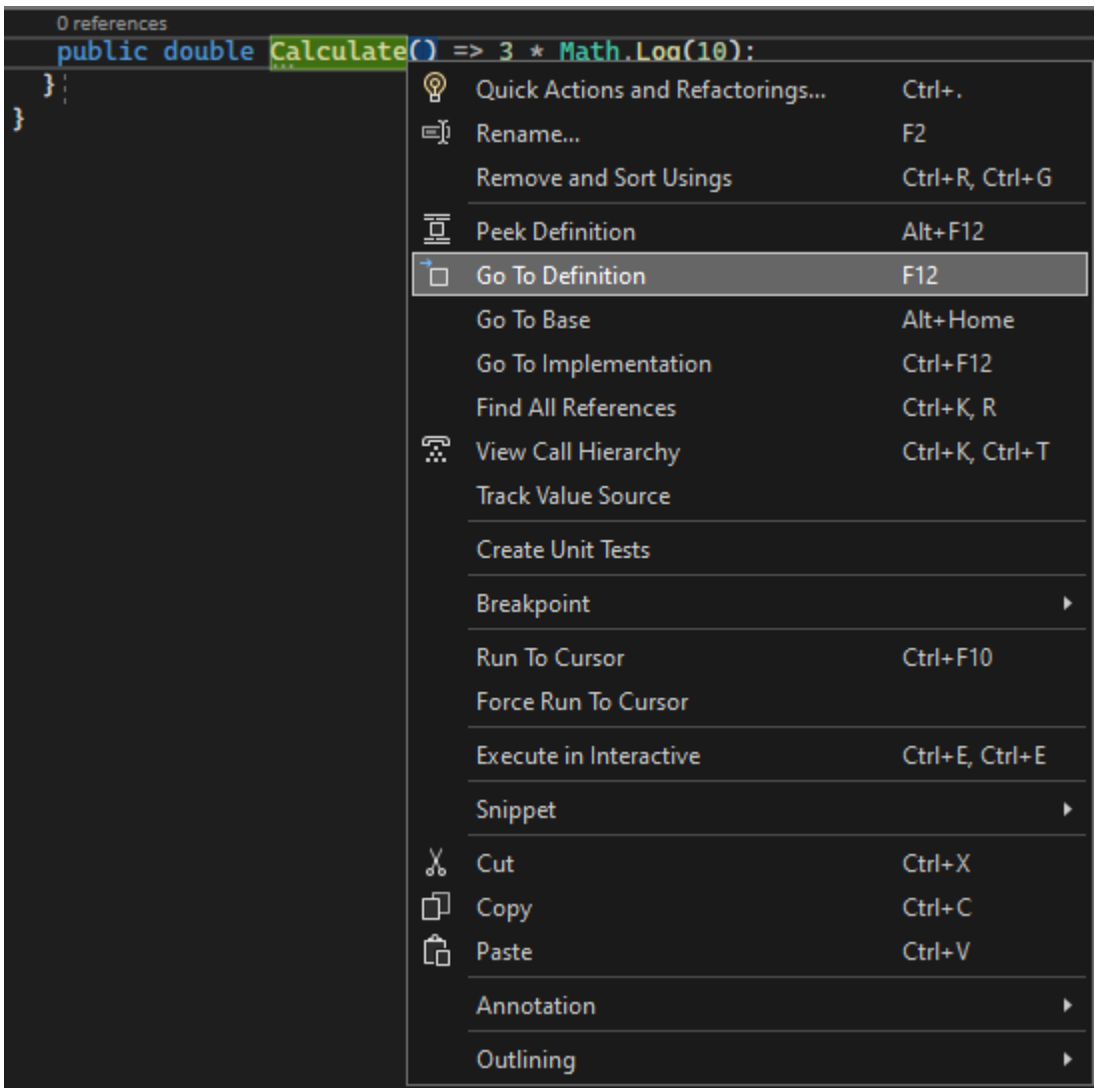
- CodeLens

  CodeLens помогает находить ссылки на код, изменения кода, связанные с кодом ошибки, рабочие элементы, проверки кода и модульные тесты — не выходя из редактора.



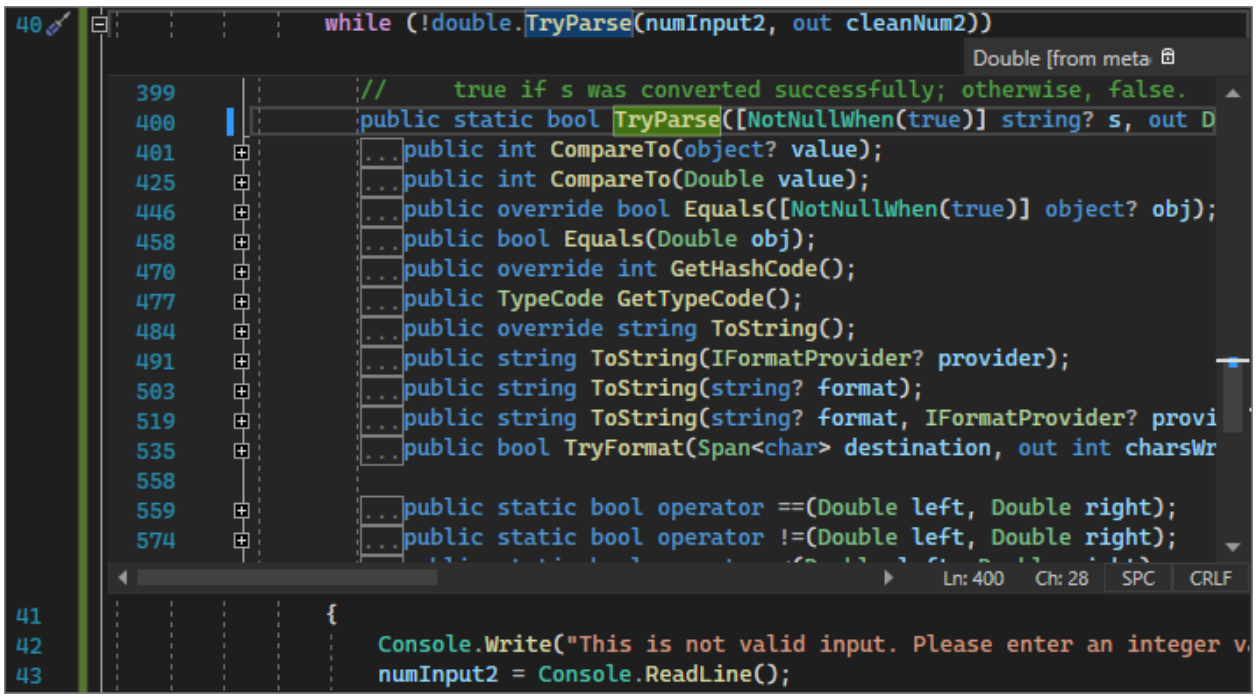
- Перейти к определению

  Функция **Перейти к определению** позволяет перейти к расположению, где определена выбранная функция или тип.



