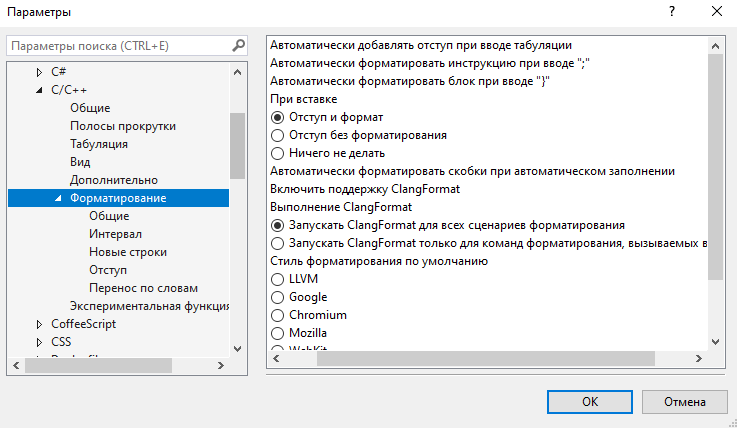
**Настройка внешнего вида и функциональности среды разработки Visual Studio**

**1.1** Параметры текстового редактора

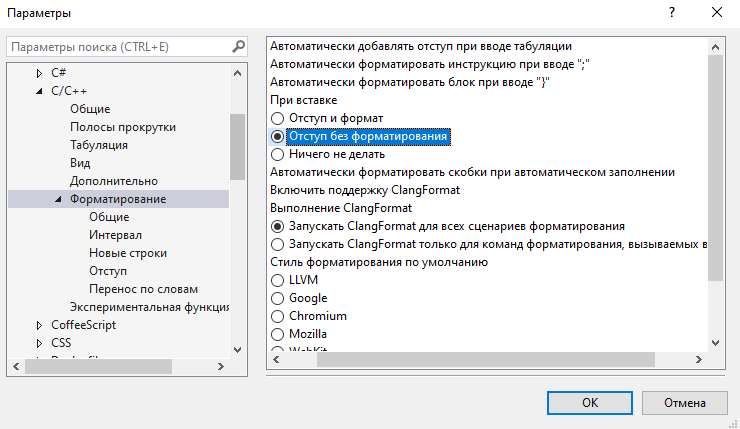
В параметрах для выбранного языка программирования изменить параметры форматирования для вставки. Найти код в интернете для выбранного языка и вставить в программу при разных значениях.

Продемонстрировать автоматическое формирование кода при вводе } и ; убрать автоматическое форматирование.

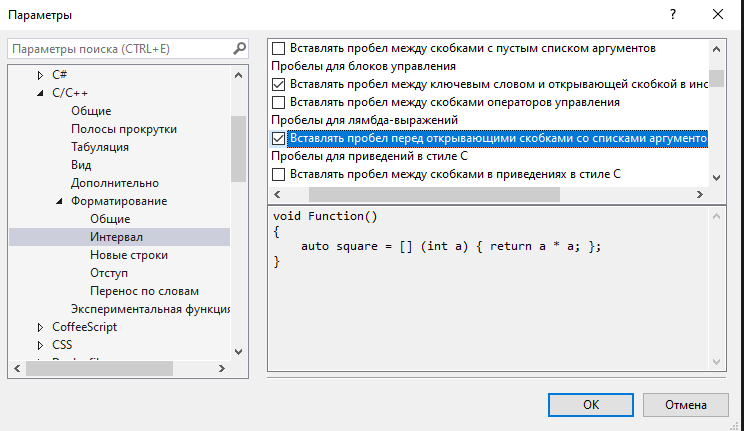
**До:**

****

**После:**

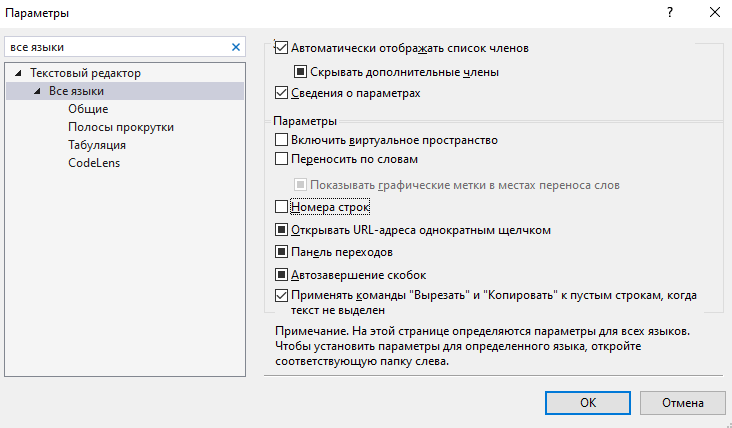


Во вкладке «форматирование» - «интервал» продемонстрировать изменение форматирования при установлении/снятии флагов.

