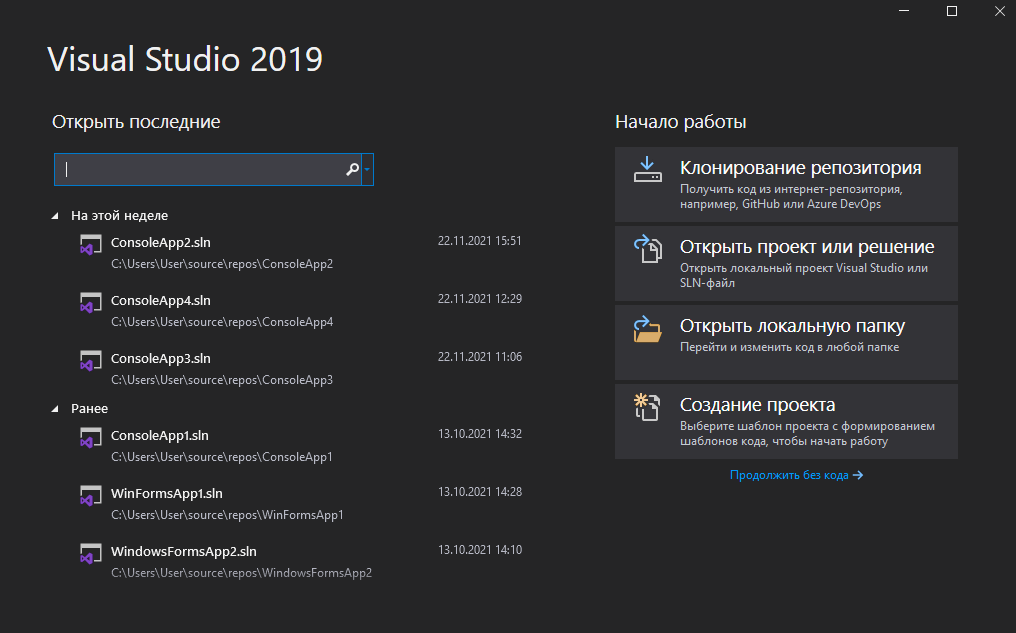
1. **Измерение производительности приложения посредством анализа использования центрального процессора**

### Сбор данных профилирования

* + 1. Запускаем Visual Studio, открываем любой проект

Рис. 1 «Visual Studio»



3.1.2. Установите точку останова в приложении в точке, где вы хотите проверить загрузку центрального процессора.

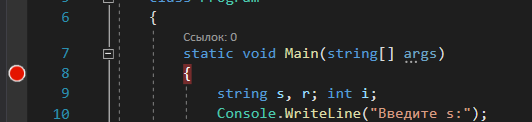


Рис. 2 «Первая точка остановы»

3.1.3. Устанавливаем вторую точку останова в конце функции или области кода, который требуется проанализировать.

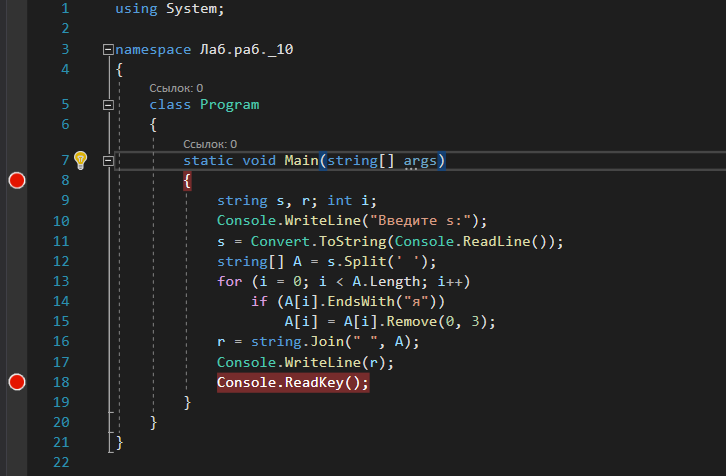


Рис. 3 «Вторая точка остановы»

3.1.4. Окно **Средства диагностики** появится автоматически, если вы не отключали эту функцию. Чтобы снова открыть окно, щелкните **Отладка > Окна > Показать средства диагностики.**

3.1.5. Вы можете выбрать, что следует просмотреть, Использование памяти или Использование ЦП (либо оба средства), с помощью параметра Выбор средств на панели инструментов.