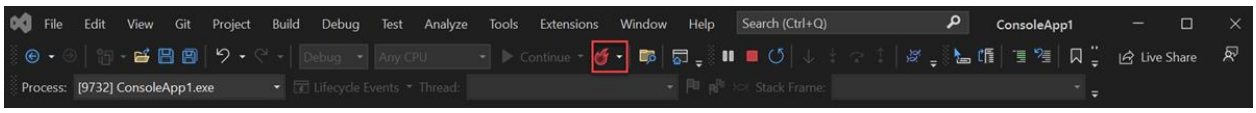
- Показать определение

  В окне **Показать определение** можно отобразить метод или определение типа, не открывая отдельный файл.



- Горячая перезагрузка

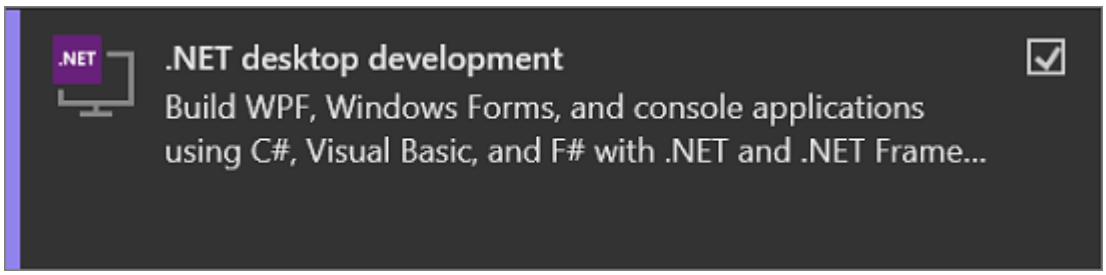
  Горячая перезагрузка позволяет изменять файлы кода приложения и сразу же применять изменения кода к работающему приложению.



Установите Visual Studio.

В рамках этого раздела вы создаете простой проект для тестирования некоторых возможностей Visual Studio. IntelliSense используется как вспомогательный инструмент для программирования и отладки приложения, который позволяет просмотреть значение переменной во время выполнения приложения или изменить цветовую тему.

Чтобы начать работу, скачайте и установите Visual Studio. В модульном установщике вы можете выбрать и установить *рабочие нагрузки*, которые являются группами функций, необходимыми для предпочитаемого языка программирования или платформы. Чтобы выполнить следующую процедуру создания программы, обязательно выберите во время установки рабочую нагрузку **Разработка классических приложений .NET**.

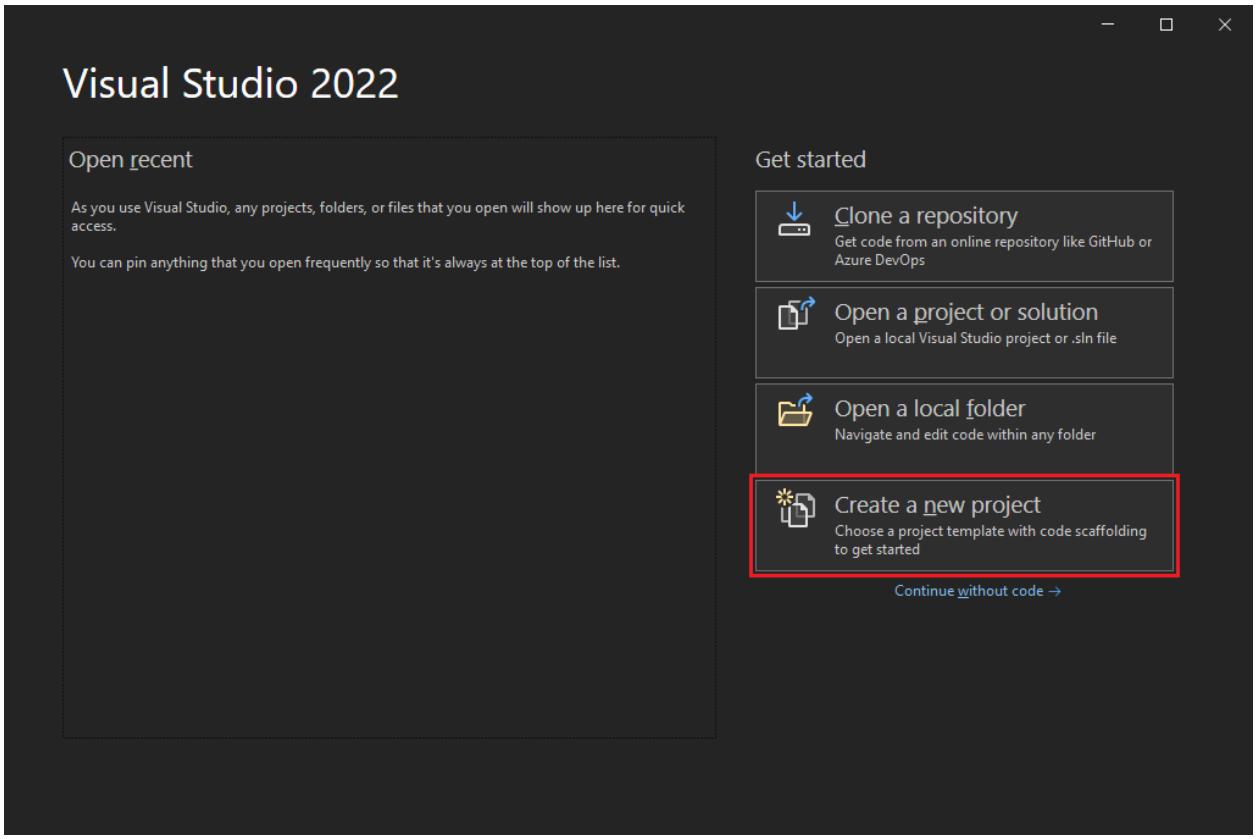


При первом запуске Visual Studio можно выполнить вход с использованием учетной записи Майкрософт или рабочей учетной записи.

Создание программы

Давайте создадим простую программу.

