Задание на практику №1

Измерение производительности приложения посредством анализа использования ЦП

Для измерения производительности приложения необходимо иметь уже готовую программу, которая исправно работает и не имеет ошибок. Первым шагом нужно скачать программу. Для этого переходим по ссылки на GIT и скачиваем папку с программой.



Рис. 1 GIT с программой

Далее открываем Visual Studio, для дальнейшей работы с программой для измерения её производительности.

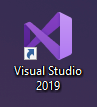


Рис. 2 Иконка Visual Studio

Затем в появившемся окне выбираем в правой части пункт под названием «Открыть локальную папку». Ищем папку и открываем её. После всех данных действий у нас имеется готовая программа.

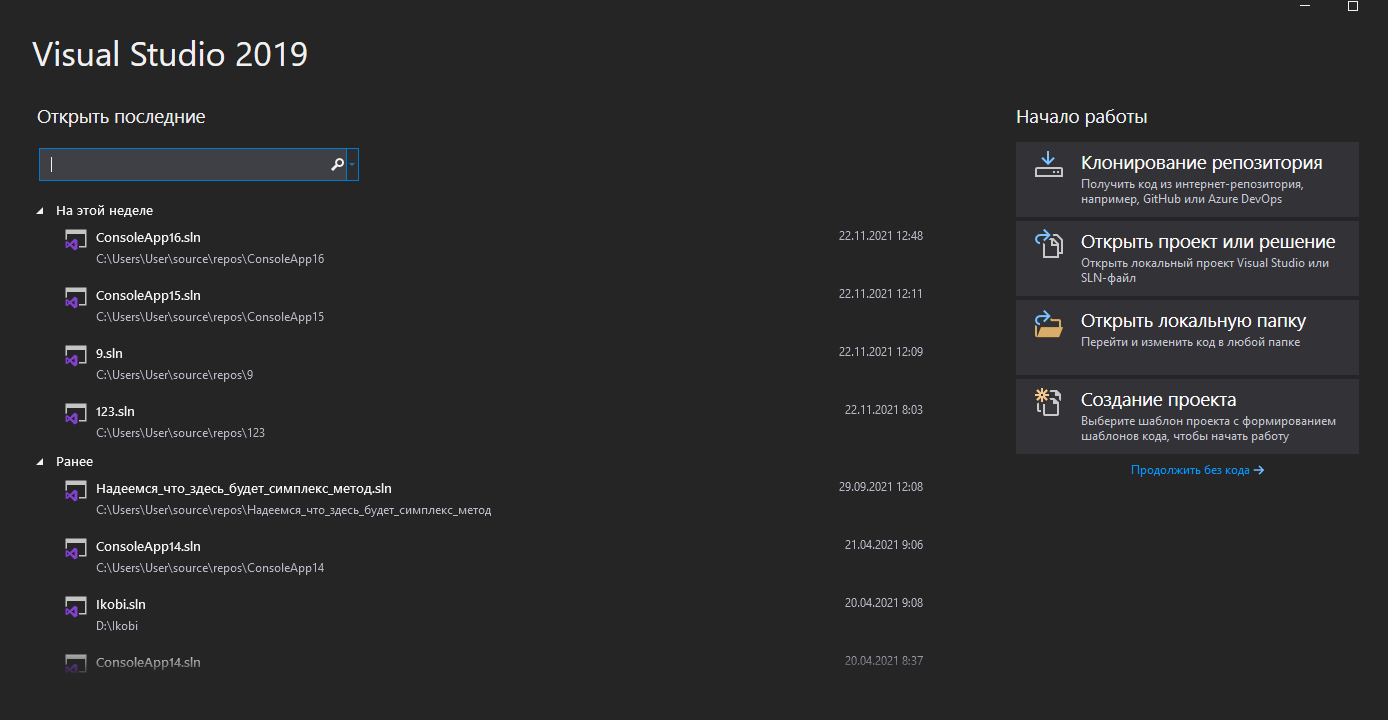


Рис. 3 Начальная страница Visual Studio

Для измерения производительности программы ставим точку остановки в коде для того, чтобы при откладке программа смогла определить, где необходимо сделать остановку и дождаться следующих указаний.

