

Учебная практика №1 (Задание №1)

«Измерение производительности приложения посредством анализа использования ЦП»

N. План хода работы в учебной практике №1.

N.I. Этап 1. Сбор данных профилирования в ПО Visual Studio 2019. (I)

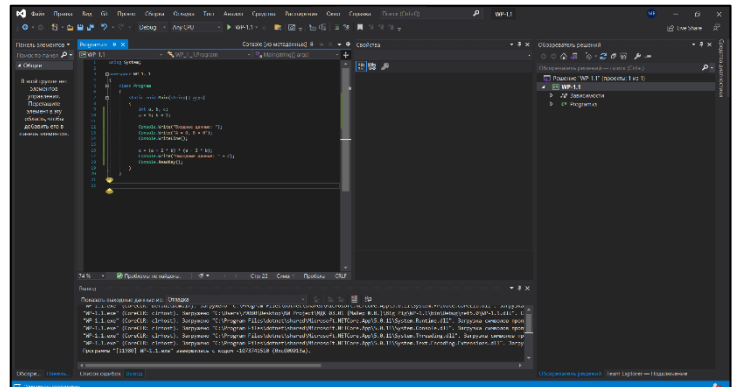
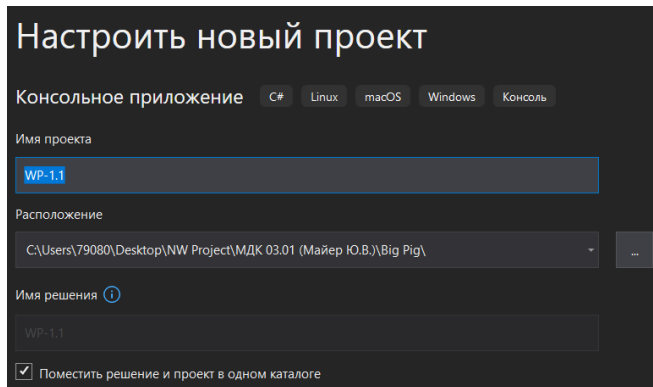
N.II. Этап 2. Анализ данных о загрузке ЦП в ПО Visual Studio 2019. (II)

N.III. Этап 3. Просмотр внешнего кода в ПО Visual Studio 2019. (III)

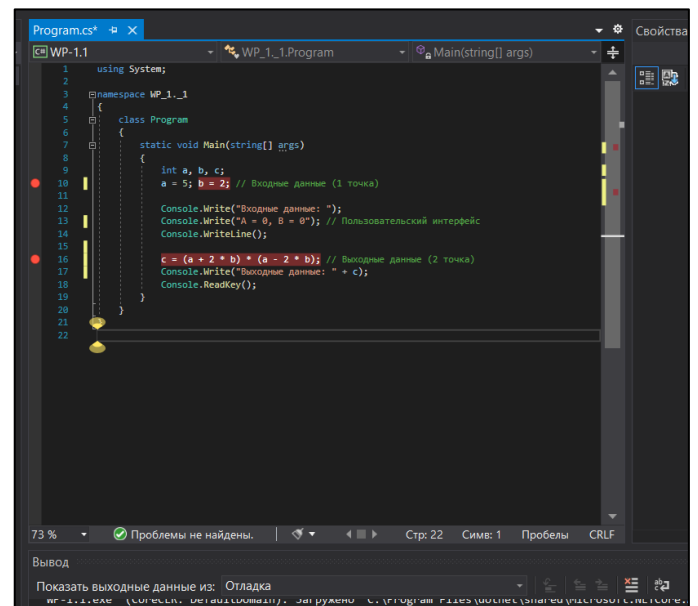
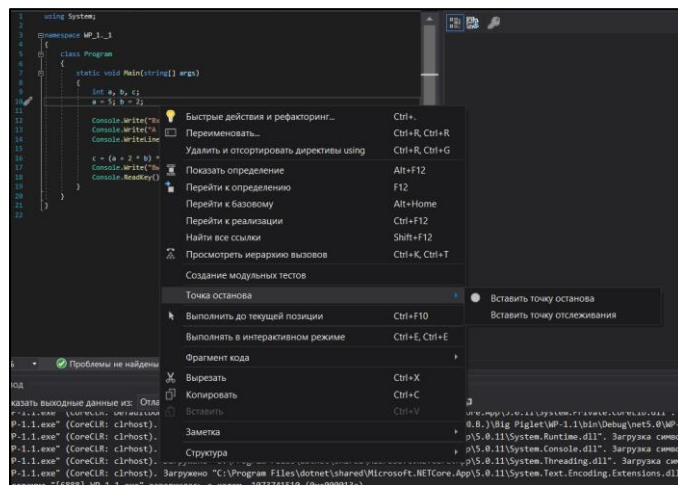
N.IV. Загрузка рабочего пакета данных в репозиторий GitHub. (IV)