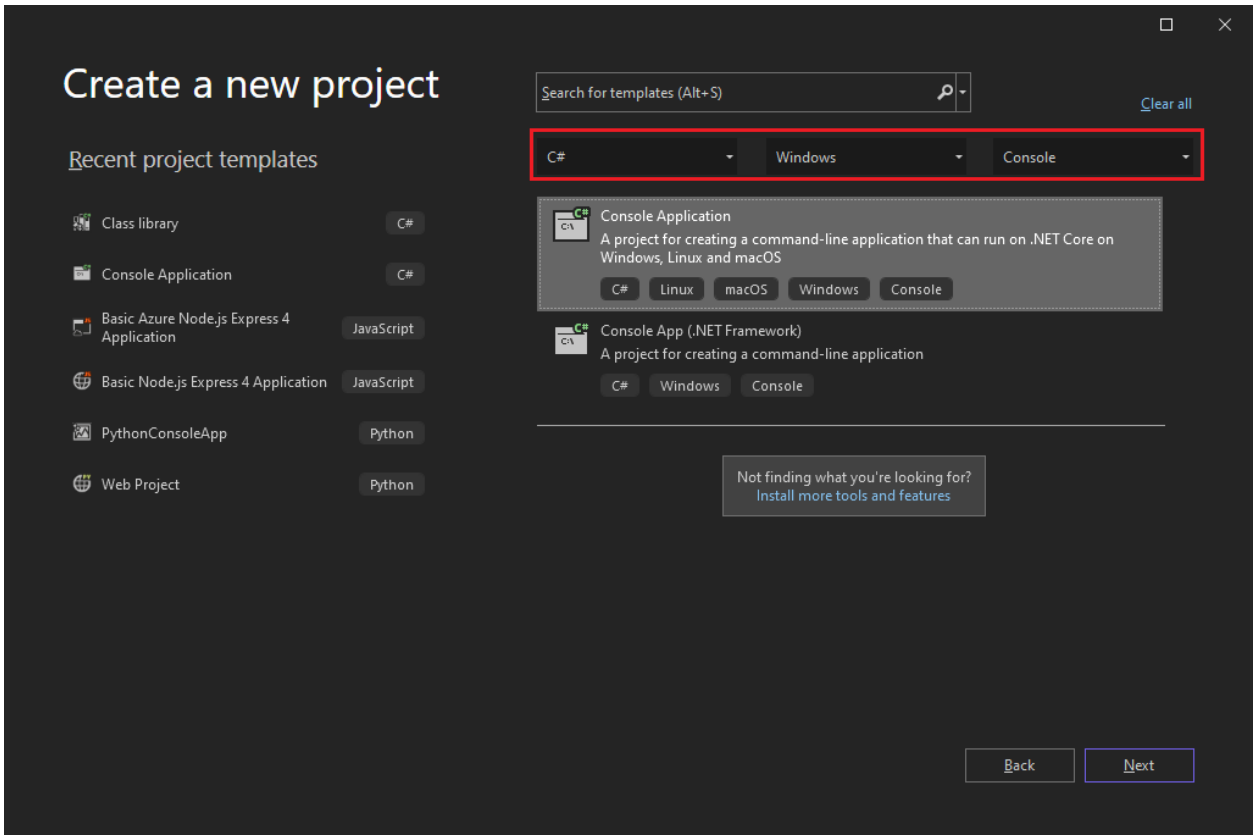
1. Запустите среду Visual Studio. Откроется начальное окно, где можно клонировать репозиторий, открыть недавно использованный проект или создать новый.
2. Выберите **Создать проект**.



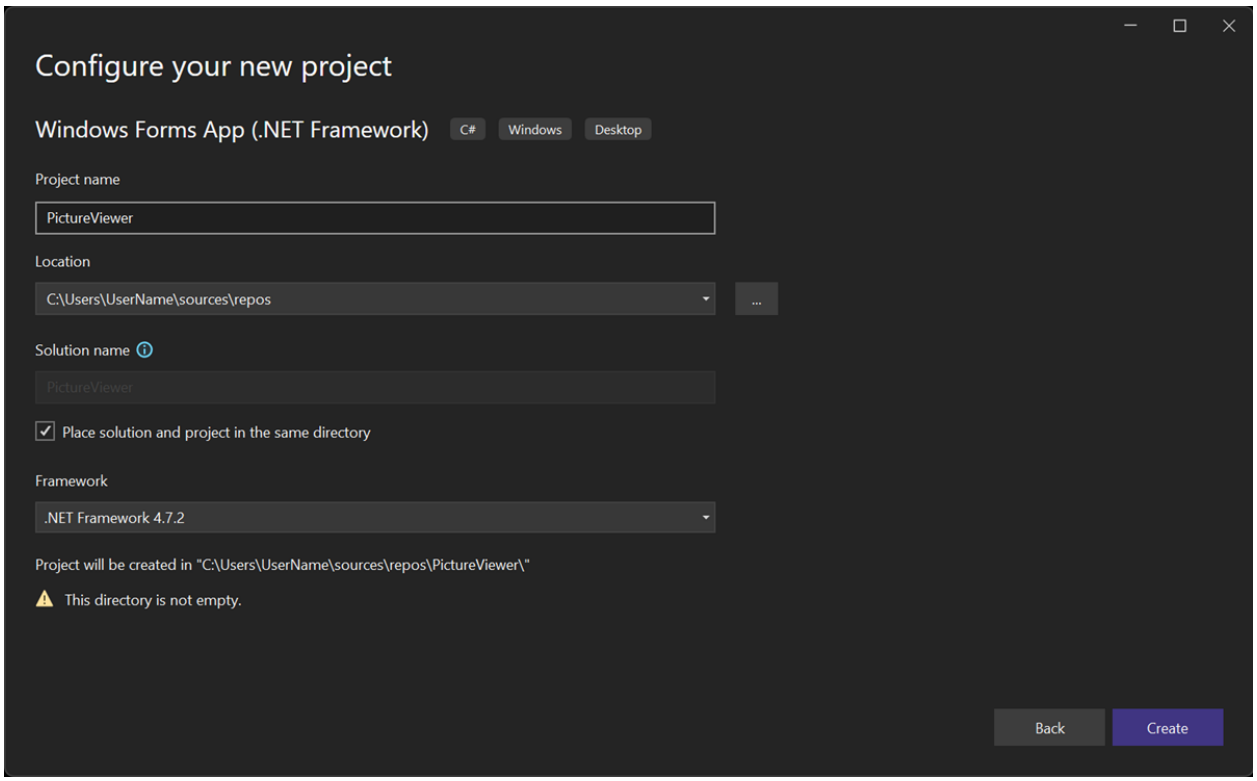
Откроется окно **Создание проекта** с отображением нескольких *шаблонов* проектов. Шаблон содержит основные файлы и параметры, которые требуются для определенного типа проекта.

3. Чтобы найти шаблон, попробуйте ввести или ввести ключевые слова в поле поиска. Список доступных шаблонов будет фильтроваться по введенным ключевым словам. Вы можете дополнительно отфильтровать результаты шаблона, выбрав **C#** в раскрывающемся списке **Все языки**, **Windows** в списке **Все платформы** и **Консоль** в списке **Все типы проектов**.