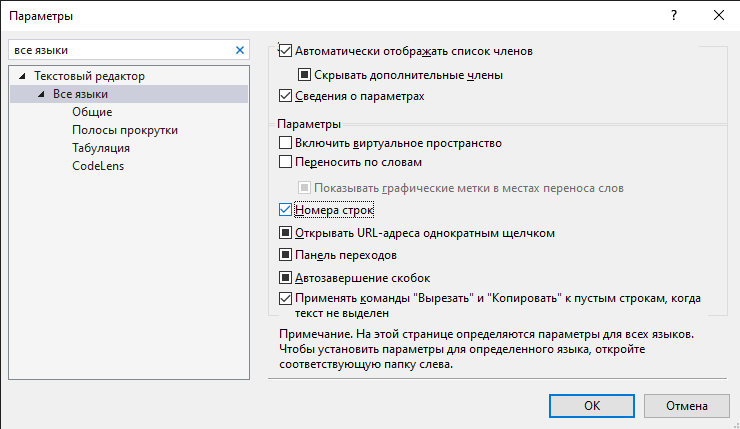


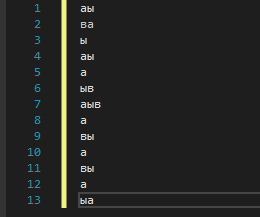
Отображение нумерации строк изменить и продемонстрировать