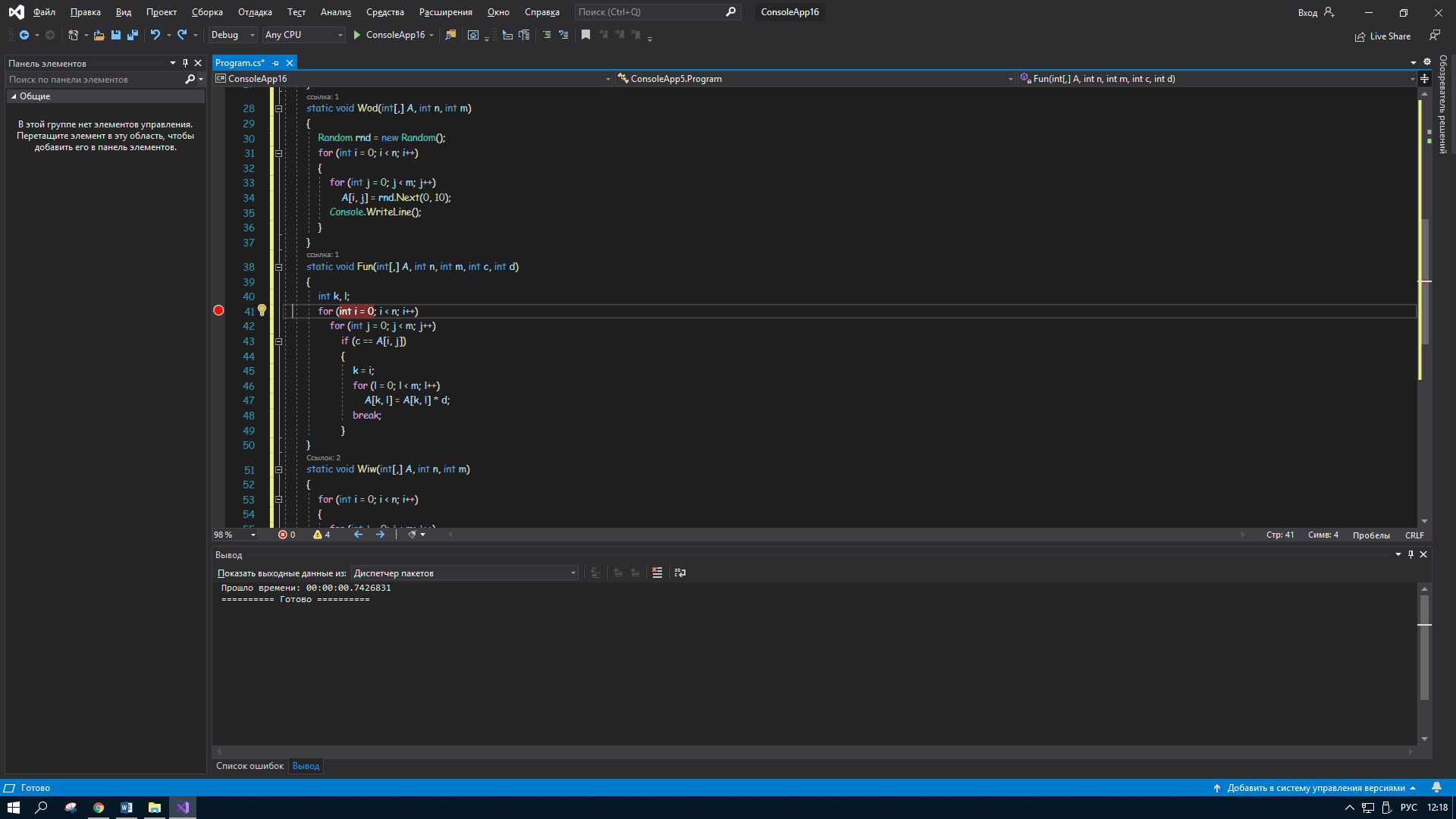


Рис. 4 Открытие нового проекта

Далее устанавливается, вторя точка остановки в том месте, где необходимо измерить производительность. При помощи двух точек остановки и откладки можно провести замеры производительности программы на данном участке кода.