I. Этап 1. Сбор данных профилирования в ПО Visual Studio 2019. (→)

I.I. Создаем проект для отладки в Visual Studio 2019*. Пишем простой программный код на C#.

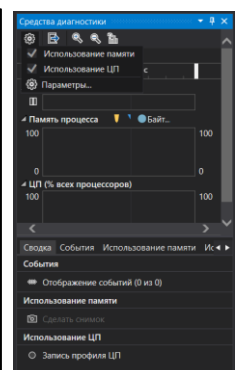
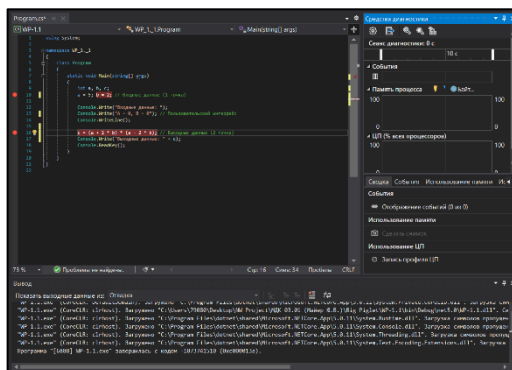
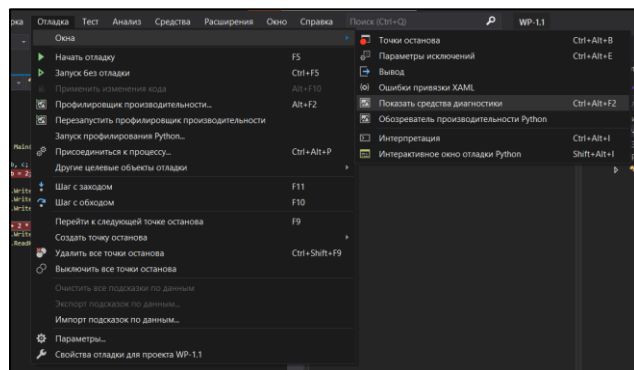


I.II. Ставит две точки останова (ПКМ) в начале/конце кода. В этом промежутке пройдет анализ.