**До:**



**После:**

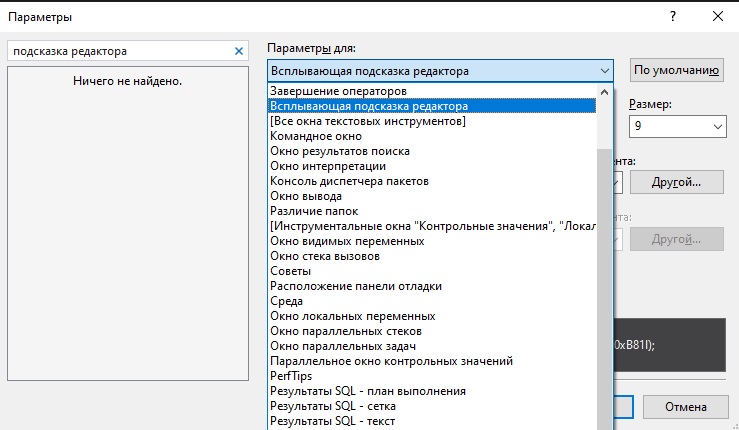




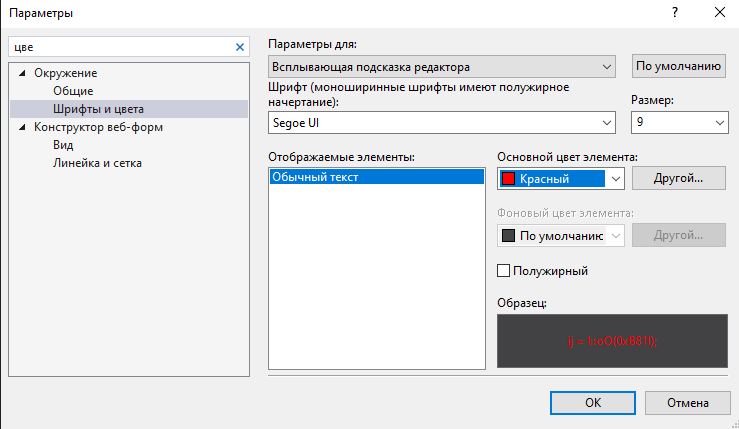
**1.2 Изменение шрифтов и цветов**

**Изменить параметр всплывающей подсказки в среде**

**До:**

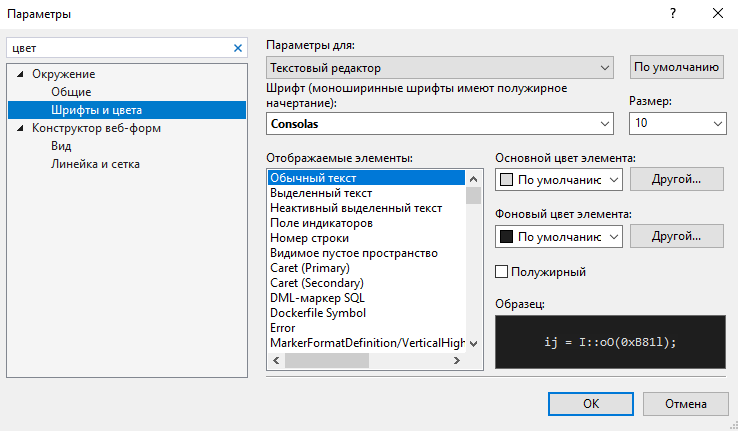
****

**После:**

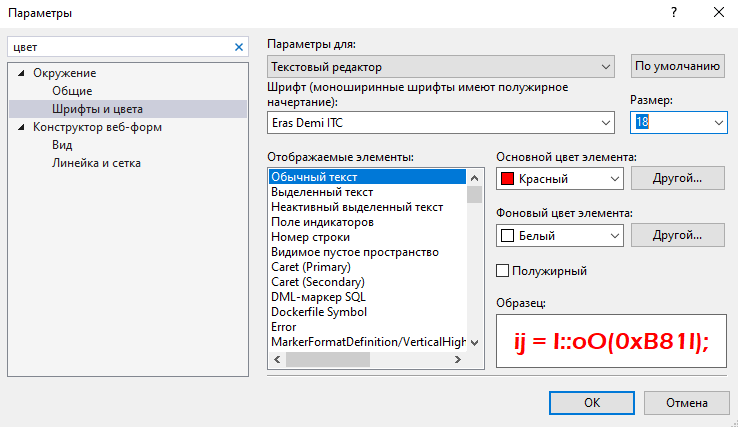
****

**Изменить параметр текста среды**

**До:**

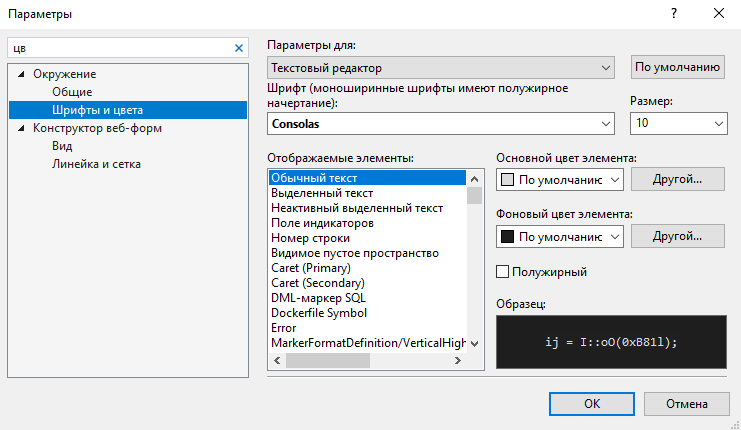
****

**После:**

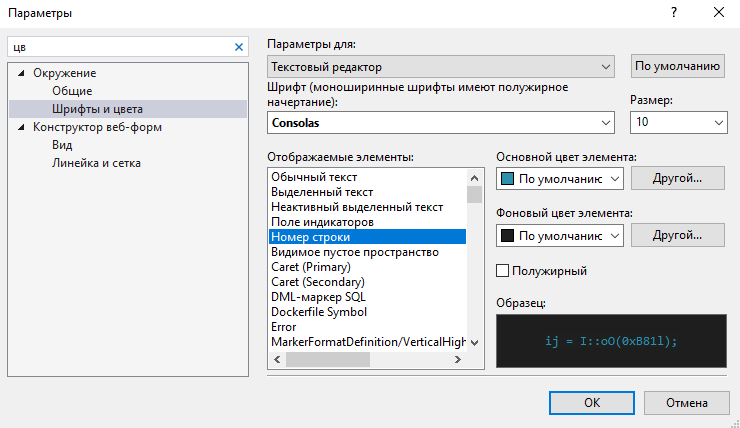