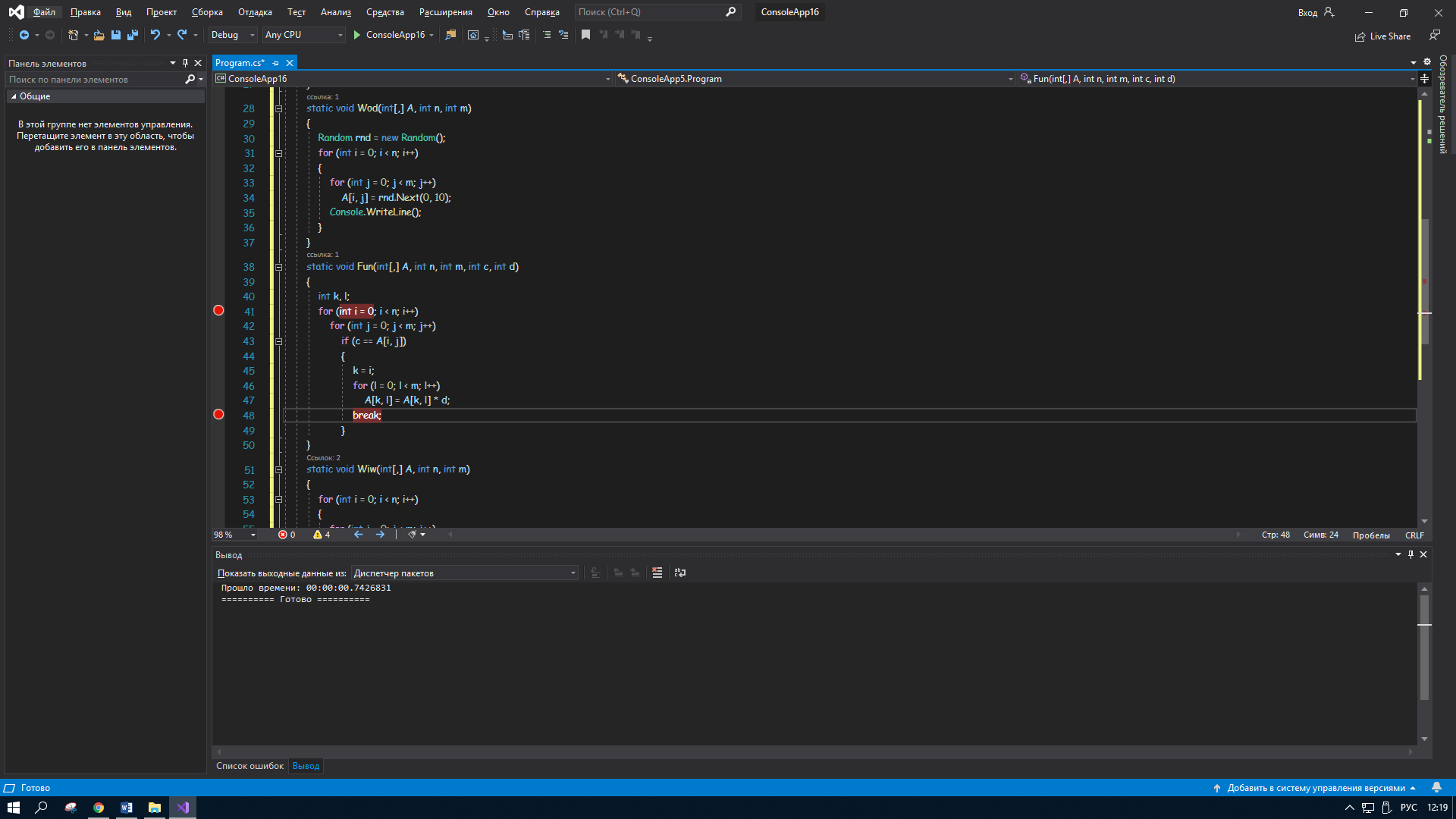


Рис. 5 Точки остановки

Запускаем откладку, для измерения производительности программы.

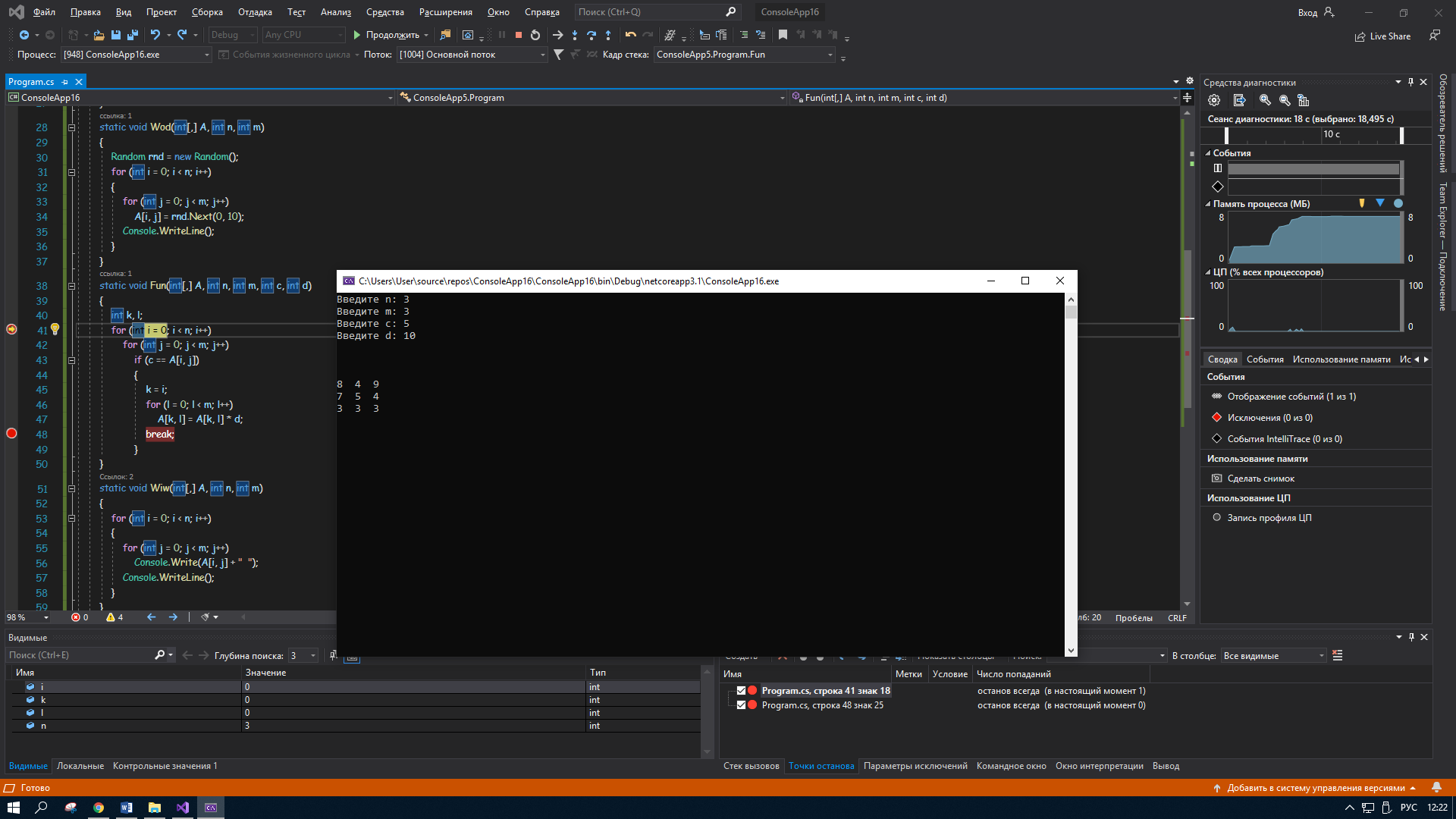


Рис. 6 Режим отладки

Далее после того, как программа достигла первой точки остановки, в новом появившемся окне «Средства диагностики» располагаются графики, на которых изображены показатели загруженности памяти процессора и загруженность ЦП.