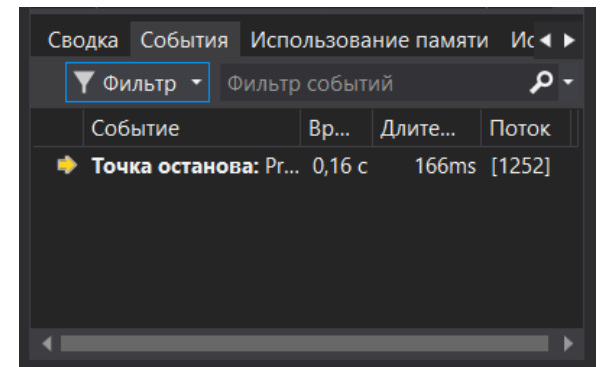
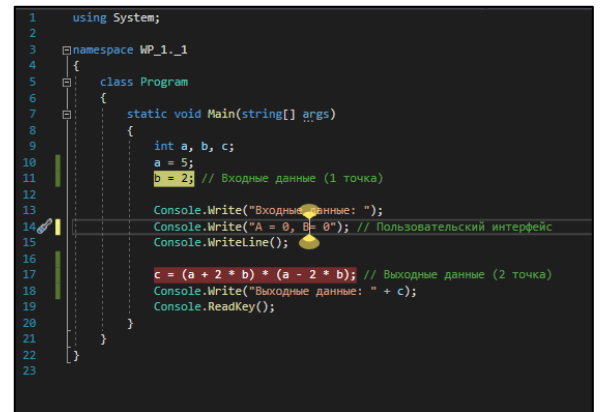
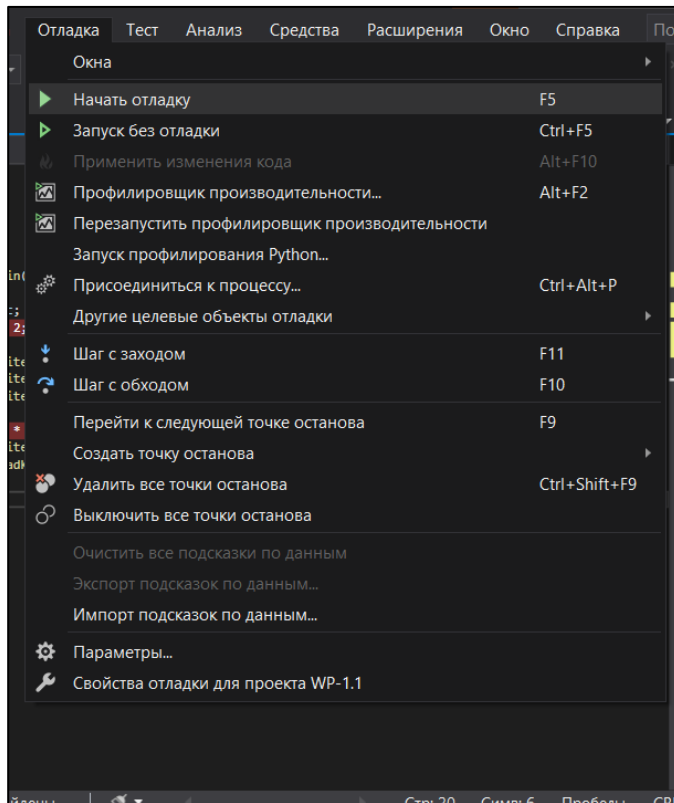


**Программа Visual Studio 2019
запускается от имени администратора!*

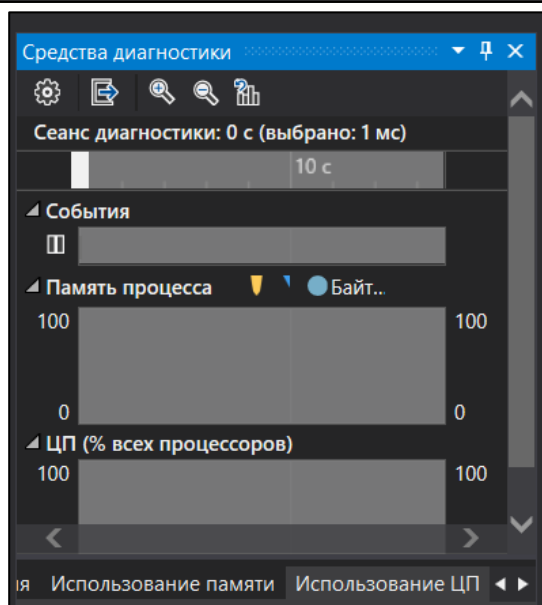
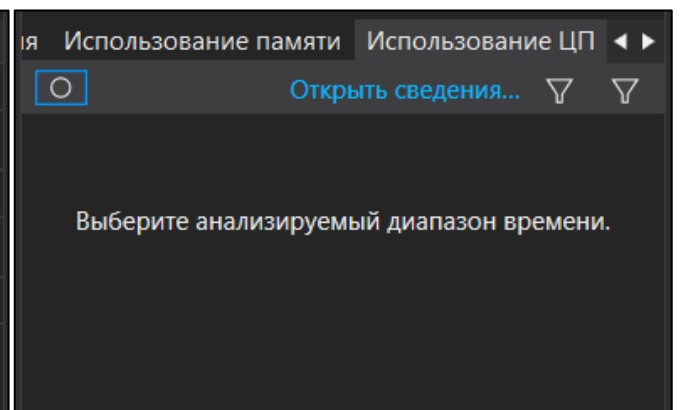
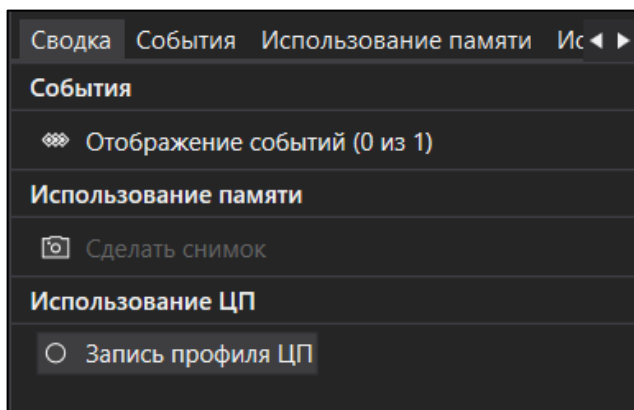
I.III. В разделе «Окна» ищем «Средства диагностики». В настройках включаем «Загрузка ЦП».



