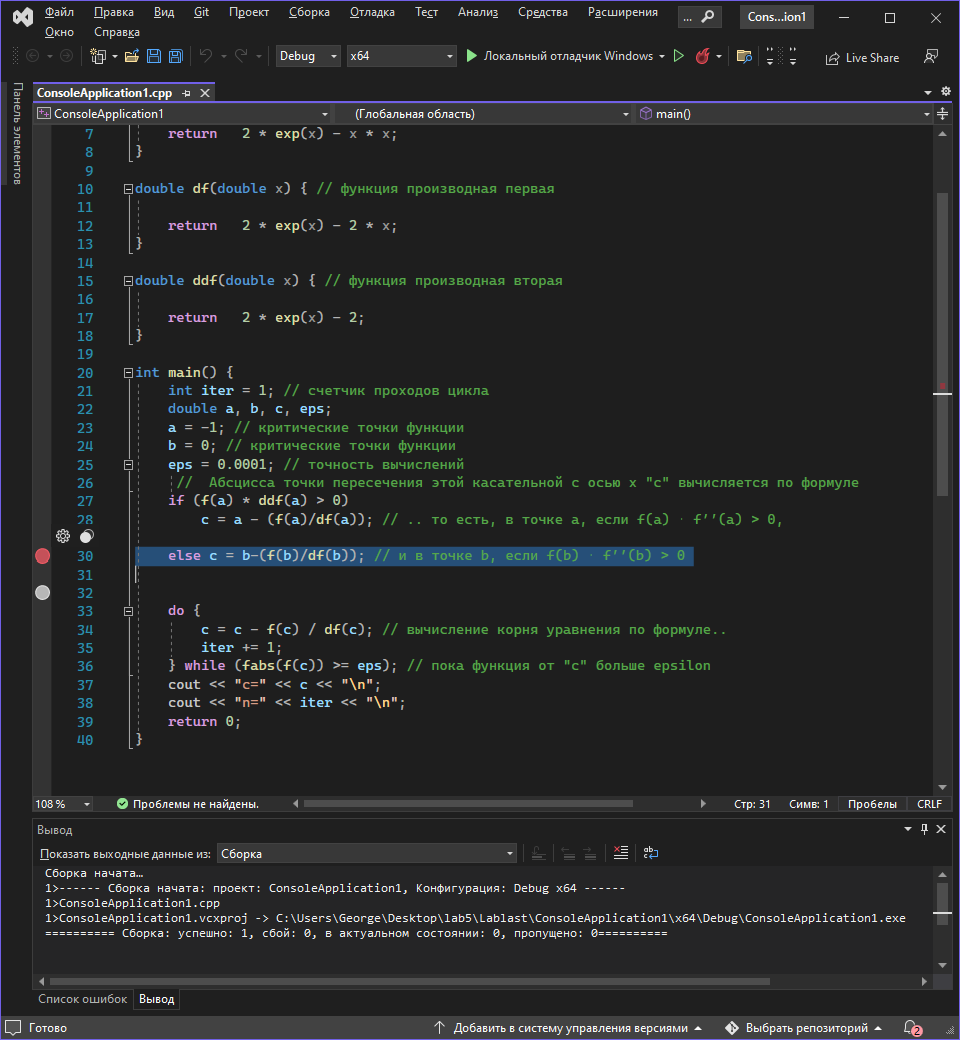
## Отладка в Visual Studio

В контексте Visual Studio отладка приложения обычно означает запуск приложения с подключенным отладчиком (то есть в режиме отладчика). 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 и т. д.

### Установка точки останова и запуск отладчика

**Точка останова** указывает, где Visual Studio следует приостановить выполнение кода, чтобы вы могли проверить значения переменных или поведение памяти либо выполнение ветви кода.