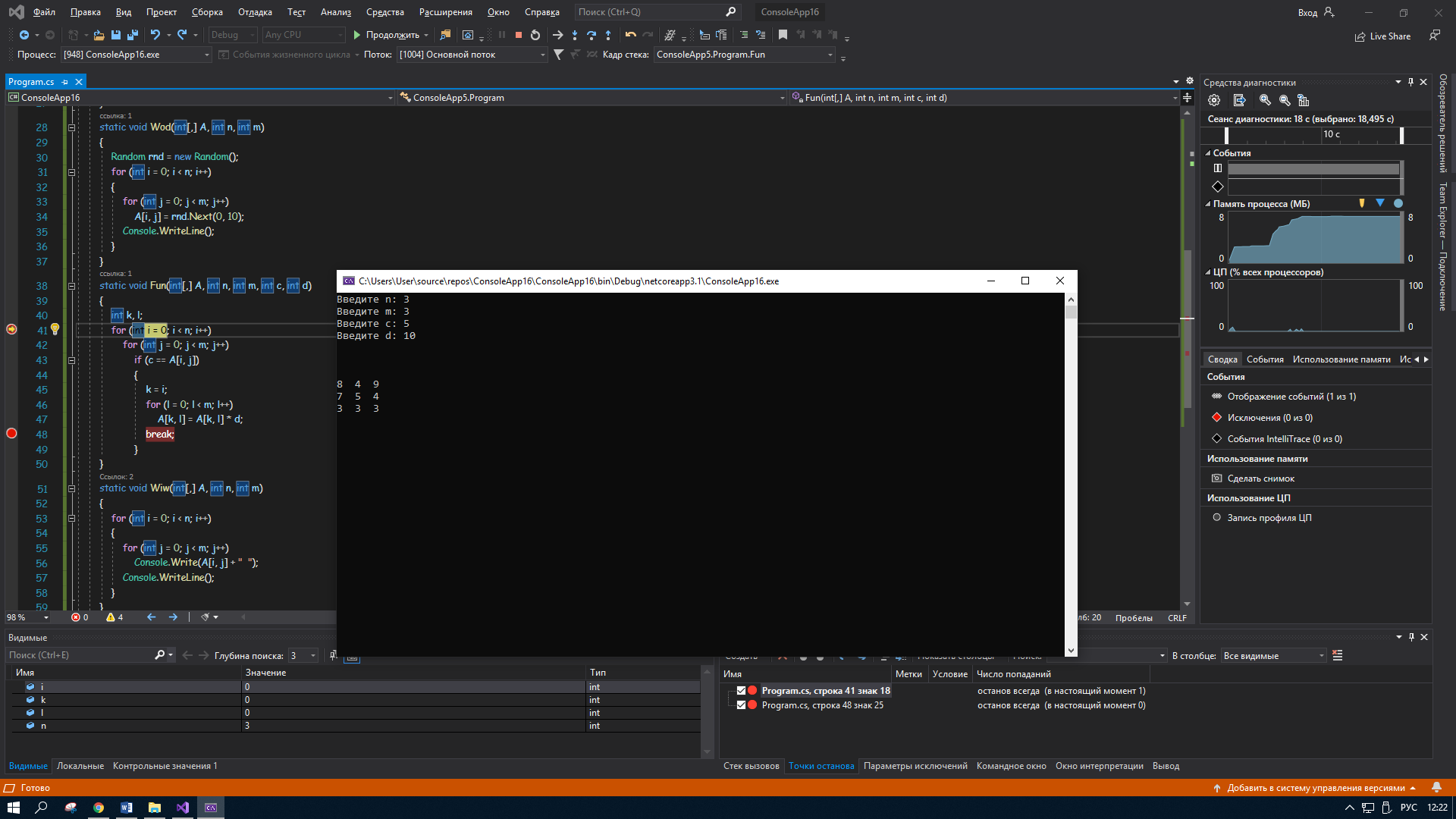


Рис. 7 Показатели производительности программы

Следующим шагом включаем показатели использования ЦП, для того чтобы увидеть, как распределяет нагрузку центральный процессор.