I.IV. Запускаем отладку и перемещаемся по частям кода с помощью клавиши F11 (Fn + F11).



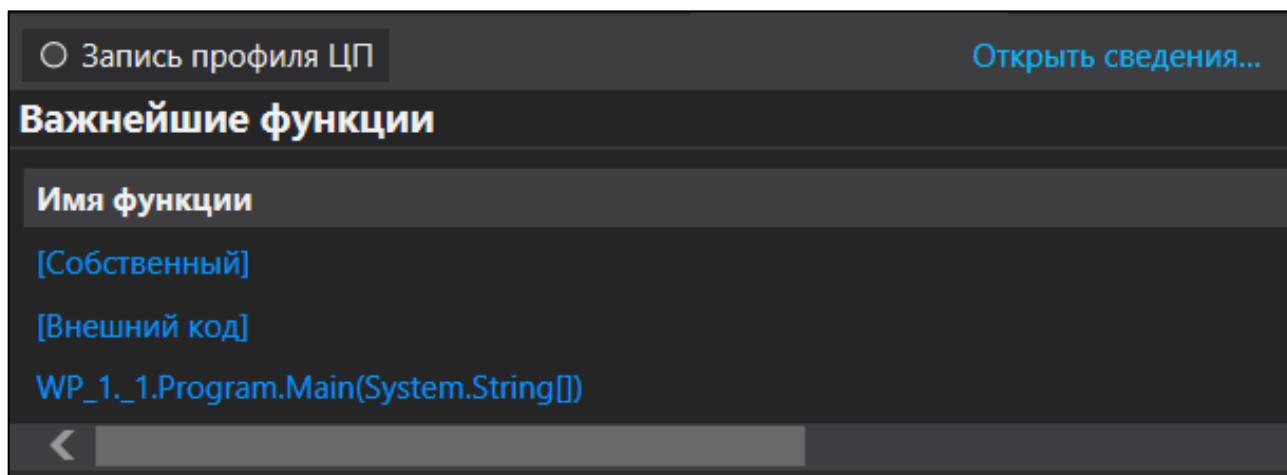
I.V. Приостанавливаем отладку и активируем раздел «Запись профиля ЦП». Выбираем нужный нам промежуток времени. С помощью F11 (Fn + F11) движемся ко второй точке останова



Итог: Этап выполнен. Переходим к следующему.

II. Этап 2. Анализ данных о загрузке ЦП в ПО Visual Studio 2019. (→)

II.I. Кликаем два раза по одному из процессов в разделе «Использование ЦП» и попадаем в раздел* «Дерево вызовов». Проводим исследование раздела* «Дерево вызовов».