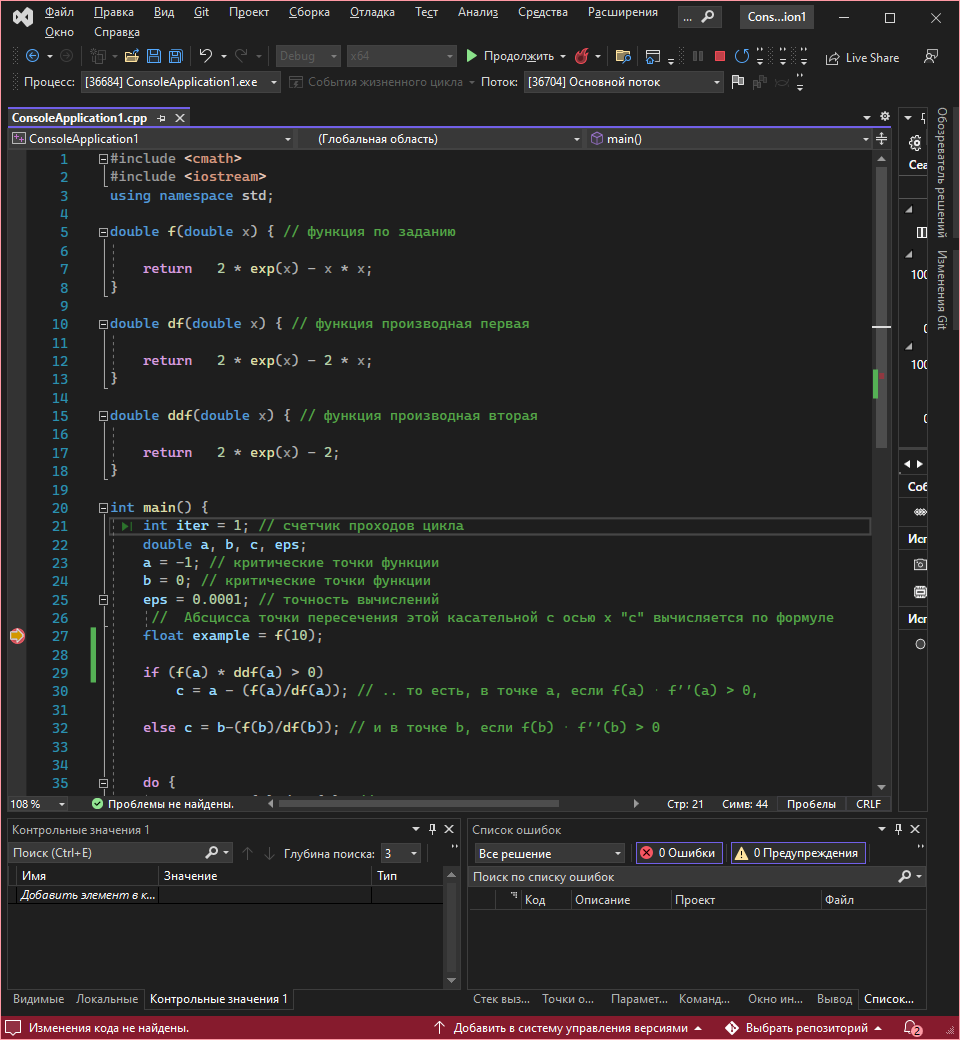
Если вы открыли файл в редакторе кода, точку останова можно задать, щелкнув в поле слева от строки кода.



Нажмите клавишу **F5 (Отладка > Начать отладку)** или кнопку **Начать отладку** на панели инструментов отладки. При этом отладчик выполняется до первой встреченной точки останова.

### Переход по коду в отладчике с помощью пошаговых команд

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



F11 — это команда **Шаг с заходом**, которая выполняет приложение с переходом к следующему оператору (с заходом в функцию).

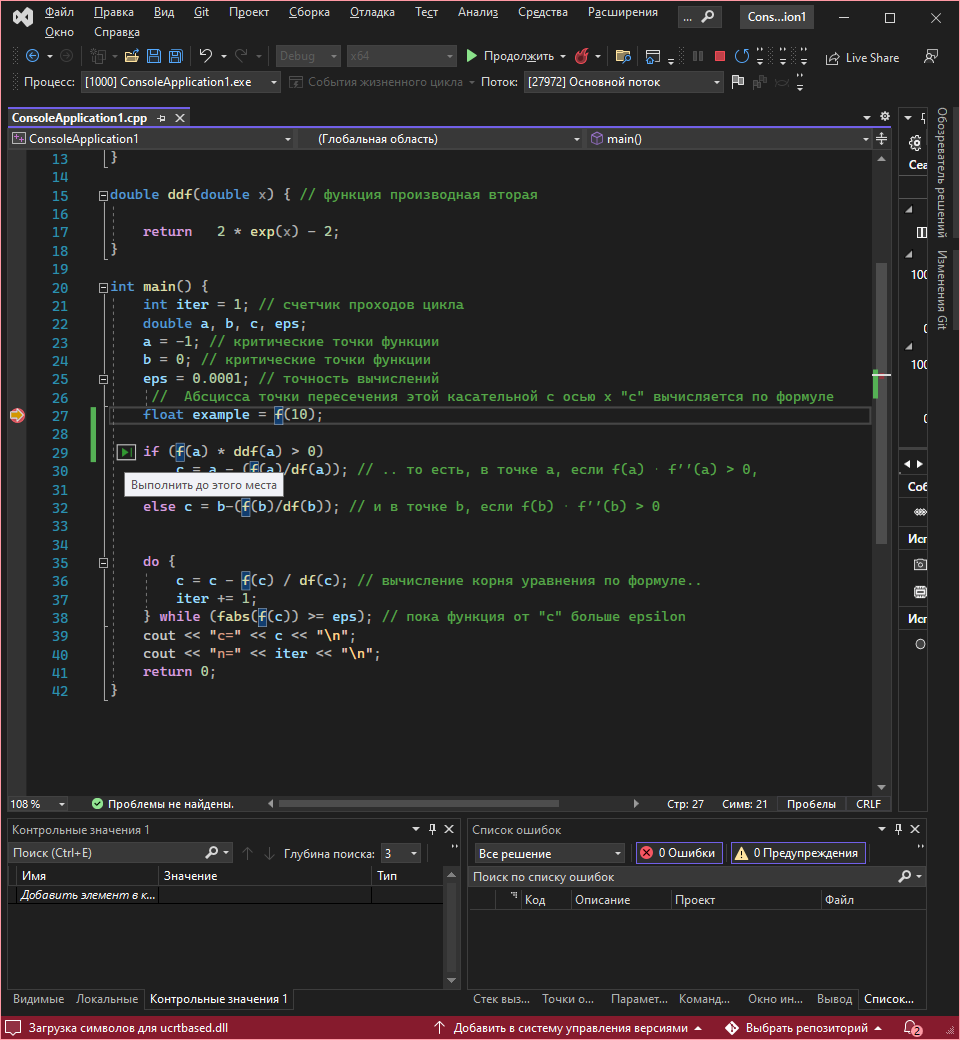
Когда вы находитесь в строке кода, представляющей собой вызов функции или метода, можно нажать **клавишу F10 (Отладка > Шаг с обходом)** вместо F11.