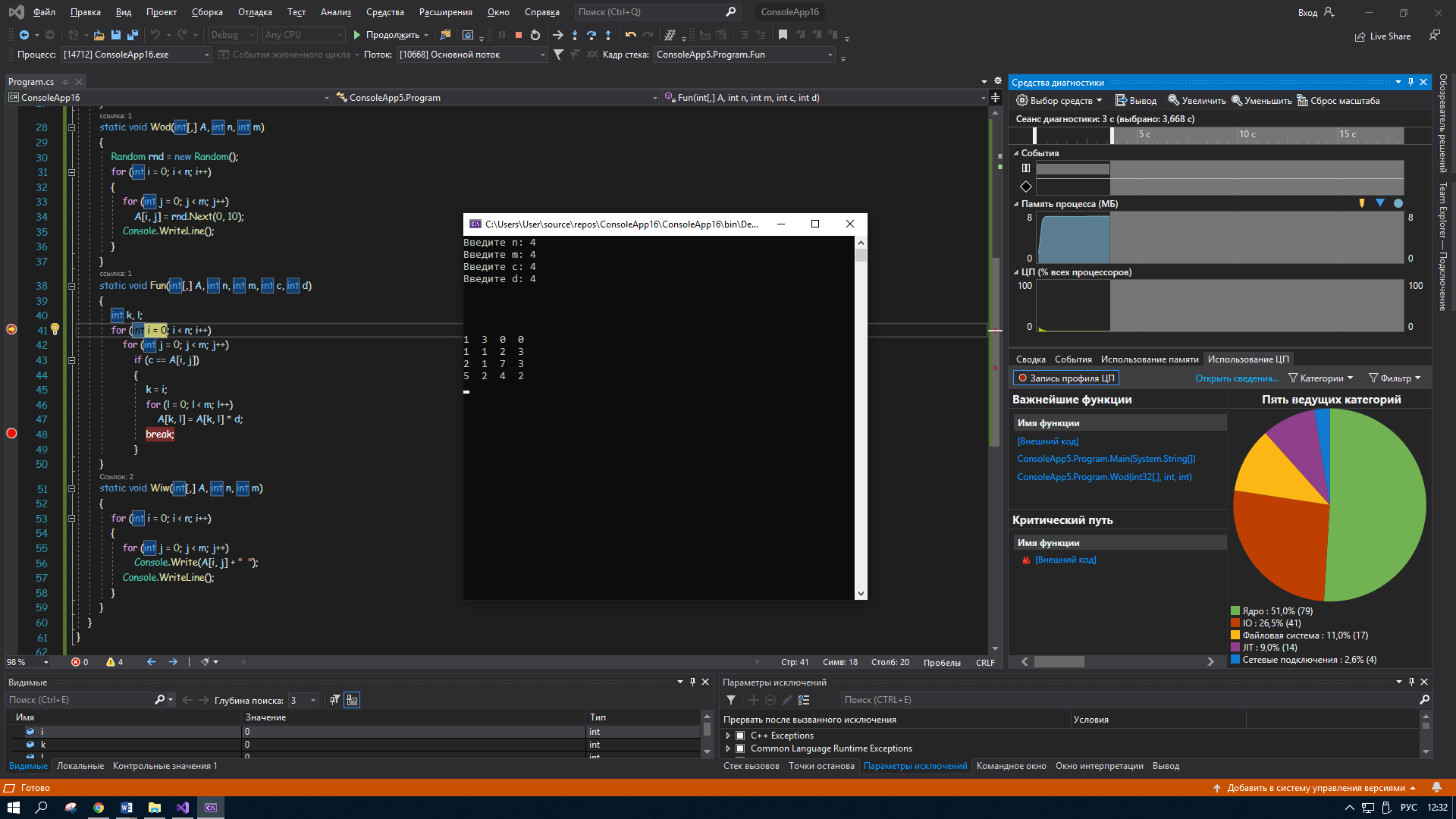


Рис. 8 Показатели программы на первой точке останова с данными ЦП

Затем, чтобы отследить производительность программы нажимаем F5, тем самым продолжая откладку. Как только программа остановиться на точке остановке, данные о использование памяти ЦП поменяются.

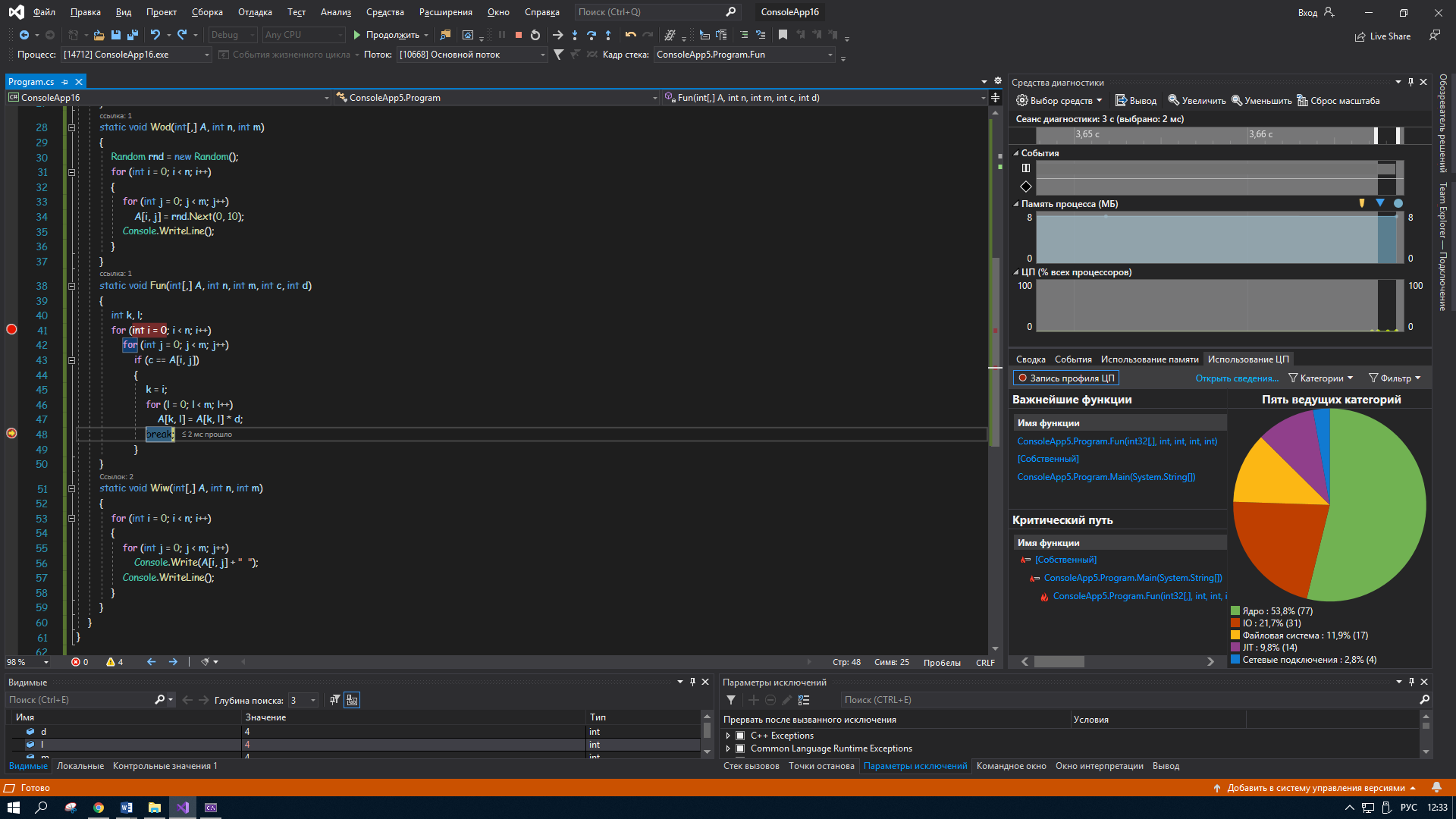


Рис. 9 Новые данные программы в другой точки остановки

*Анализ данных о загрузке ЦП*

Чтобы проанализировать данные о загрузки ЦП необходимо выделить функцию, на которую ЦП выделил большой объём данных. Для этого открываем функцию двойным щелчком и переходим на раздел «Вызывающий/вызываемый».