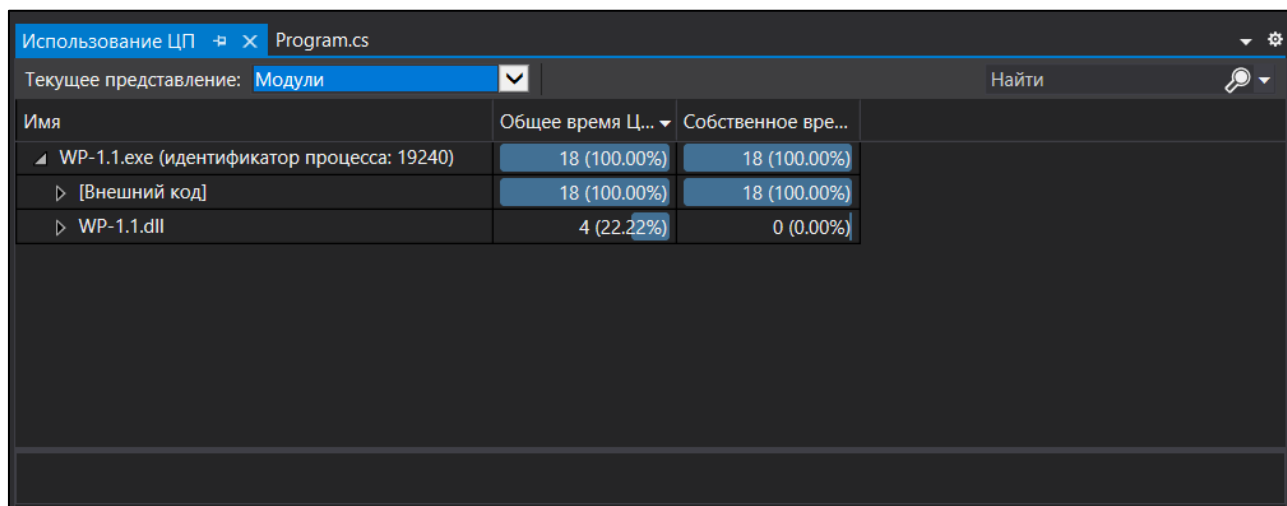


Имя функции		Общее время ЦП	Собственное вре...	Модуль	Категория
WP-1.1.exe (идентификатор процесса: 19240)		18 (100.00%)	0 (0.00%)	Несколько модул...	
[Собственный]		18 (100.00%)	14 (77.78%)	Несколько модул...	JIT Ядро
WP_1_1.Program.Main(System.String[])		4 (22.22%)	0 (0.00%)	WP-1.1.dll	JIT Ядро
[Внешний код]		4 (22.22%)	4 (22.22%)	Несколько модул...	JIT Ядро

II.II. С помощью специального переключателя заходим в раздел* «Функции».