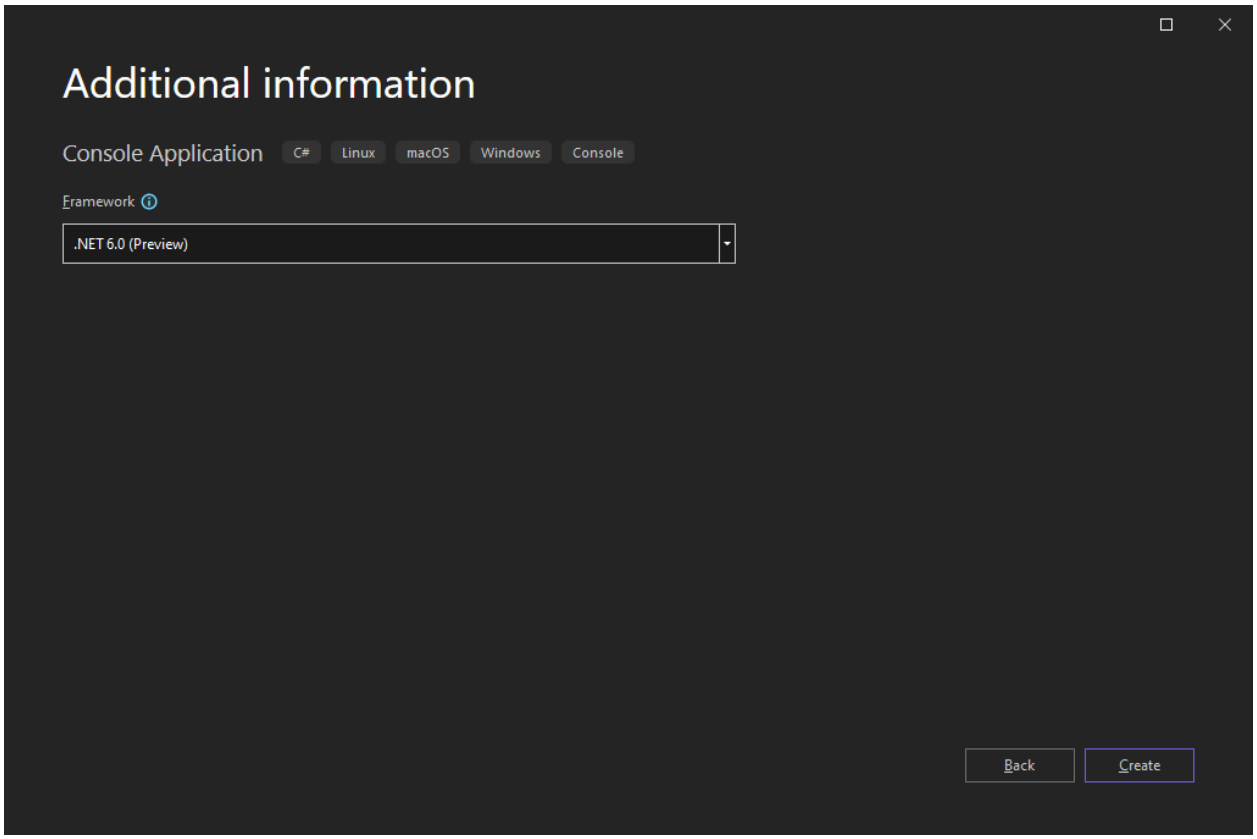
   Выберите шаблон **Консольное приложение** и нажмите кнопку **Далее**.



4. В поле **Имя проекта** окна **Настроить новый проект** введите **HelloWorld**. При необходимости измените расположение каталога проекта в расположении по умолчанию *C:\Users\<имя>\source\repos*, а затем нажмите кнопку **Далее**.

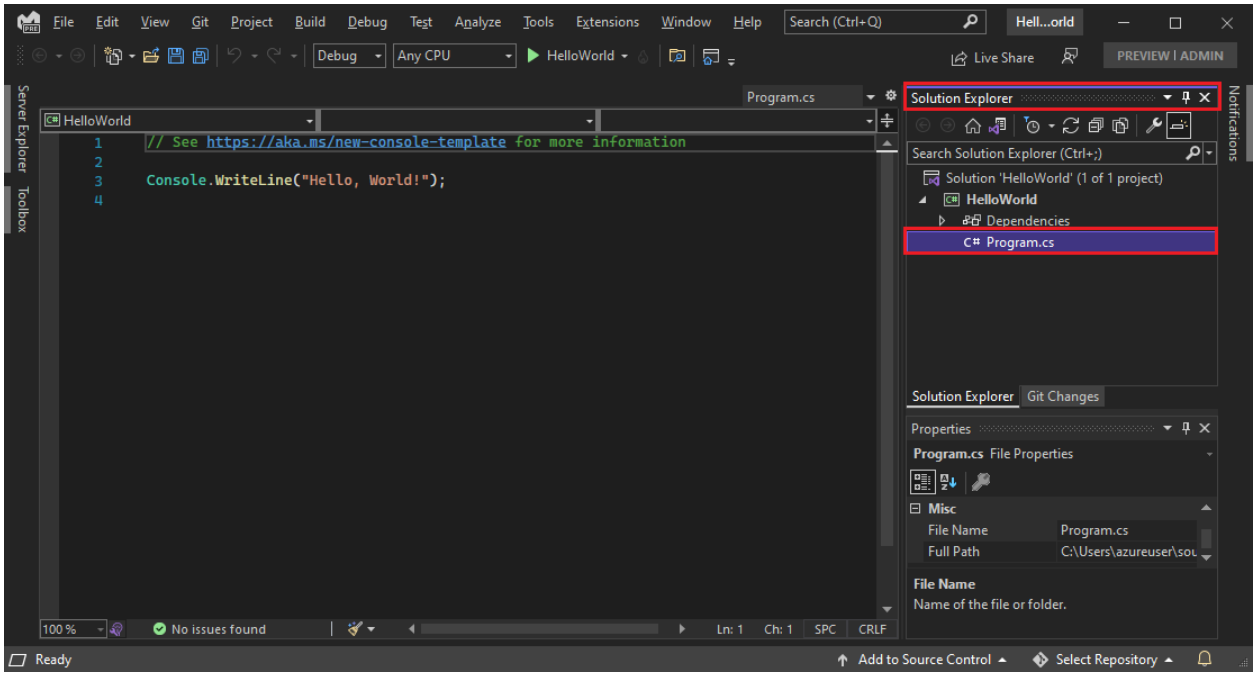


5. Убедитесь, что в окне **Дополнительные сведения** в раскрывающемся меню **Целевая платформа** указано **.NET 6.0**, а затем щелкните **Создать**.



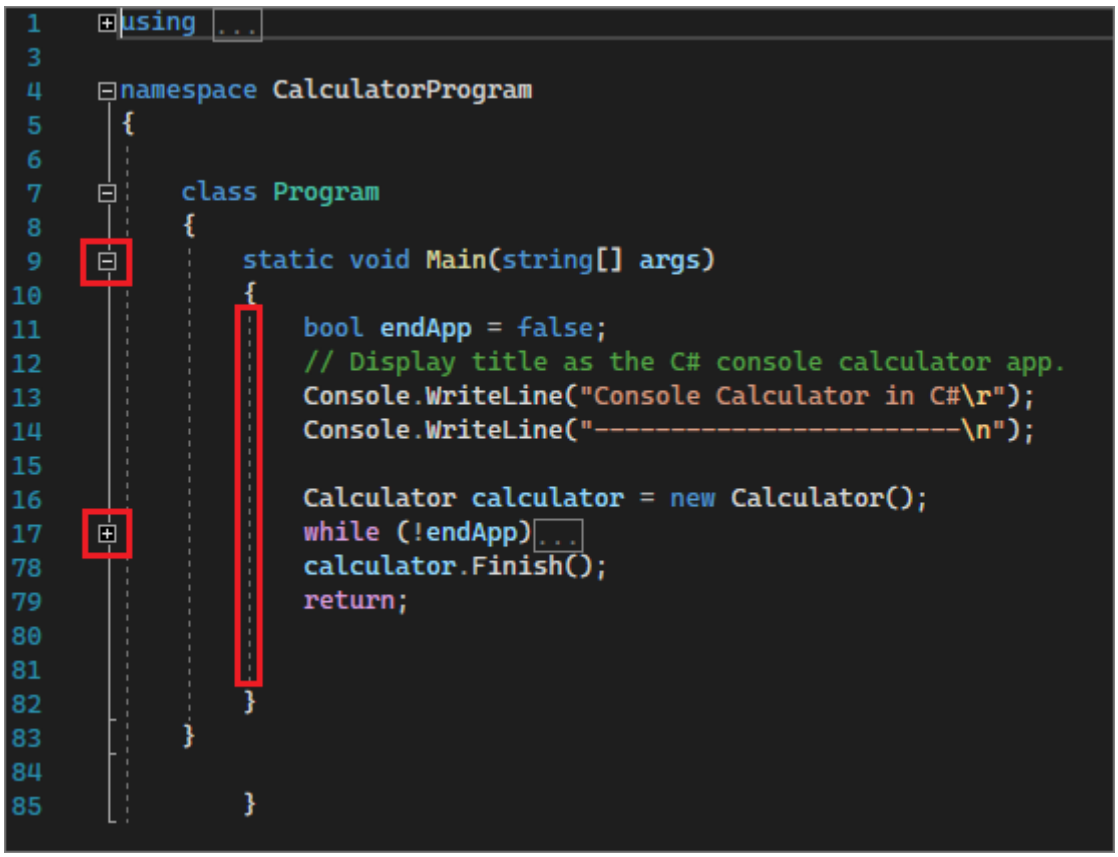
Visual Studio создаст проект. Это простейший вариант приложения Hello World, в котором вызывается метод Console.WriteLine() для вывода строки **Hello World!** в окне консоли.