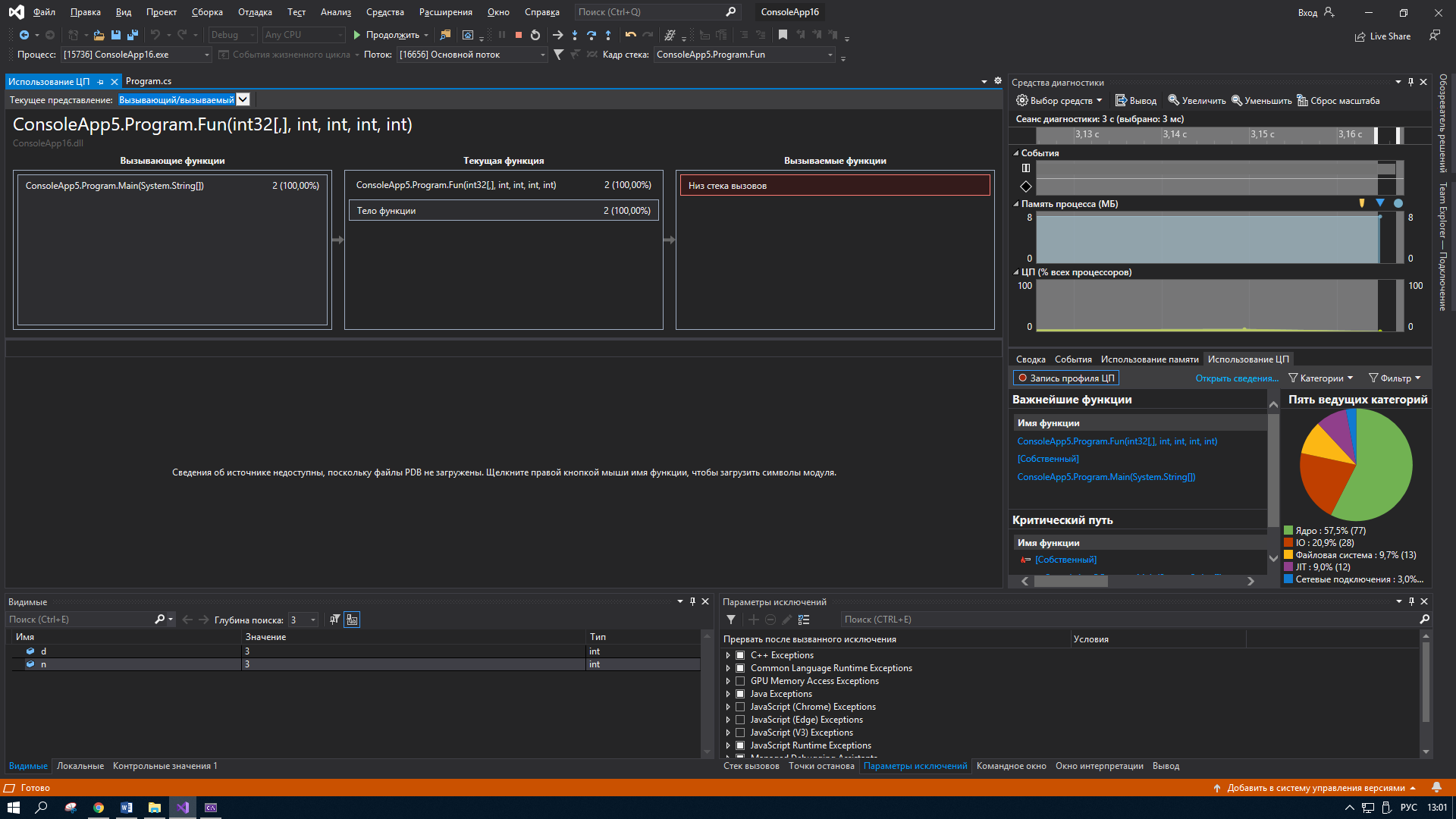


Рис. 10 Просмотр функции Fun

Чтобы увидеть более обобщенное представление, показывающее порядок, в котором вызываются функции, выбираем раздел «**Дерево вызовов**».