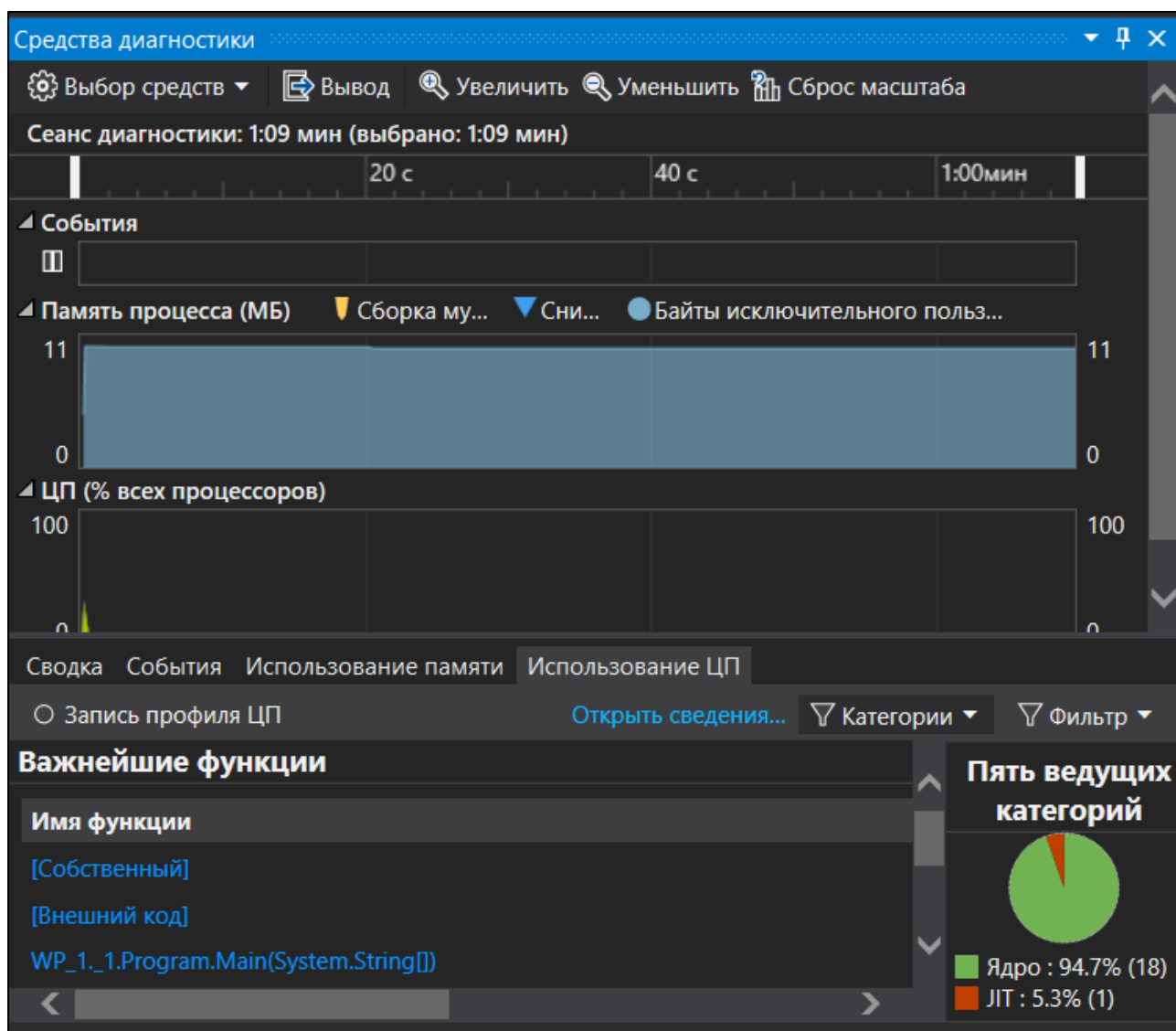
Имя функции		Общее время ЦП	Собственное вре...	Модуль
WP-1.1.exe (ИП: 19240)		18 (100.00%)	0 (0.00%)	WP-1.1.exe
[Собственный]		18 (100.00%)	14 (77.78%)	Несколько модул...
[Внешний код]		4 (22.22%)	4 (22.22%)	Несколько модул...
WP_1_1.Program.Main(System.String[])		4 (22.22%)	0 (0.00%)	WP-1.1.dll

II.III. С помощью специального переключателя попадаем в раздел* «Модули». (представление)