Файлы проекта отображаются справа в окне интегрированной среды разработки Visual Studio в окне с названием **Обозреватель решений**. В окне **Обозреватель решений** выберите файл **Program.cs**. Код C# для вашего приложения открывается в центральном окне редактора, который занимает большую часть пространства.



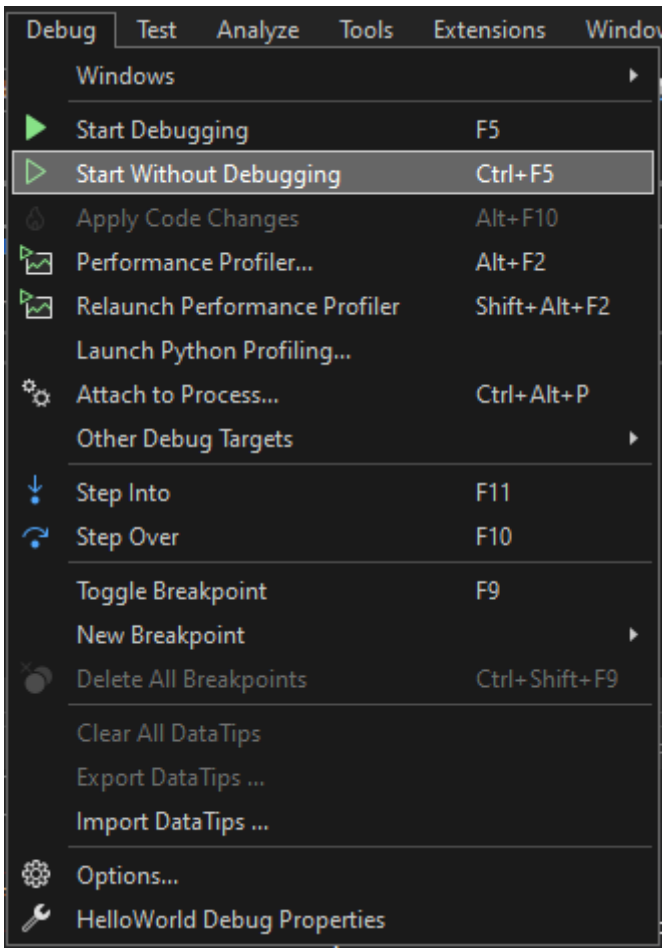
Код автоматически выделяется цветом для обозначения таких элементов, как ключевые слова и типы. Найти код можно по номерам строк.

Небольшие вертикальные пунктирные линии в коде указывают, какие фигурные скобки соответствуют друг другу. Чтобы свернуть или развернуть блоки кода, используйте небольшие рамки со знаками минус и плюс соответственно. Эта функция структурирования кода позволяет скрыть ненужный код на экране.

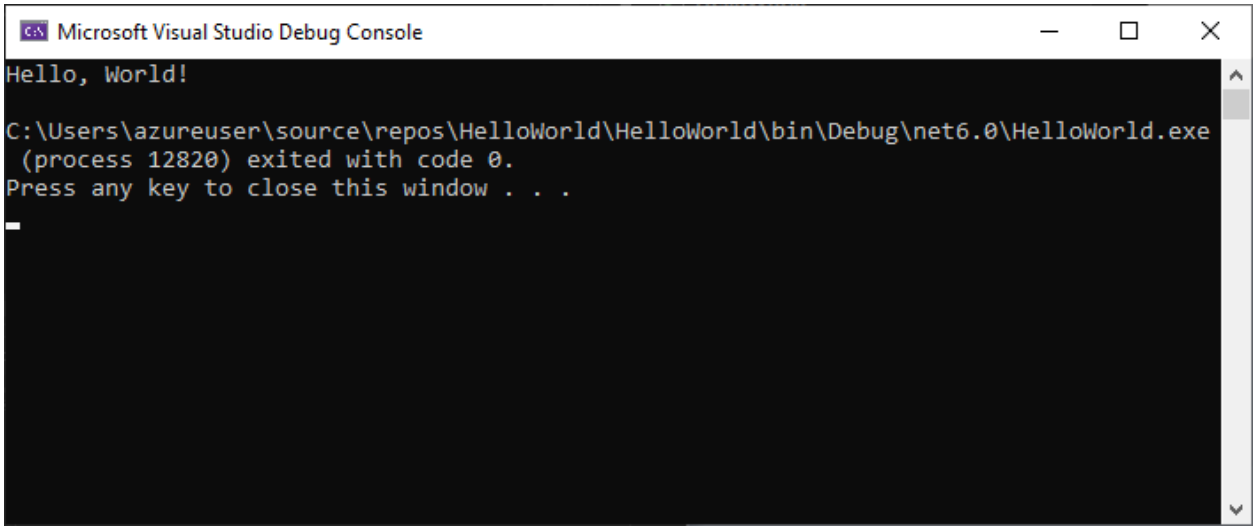


Также доступно множество других меню и окон инструментов.

6. Запустите приложение, выбрав в главном меню Visual Studio пункты **Отладка**>**Запуск без отладки**. Можно также нажать клавиши **CTRL**+**F5**.



Когда Visual Studio создаст приложение, откроется окно консоли с сообщением **Hello, World!**. Теперь у вас есть выполняемое приложение.



7. Для закрытия окна консоли нажмите любую клавишу.
8. Давайте добавим новый год в это приложение. Перед строкой `Console.WriteLine("Hello World!");` добавьте следующий код C#:

C#Копировать

```
Console.WriteLine("\nWhat is your name?");
var name = Console.ReadLine();
```

Этот код позволяет отобразить сообщение **What is your name?** (Введите имя) в окне консоли и ожидает, чтобы пользователь ввел текст.