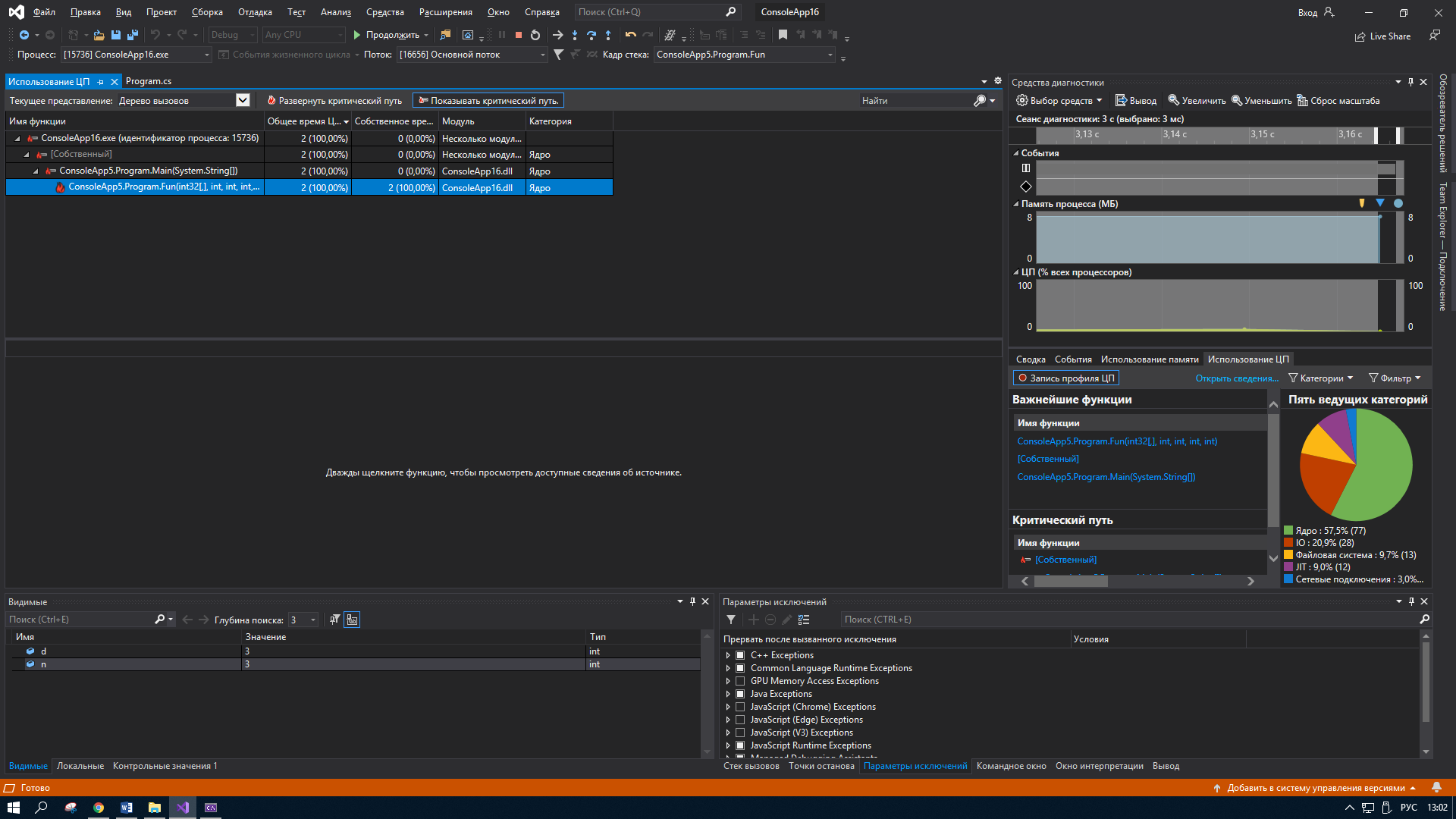


Рис. 11 Дерево вызовов функций

Для того чтобы увидеть все остальные функции можно нажать во вкладке «Фильтры» и поставить галочку на против пункта «Показать внешний код».