Клавиша F10 продолжает выполнение отладчика без захода в функции или методы в коде приложения (код продолжает выполняться). Нажав клавишу F10, вы можете обойти код, который вас не интересует.

**Шаг с выходом (Shift+F11)** позволяет возобновить выполнение приложения (и перемещает отладчик) до возврата текущей функции (выводит отладчик назад из текущей функции).

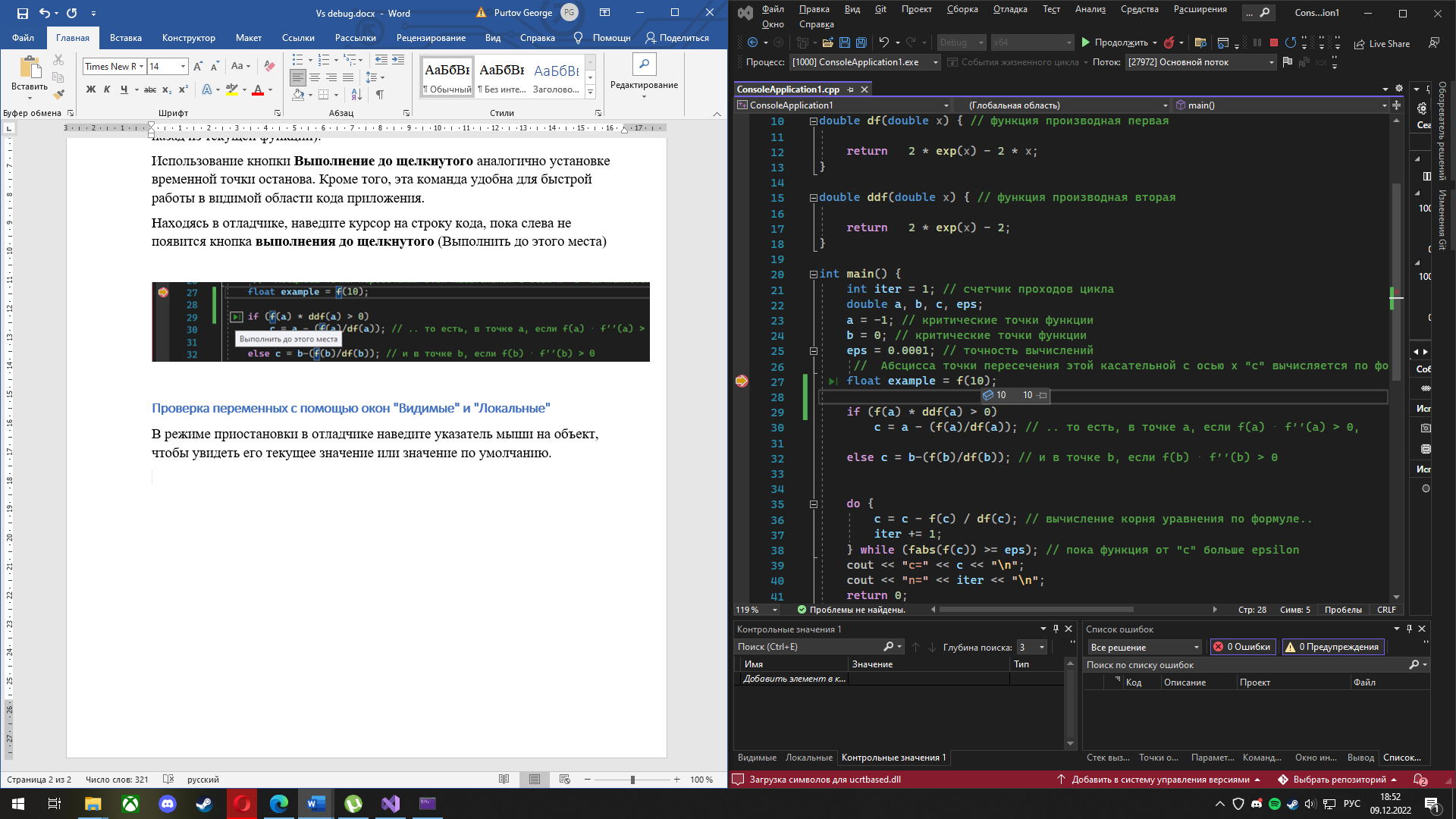
Использование кнопки **Выполнение до щелкнутого** аналогично установке временной точки останова. Кроме того, эта команда удобна для быстрой работы в видимой области кода приложения.