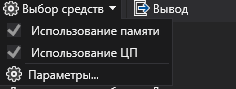


Рис. 4 «Выбор средств»

3.1.6. По завершении загрузки приложения отображается представление "Сводка" средств диагностики.

3.1.7. Запустите сценарий, который вызвал срабатывание первой точки останова.

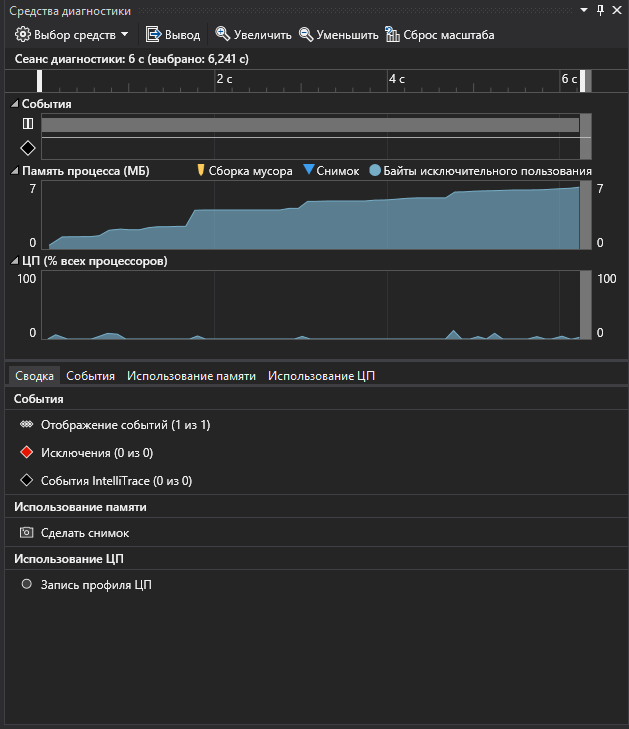


Рис. 5 «Окно диагностики во время отладки»

3.1.8. Приостановив отладчик, включите сбор данных о загрузке центрального процессора, а затем откройте вкладку **Загрузка ЦП**. При выборе пункта Запись профиля ЦП Visual Studio начнет записывать функции и сведения о времени их выполнения. Эти собранные данные можно просматривать только в том случае, если приложение останавливается в точке останова.

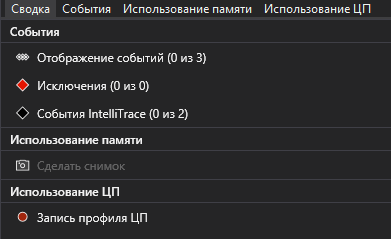


Рис. 6 «Запись профиля ЦП»

3.1.9. Нажмите клавишу F5, чтобы запустить приложение до второй точки останова.

Теперь у вас есть данные о производительности приложения именно для той области кода, которая выполняется между двумя точками останова.

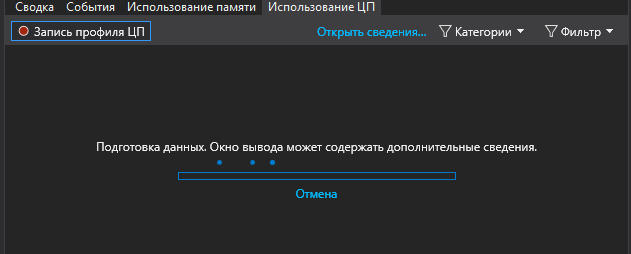
Профилировщик начинает подготавливать данные потока. Дождитесь завершения этой операции.

Рис. 7 «Подготовка данных»

Средство "Загрузка ЦП" выведет отчет на вкладке **Загрузка ЦП**.