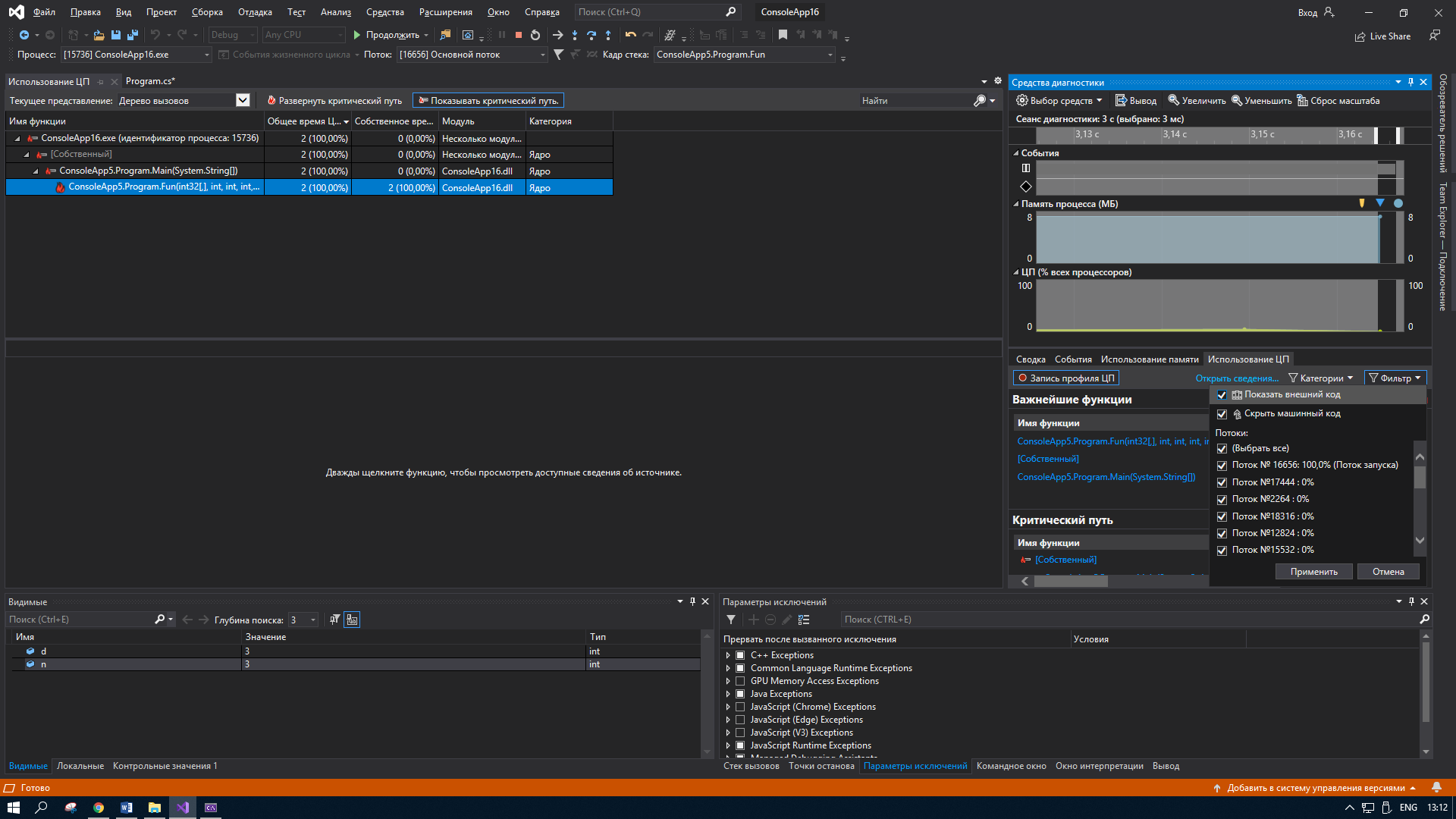


Рис. 12 Кнопка «Показать внешний код»

Далее во вкладке «Дерево вызовов» появились многие новые функции программы. Следовательно, при разборе функций можно определить, как программа отправляет данные на ЦП, а центральный процессор в свою очередь распределяет данные так, чтобы программа работала лучше и быстрее.

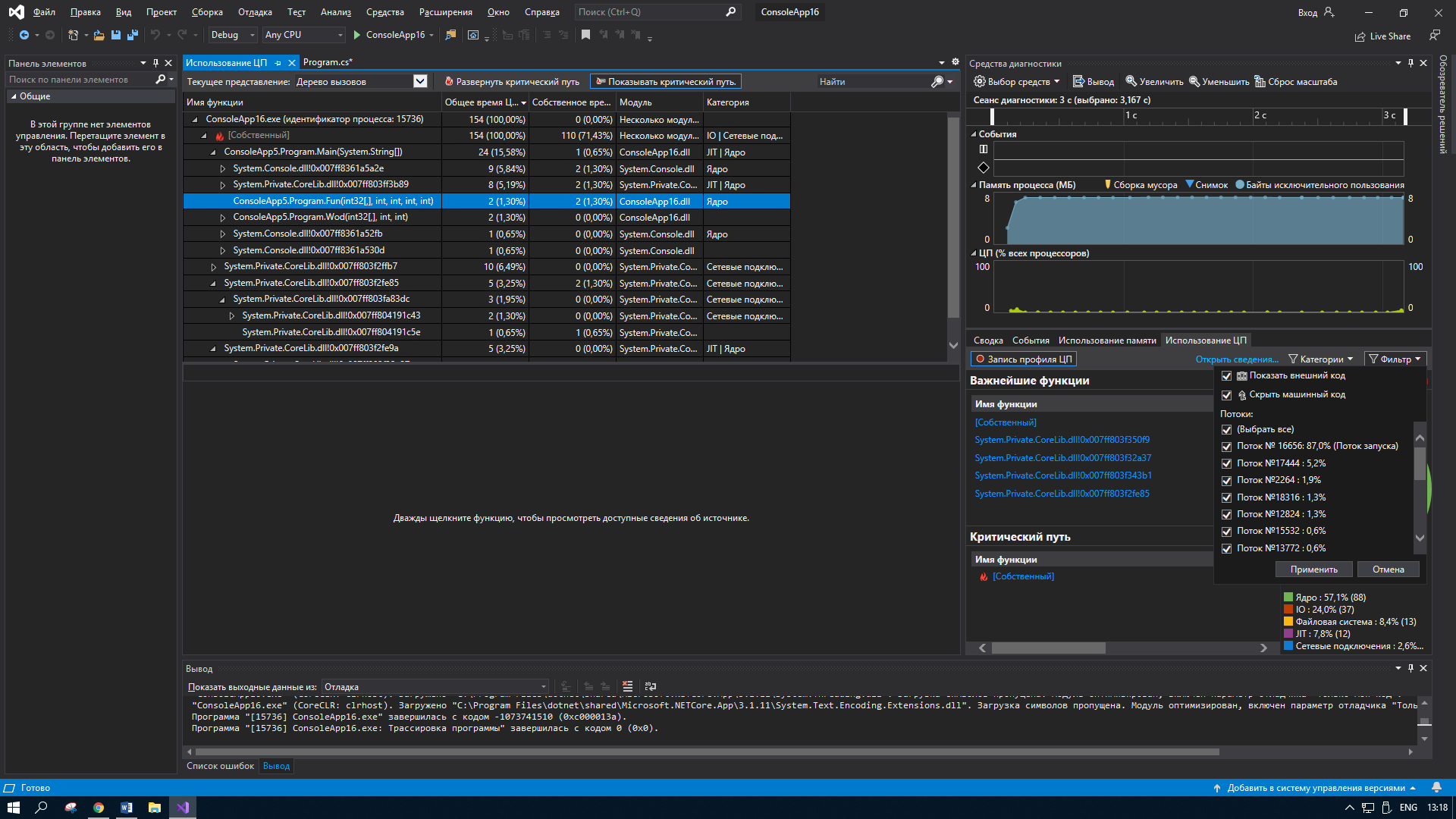


Рис. 13 Дерево вызовов функций с внешними кодами

В заключение хочется сказать, что при усовершенствовании кода можно оптимизировать процесс загрузки ЦП. При изменении ввода массива путём ручного ввода за место рандомизации нагрузка на ЦП измениться. При маленькой нагрузки на ЦП, программа будет обрабатывать данные быстрее.