Находясь в отладчике, наведите курсор на строку кода, пока слева не появится кнопка **выполнения до щелкнутого** (Выполнить до этого места)

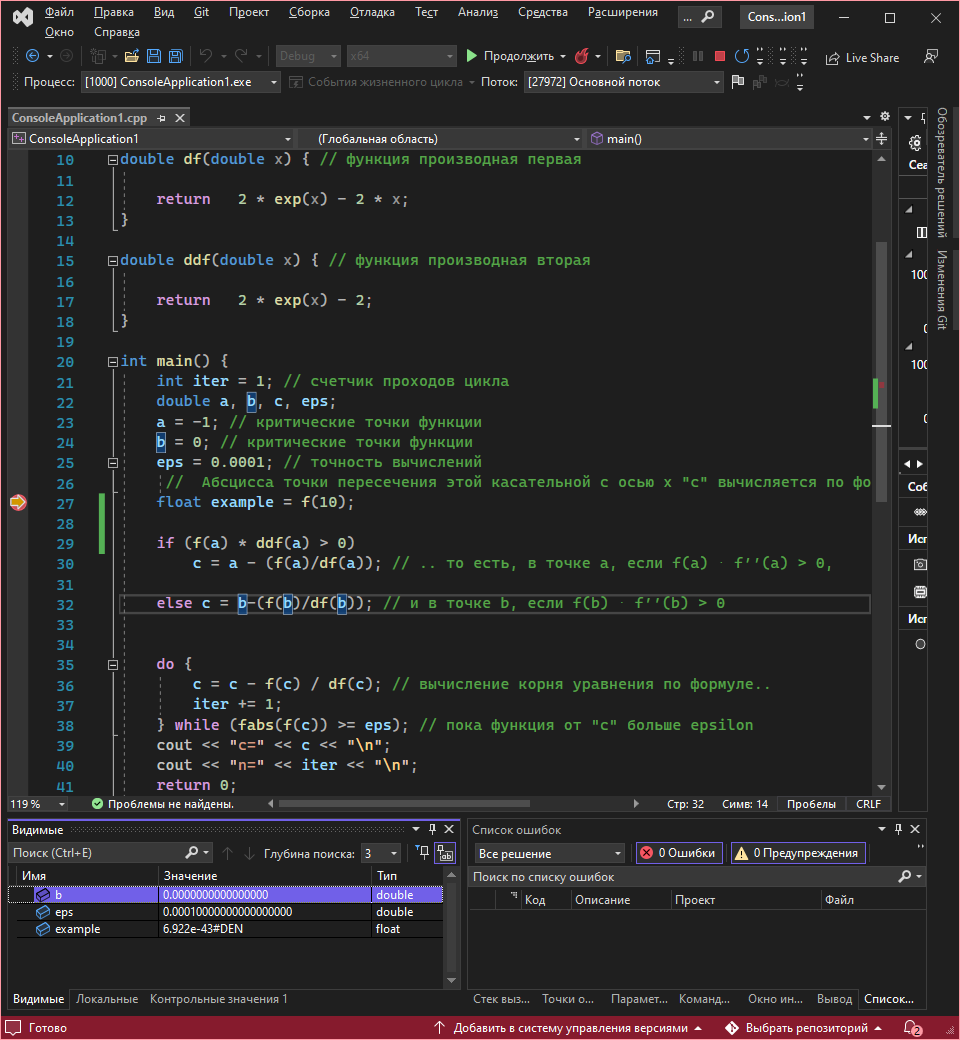


### Проверка переменных с помощью окон "Видимые" и "Локальные"

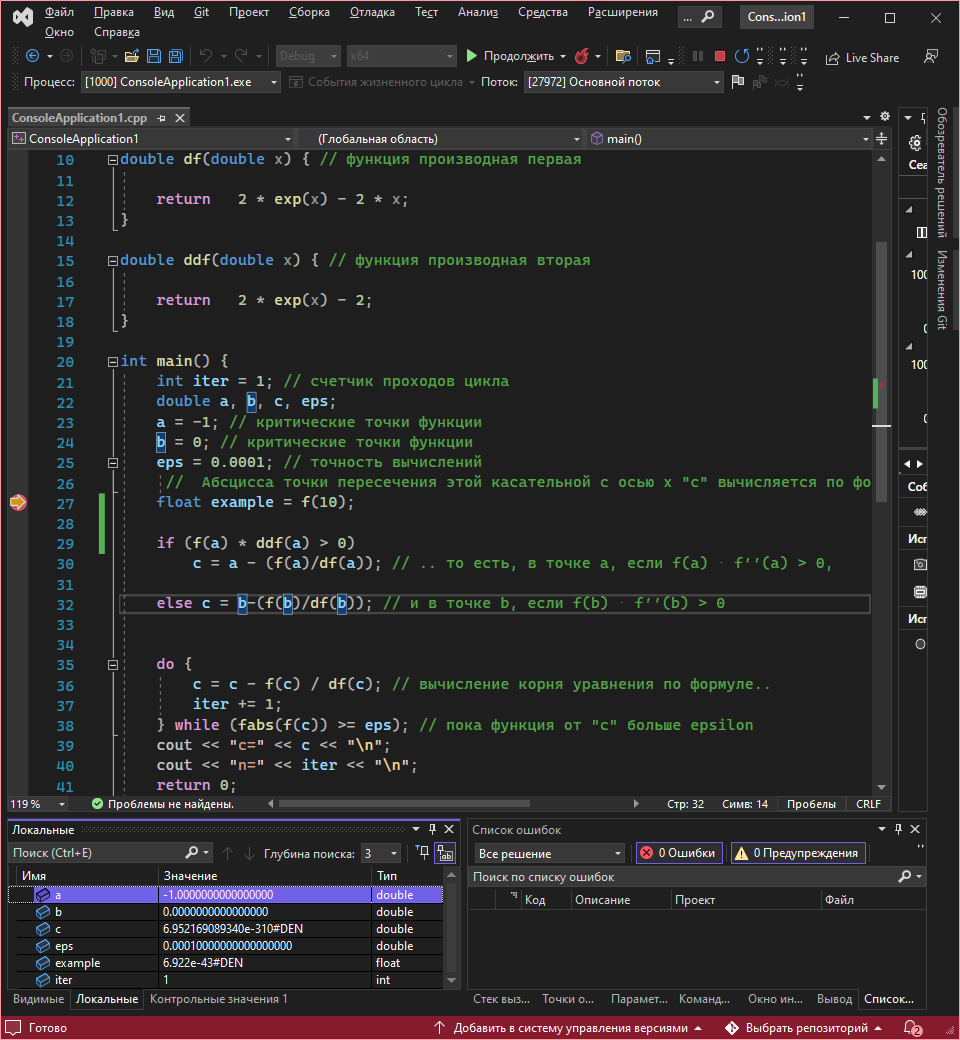
В режиме приостановки в отладчике наведите указатель мыши на объект, чтобы увидеть его текущее значение или значение по умолчанию.



В окне **Видимые** отображаются переменные вместе с текущим значением и типом. Окно **Видимые** показывает все переменные, используемые в текущей и в трех предыдущих строках кода (для C++).