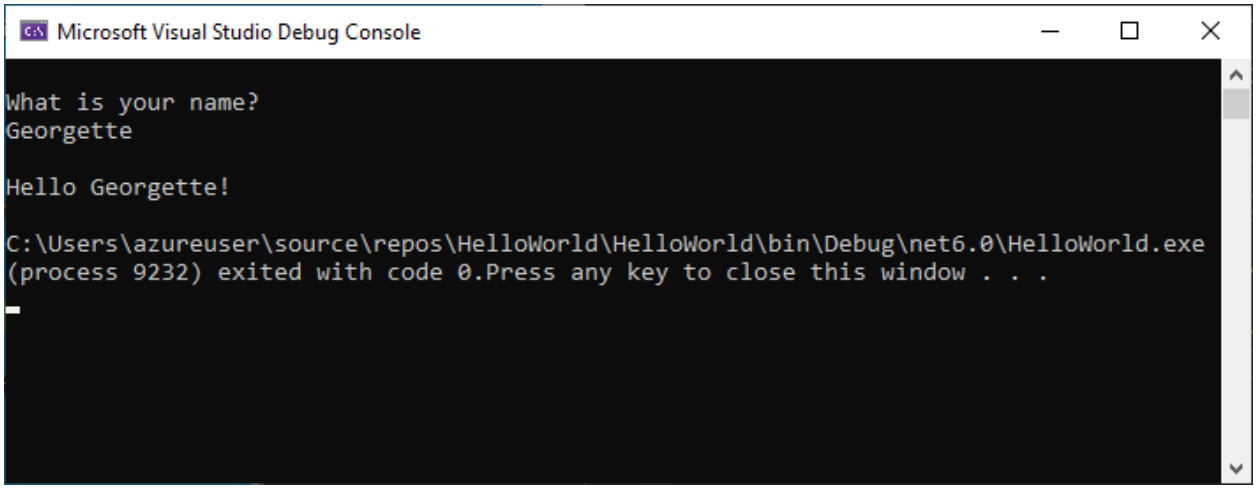
9. Измените строку с текстом `Console.WriteLine("Hello World!");` на следующую:

   C#Копировать
   ```
   Console.WriteLine($"\nHello {name}!");
   ```

10. Снова запустите приложение, выбрав пункты **Отладка** >**Запуск без отладки** или нажав клавиши **CTRL**+**F5**.

    Visual Studio выполнит повторную сборку приложения. В открывшемся окне консоли отобразится запрос на ввод имени.

11. Введите свое имя в окне консоли и нажмите клавишу **ВВОД**.



12. Нажмите любую клавишу, чтобы закрыть окно консоли и остановить выполняющуюся программу.

Использование рефакторинга и IntelliSense

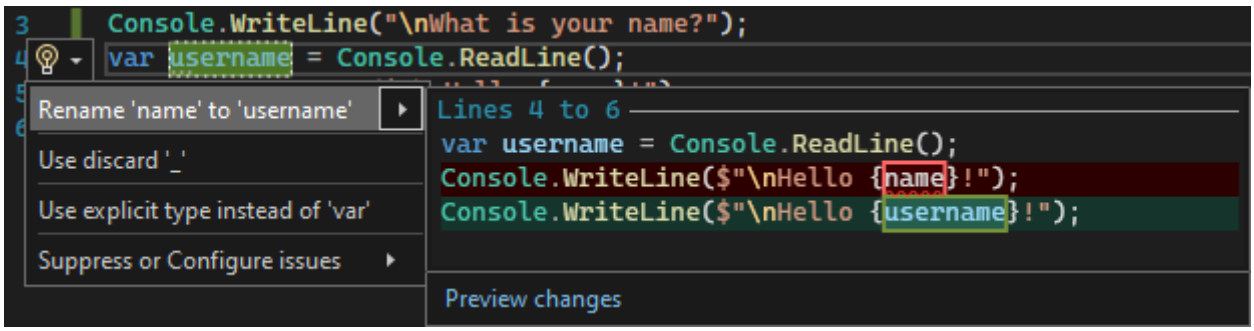
Рассмотрим несколько примеров того, как рефакторинг и IntelliSense помогают повысить эффективность кода.

Сначала переименуйте переменную `name`:

1. Дважды щелкните переменную `name` и введите для нее новое имя: *username*.

   Вокруг переменной появится прямоугольник, а в поле появится значок лампочки.

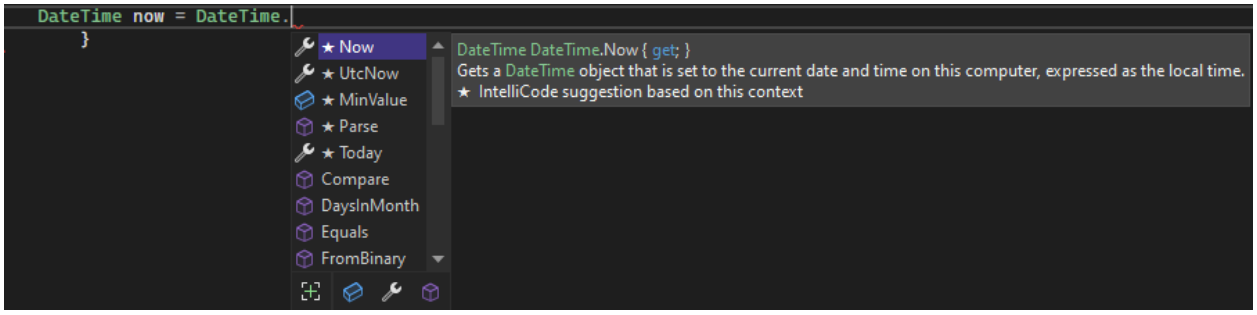
2. Выберите значок лампочки для отображения доступных быстрых действий. Выберите **Переименовать name в username**.



Переменная переименовывается во всем проекте, то есть в нашем случае только в двух местах.

3. Теперь рассмотрим возможности IntelliSense. Под строкой `Console.WriteLine($"\nHello {username}!");` введите `DateTime now = DateTime.`.

   Появится поле с членами класса DateTime. В отдельном поле отобразится описание выбранного элемента.



4. Выберите член с именем **Now**(свойство), который является свойством класса. Для этого дважды щелкните его или нажмите клавишу **Tab**. Завершите строку кода, добавив точку с запятой в конец строки: `DateTime now = DateTime.Now;`.
5. Под этой строкой добавьте следующий фрагмент кода:

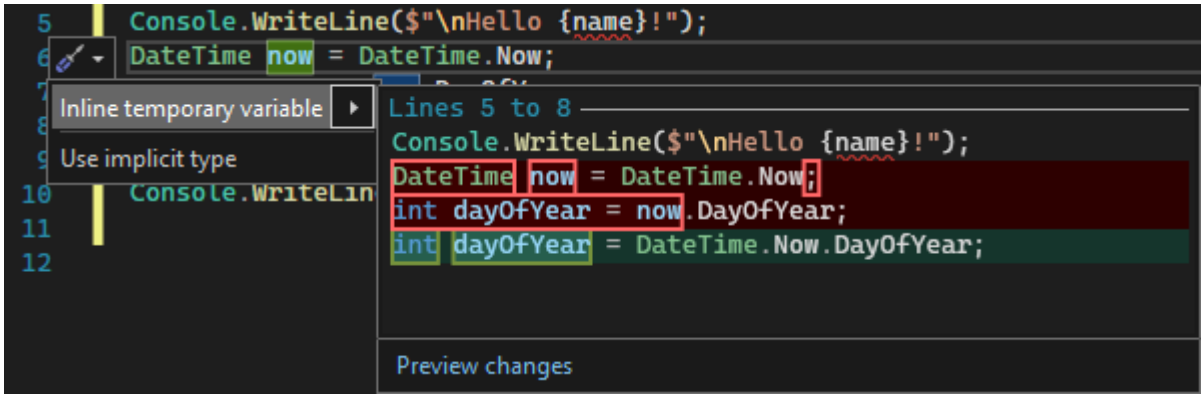
   C#Копировать
   ```
   int dayOfYear = now.DayOfYear;

   Console.Write("Day of year: ");
   Console.WriteLine(dayOfYear);
   ```