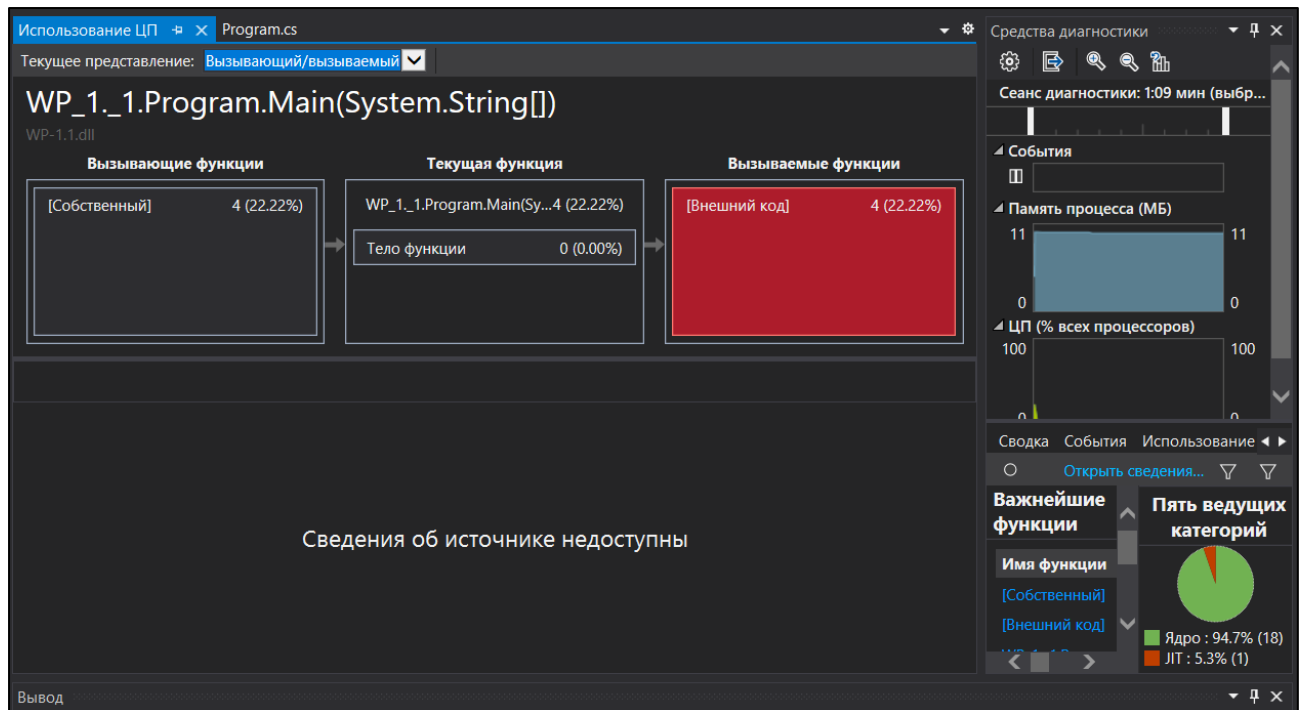
Имя	Общее время ЦП	Собственное вре...
WP-1.1.exe (идентификатор процесса: 19240)	18 (100.00%)	18 (100.00%)
> [Внешний код]	18 (100.00%)	18 (100.00%)
> WP-1.1.dll	4 (22.22%)	0 (0.00%)

II.IV. Знакомимся с изменившимся разделом - «Средства диагностики». (находится справа)



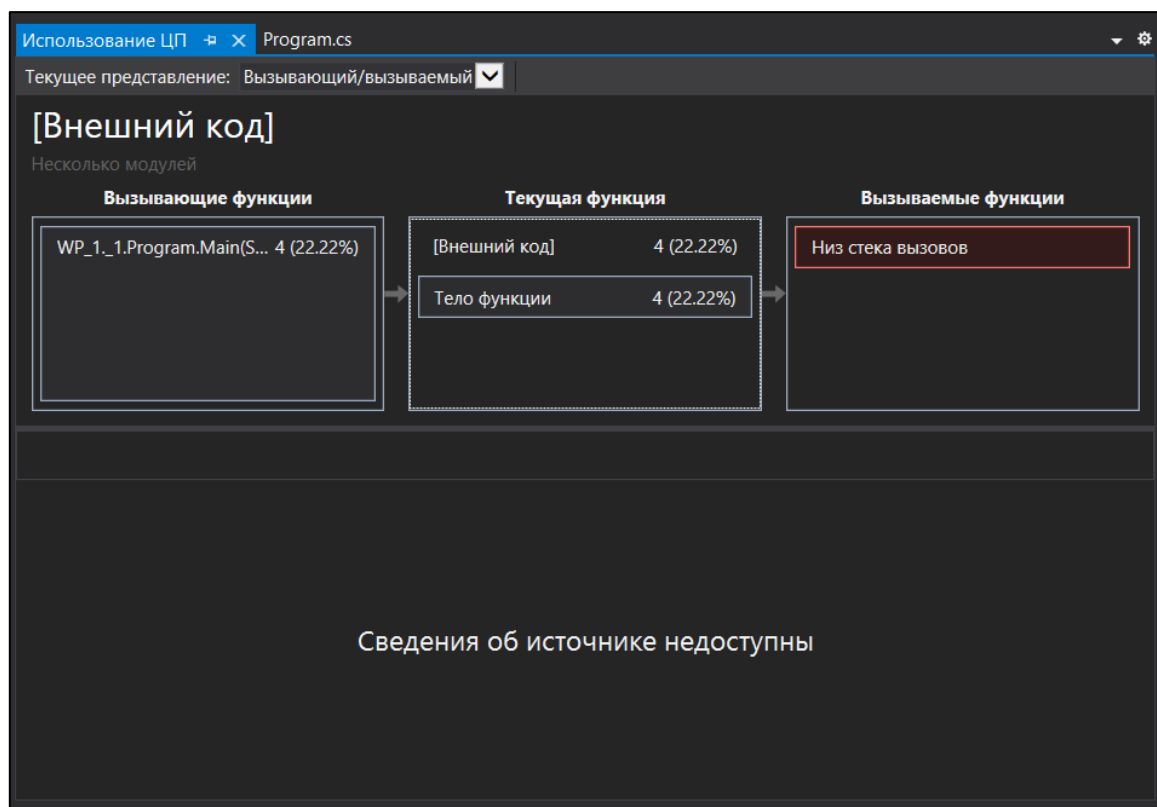
Итог: Этап выполнен. Переходим к следующему.