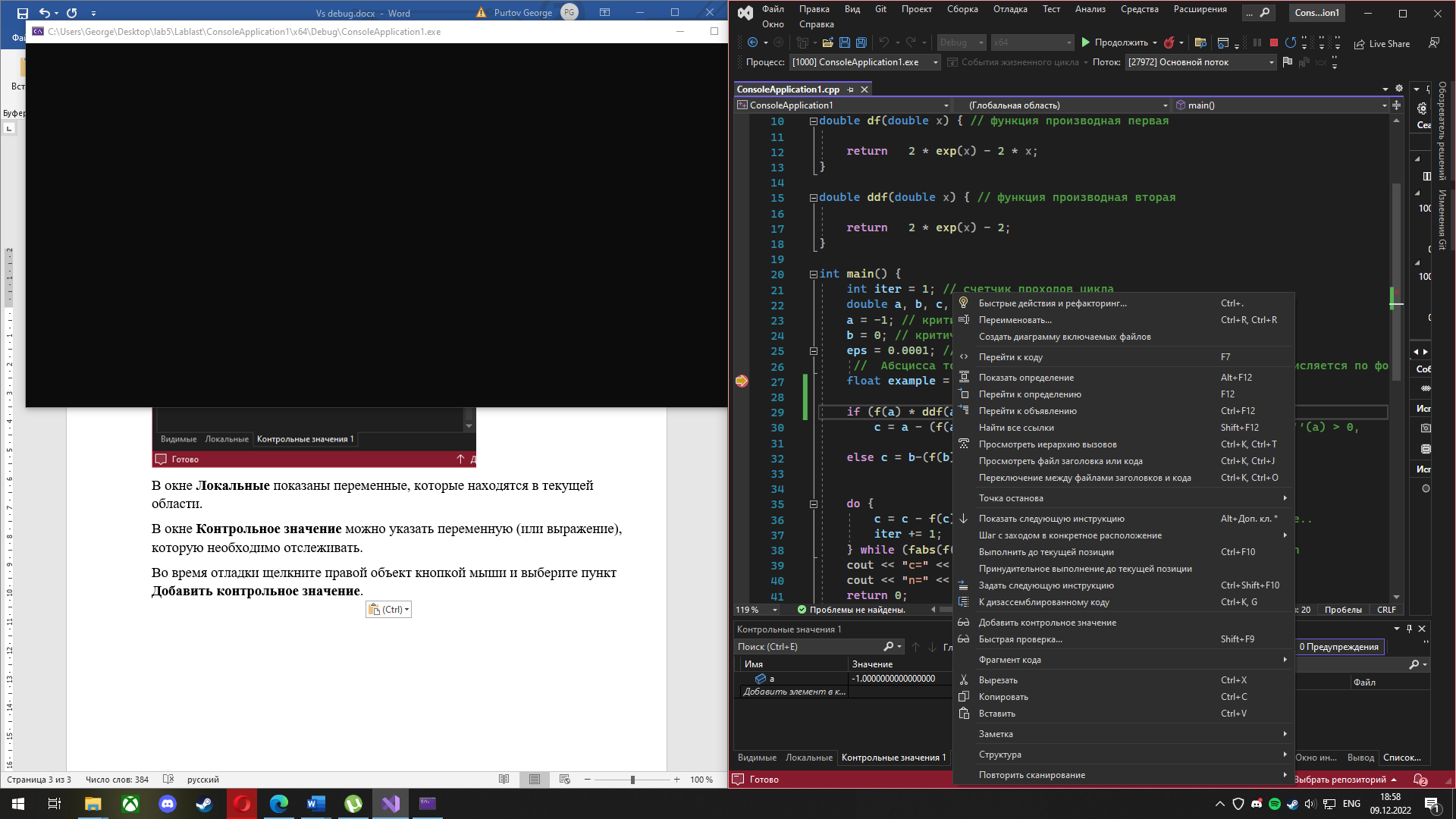


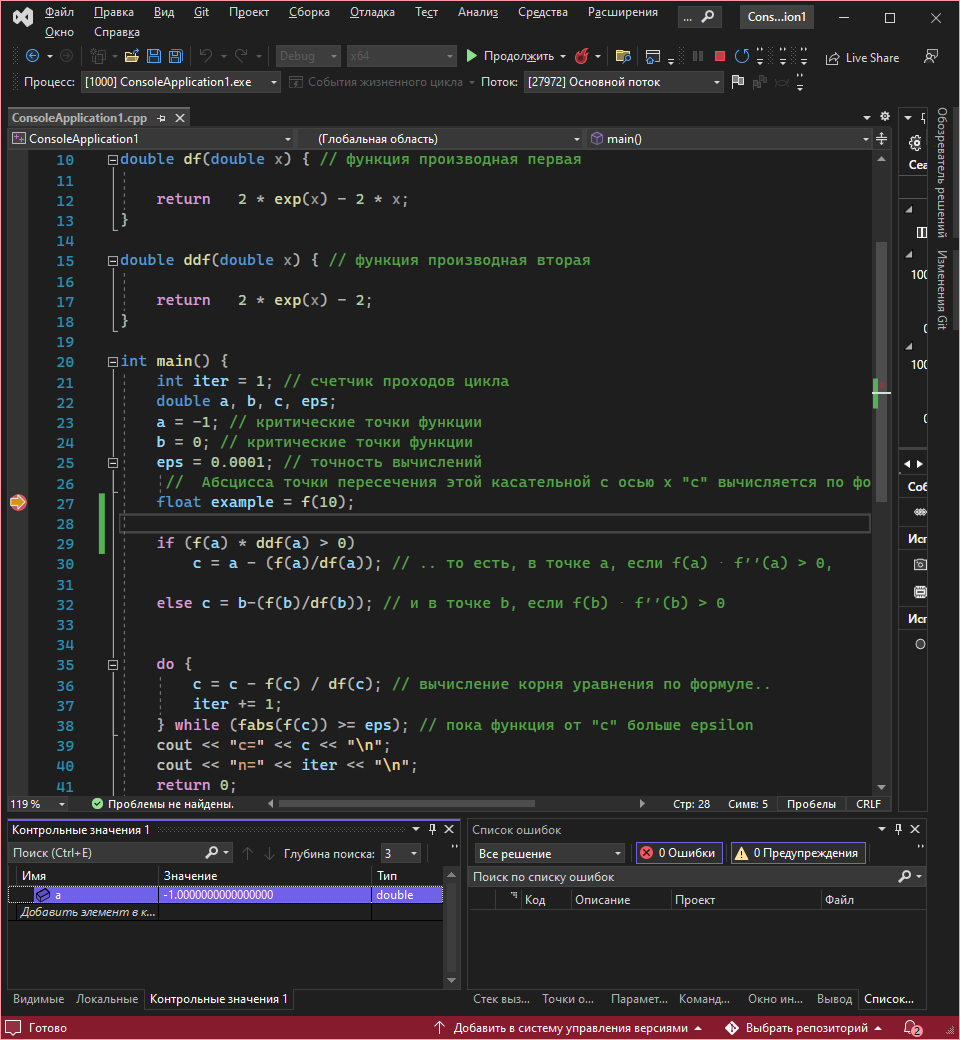
В окне **Локальные** показаны переменные, которые находятся в текущей области.



В окне **Контрольное значение** можно указать переменную (или выражение), которую необходимо отслеживать.

Во время отладки щелкните правой объект кнопкой мыши и выберите пункт **Добавить контрольное значение**.