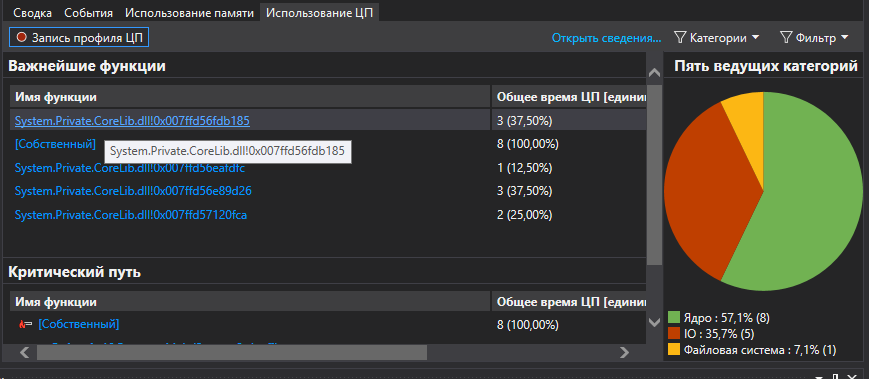
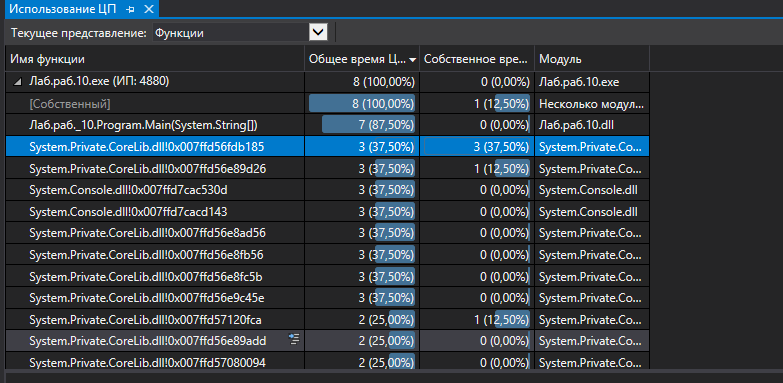


Рис. 8 «Загрузка центрального процессора»



### 

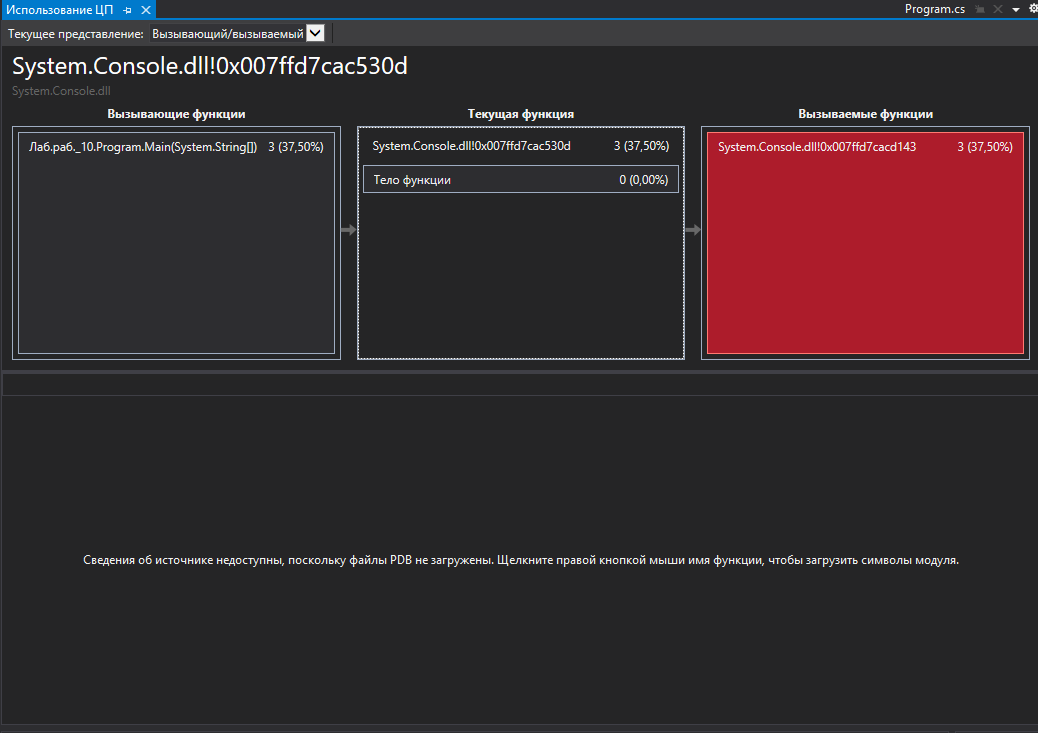
Рис. 9 «Список функций»

### 3.2. Анализ данных о загрузке центрального процессора

3.2.1. В списке функций дважды щелкните одну из функций вашего приложения, которая выполняет много работы.