   Совет

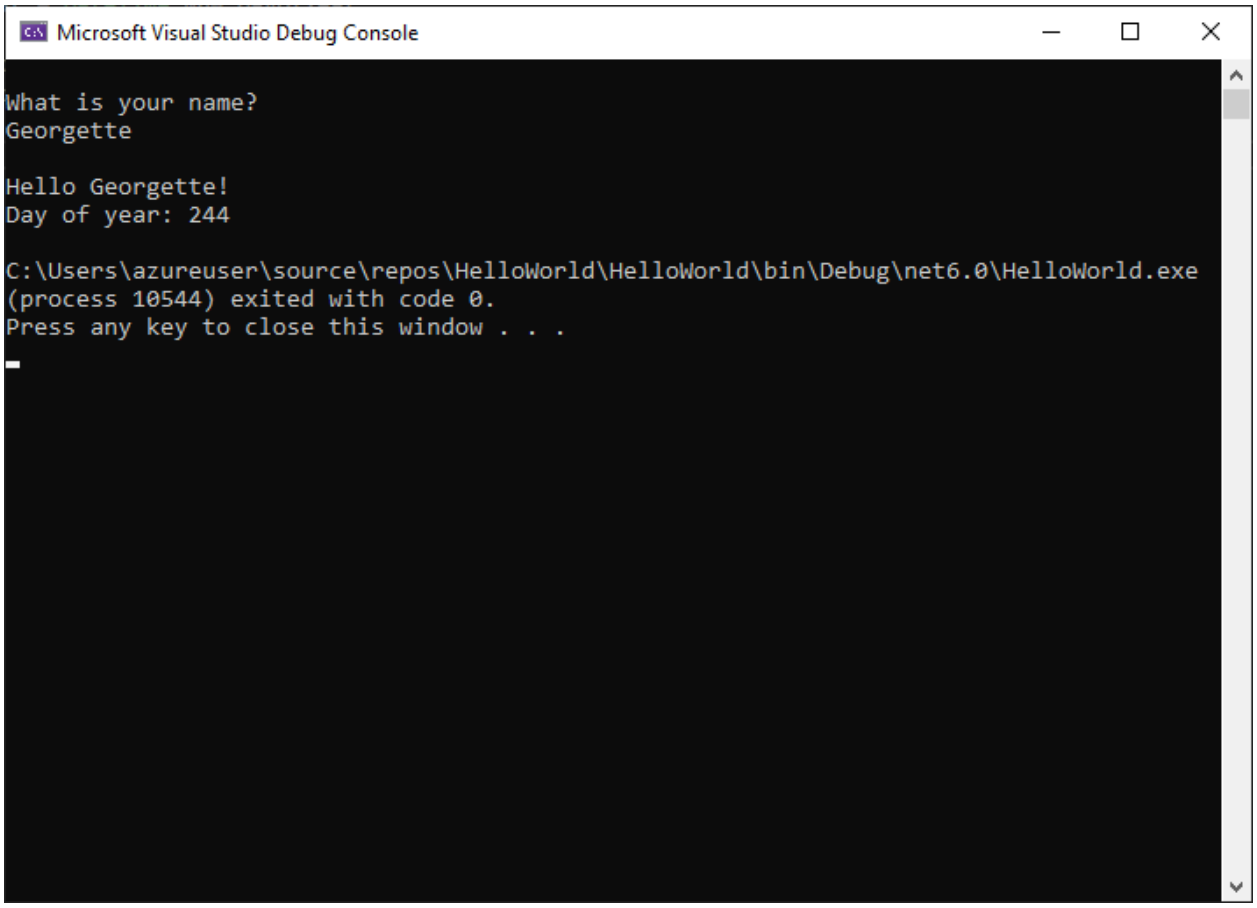
   Код **Console.Write** отличается от **Console.WriteLine** тем, что не добавляет знак завершения строки после ее вывода. Это означает, что следующий фрагмент текста, отправляемый на вывод, будет выводиться в той же строке. Можно навести указатель мыши на каждый из этих методов в коде, чтобы просмотреть его описание.

6. Далее мы снова применим рефакторинг, чтобы сократить код. Выберите переменную `now` в строке `DateTime now = DateTime.Now;`. На поле в этой строке появится значок отвертки.

7. Щелкните этот значок отвертки, чтобы просмотреть доступные предложения Visual Studio. В этом случае отображается рефакторинг Встроенная временная переменная для удаления строки кода без изменения его общего поведения.



8. Щелкните пункт **Встроенная временная переменная**, чтобы выполнить рефакторинг кода.
9. Снова запустите программу, нажав клавиши **Ctrl**+**F5**. Выходные данные выглядят примерно так:



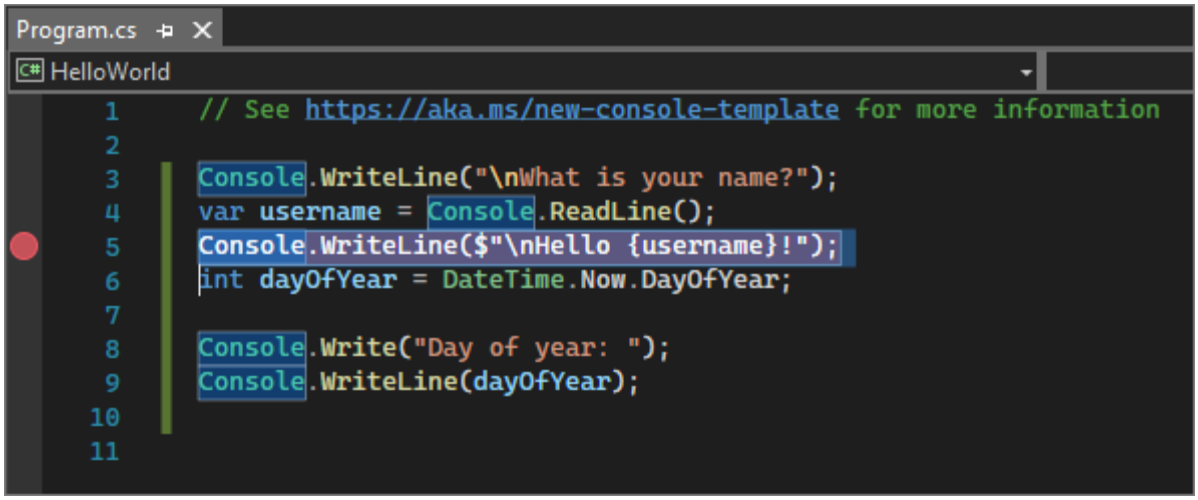
Отладка кода

При написании кода его следует регулярно запускать и проверять на предмет ошибок. Система отладки Visual Studio позволяет просматривать код с шагом в одну инструкцию, проверяя значения переменных. Вы можете задать *точки останова*, которые позволяют приостановить выполнение кода в определенной строке и увидеть, как изменяется значение переменной при выполнении кода.