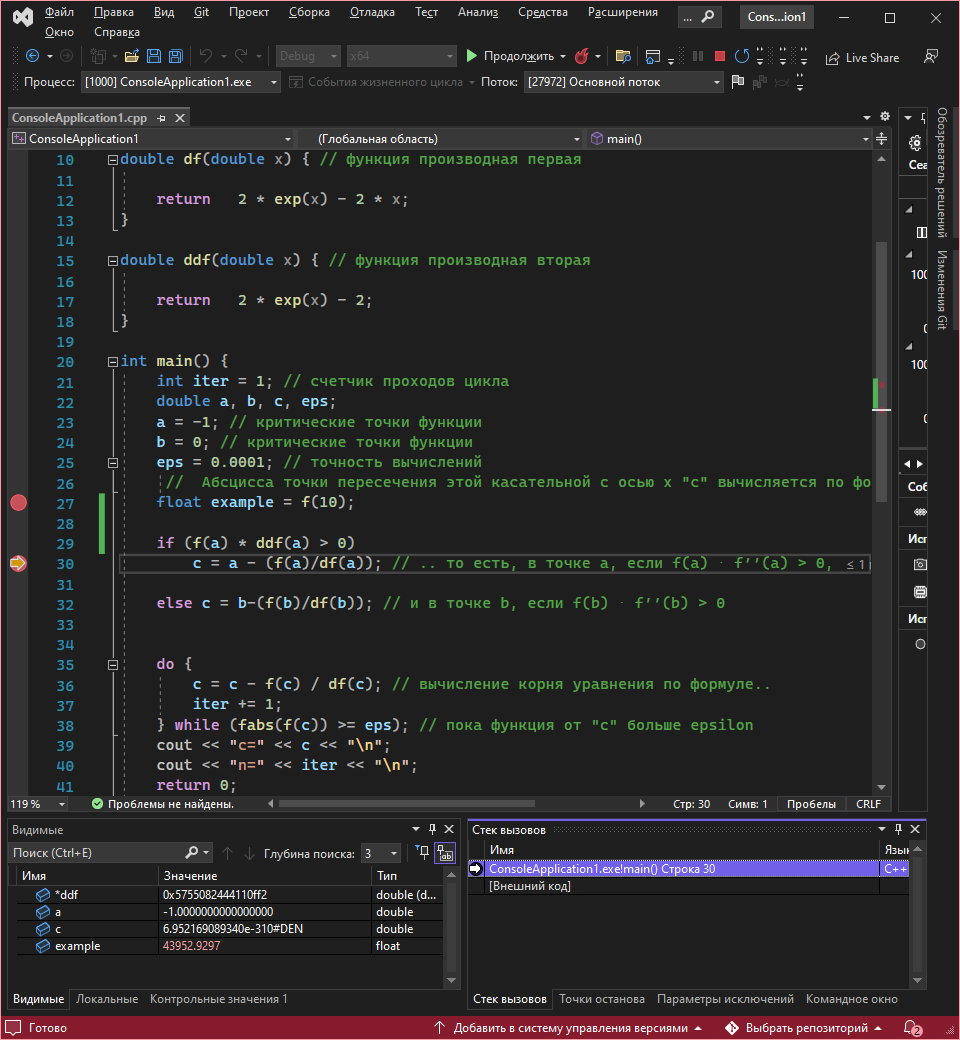




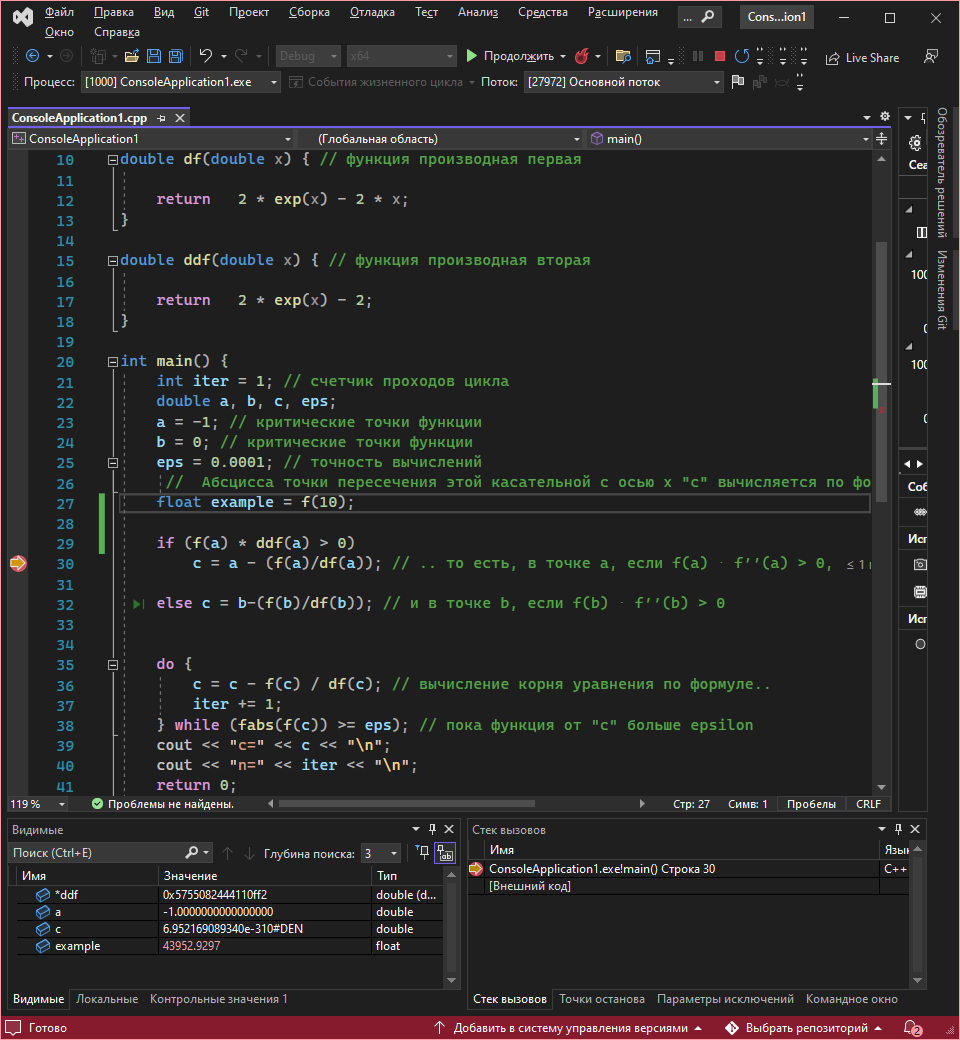
По мере перемещения по отладчику вы можете наблюдать за изменением его значения. В отличие от других окон переменных, в окне **Контрольное значение** всегда отображаются просматриваемые вами переменные (они выделяются серым цветом, когда находятся вне области действия).

### Просмотр стека вызовов

В окне **Стек вызовов** показан порядок вызова методов и функций. В верхней строке показана текущая функция. Во второй строке показана функция или свойство, из которого она вызывалась, и т. д. Стек вызовов хорошо подходит для изучения и анализа потока выполнения приложения.



Дважды щелкните строку кода, чтобы просмотреть исходный код. При этом также изменится текущая область, проверяемая отладчиком. Это не перемещает отладчик.



### Источники

1) <https://learn.microsoft.com/ru-ru/visualstudio/debugger/debugger-feature-tour?view=vs-2022>