II.V. С помощью переключателя попадаем в раздел* «Вызывающий/вызываемый».



III. Этап 3. Просмотр внешнего кода в ПО Visual Studio 2019. (→)

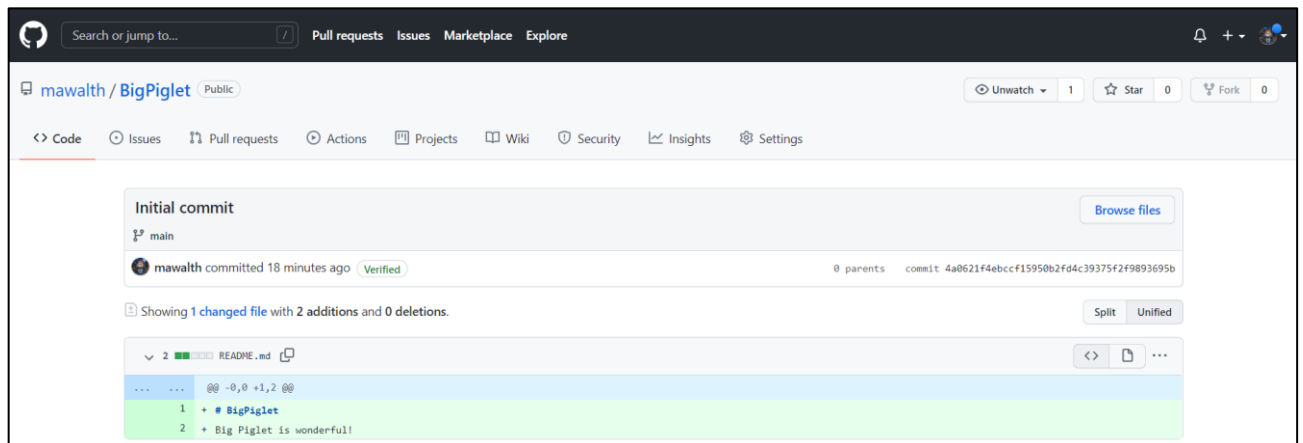
III.I. После нескольких кликов по модулю «Внешний код» попадаем в описание и изучаем внешний код.



Имя функции	Общее время ЦП...	Собственное вре...	Модуль
WP_1_1.exe (ИП: 19240)	18 (100.00%)	0 (0.00%)	WP-1.1.exe
[Собственный]	18 (100.00%)	14 (77.78%)	Несколько моду...
[Внешний код]	4 (22.22%)	4 (22.22%)	Несколько моду...
WP_1_1.Program.Main(System.String[])	4 (22.22%)	0 (0.00%)	WP-1.1.dll

IV. Загрузка рабочего пакета данных в репозиторий GitHub.

IV.I. Создаем репозиторий в GitHub и загружаем туда все файлы, используемые в работе. Тип репозитория «Public».



Итог: Работа проделана. Работа завершена.