Зададим точку останова, чтобы во время выполнения программы отобразилось значение переменной `username`.

1. Установите точку останова в строке с кодом `Console.WriteLine($"\nHello {username}!");`, щелкнув крайнее поле слева (область навигации) в этой строке. Кроме того, вы можете выбрать строку кода и нажать клавишу **F9**.

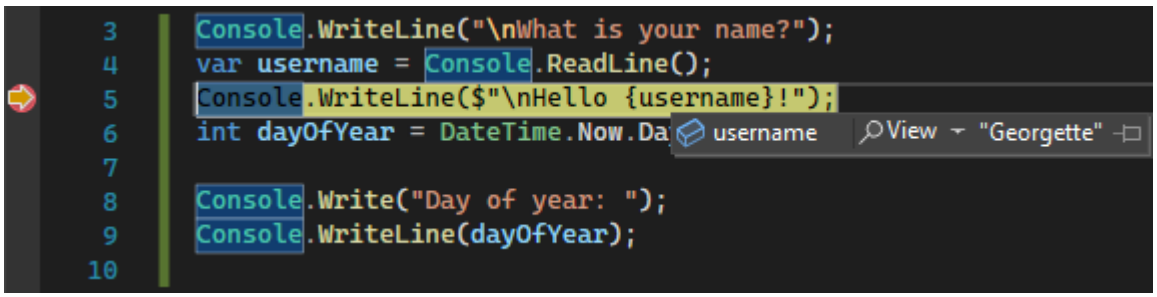
   В области навигации появится красный кружок, и эта строка будет выделена.



2. Начните отладку, выбрав пункты **Отладка**>**Начать отладку** или нажав клавишу **F5**.
3. Когда появится окно консоли с запросом имени, введите свое имя.

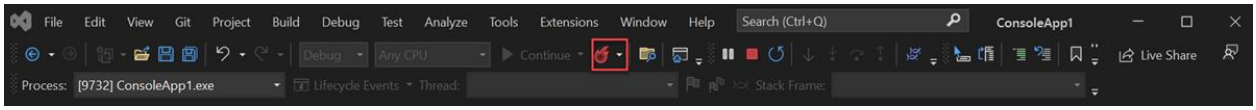
   Фокус возвратится в редактор кода Visual Studio, и строка кода с точкой останова будет выделена желтым цветом. Желтый цвет означает, что эта строка кода будет выполняться следующей. Точка останова отвечает за приостановку работы приложения на этой строке.

4. Наведите указатель мыши на переменную `username` для просмотра ее значения. Кроме того, вы можете щелкнуть `username` правой кнопкой мыши и выбрать пункт **Добавить контрольное значение**, чтобы добавить переменную в окно **контрольных значений**, где можно будет просмотреть ее значение.



5. Нажмите клавишу **F5** еще раз, чтобы завершить работу приложения.

После запуска приложения можно применить изменения кода к работающему приложению, нажав кнопку "Горячая перезагрузка".



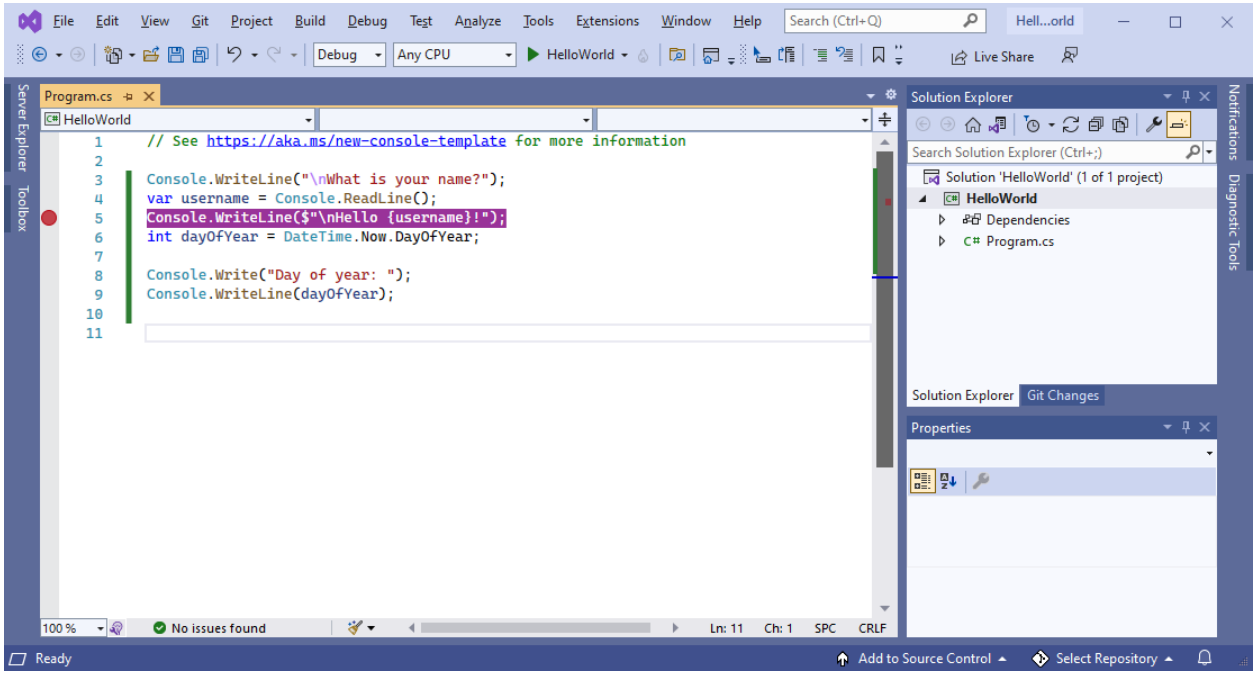
Дополнительные сведения об отладке в Visual Studio см. в статье Знакомство с отладчиком Visual Studio.

Настройка Visual Studio

Вы можете настроить пользовательский интерфейс Visual Studio, в том числе изменить цветовую тему, установленную по умолчанию. Изменение цветовой темы

1. В строке меню выберите **Сервис**>**Параметры**, чтобы открыть диалоговое окно **Параметры**.
2. На странице параметров **Среда** >**Общие** измените значение параметра **Цветовая тема** на **Синий** или **Светлый**. Затем нажмите кнопку **ОК**.

   Цветовая тема для всей интегрированной среды разработки соответствующим образом изменится. На следующем снимке экрана показана синяя цветовая тема:



Дополнительные сведения о других способах персонализации интегрированной среды разработки см. в разделе Персонализация Visual Studio.

Выбор параметров среды

Вы можете настроить Visual Studio для использования параметров среды, предназначенных для разработчиков на C#.

1. В строке меню выберите **Сервис**>**Импорт и экспорт параметров**.
2. В **мастере импорта и экспорта параметров** выберите **Сбросить все параметры**, а затем нажмите кнопку **Далее**.
3. На странице **Сохранить текущие параметры** выберите, следует ли сохранить текущие параметры перед сбросом. Если вы не изменяли параметры, выберите **Нет, только сбросить параметры, перезаписав мои текущие значения**. Затем выберите **Далее**.
4. На странице **Выбор набора параметров, используемого по умолчанию** выберите **Visual C#**, а затем нажмите кнопку **Готово**.
5. На странице **Сброс завершен** нажмите **Закрыть**.