При двойном щелчке функции в левой панели откроется представление **Вызывающий/вызываемый**.

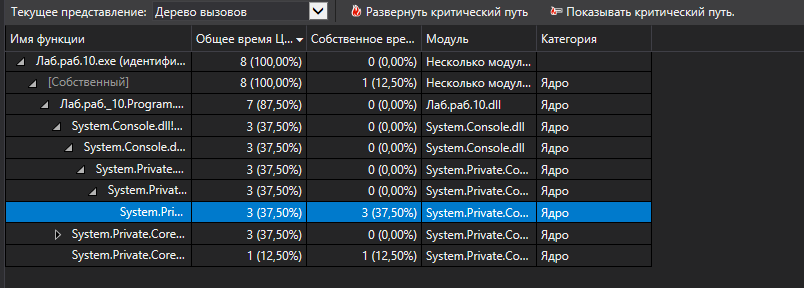
Рис. 10 «Представление **Вызывающий/вызываемый**»



В этом представлении выбранная функции отображается в заголовке и в поле Текущая функция. Функция, вызывавшая текущую функцию, отображается в левой части окна в разделе Вызывающие функции, а все функции, вызываемые текущей функцией, отображаются в поле Вызываемые функции справа. (Можно выбрать любое поле, чтобы изменить текущую функцию.) В этом представлении показано общее время (мс) и доля общего времени выполнения приложения, затраченного на выполнение функции. В поле Тело функции также показан общий объем времени (и доля времени), затраченного в теле функции за исключением времени, затраченного в вызываемых и вызывающих функциях.

3.2.2. Чтобы увидеть более обобщенное представление, показывающее порядок, в котором вызываются функции, выберите в раскрывающемся списке в верхней части панели пункт **Дерево вызовов**.

Рис. 11 «Дерево вызовов»



Чтобы увидеть вызовы функций, которые используют самый высокий процент ЦП в представлении дерева вызовов, нажмите «**Развернуть критический путь»**.

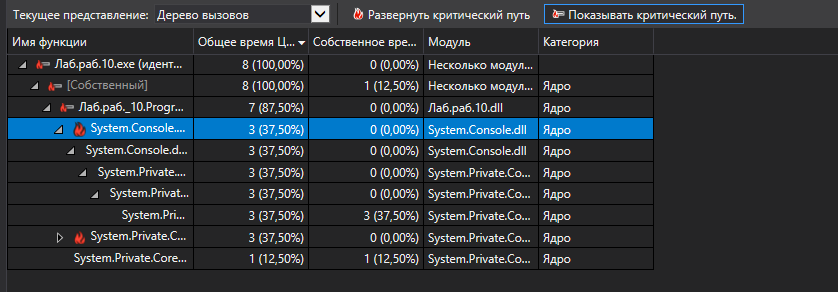
****

Рис. 12 «Развернули критически путь»

Просмотр внешнего кода

Чтобы посмотреть пути к вызовам внешнего кода, выберите **Показать внешний код** в списке **Представление фильтра** и выберите **Применить**.

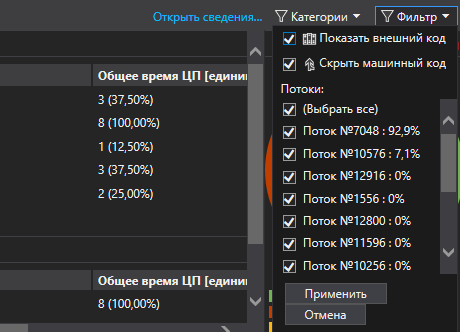


Рис.13 «Показать внешний код»