****

**Изменить параметр номера строки в текстовом редакторе**

**До:**

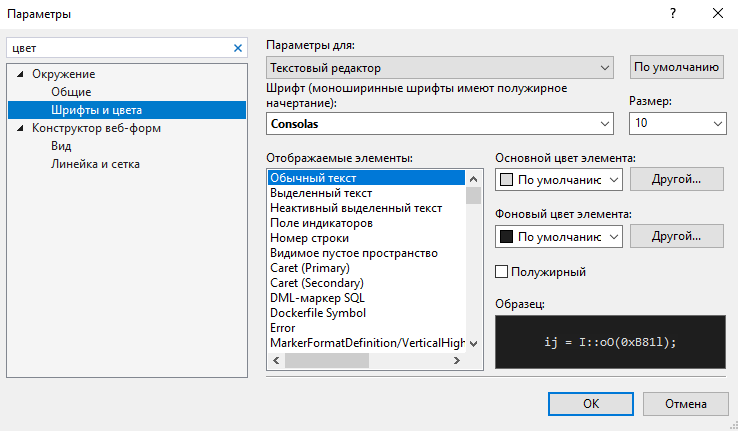
****

**После:**

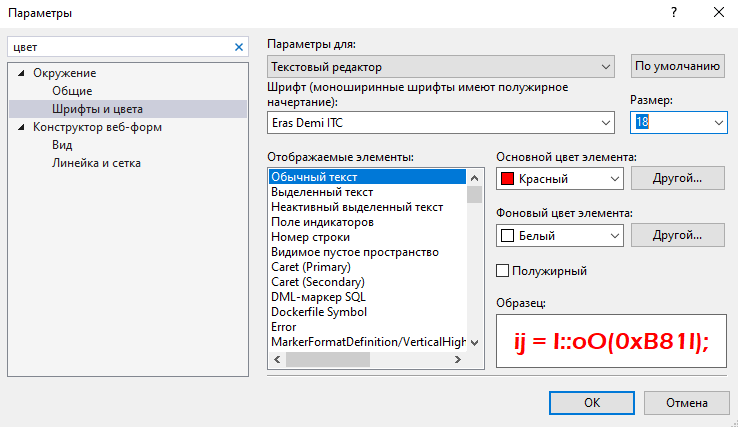
****

**Изменить параметр обычного текста в текстовом редакторе**

**До:**

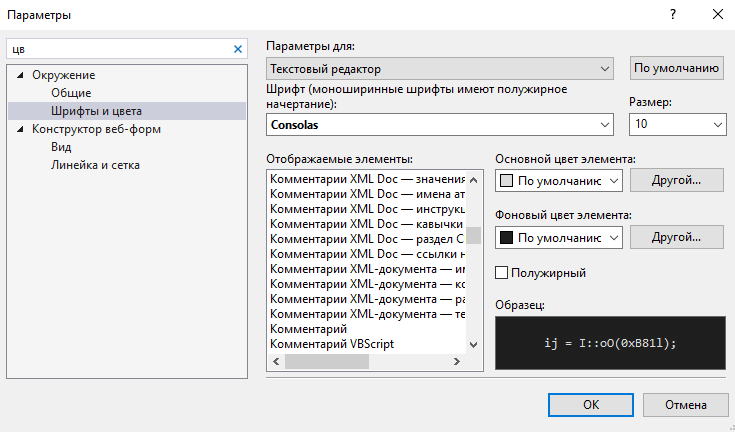
****

**После:**

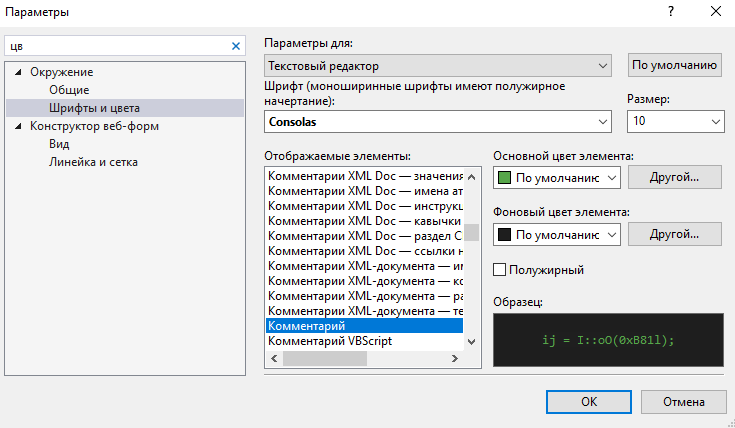
****

**Изменить параметр комментарий в текстовом редакторе**

**До:**

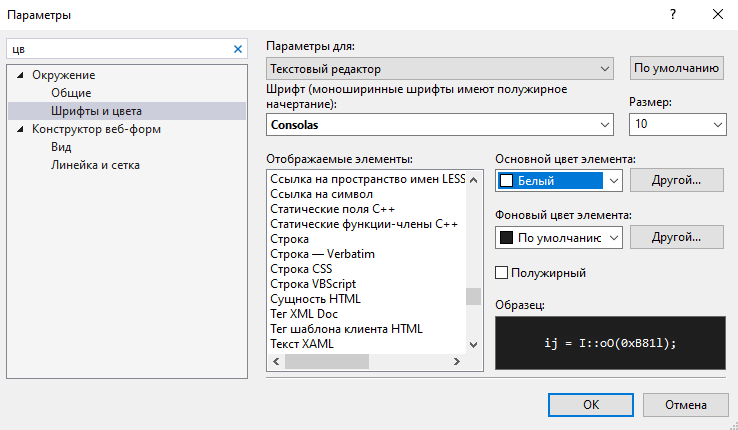
****

**После:**

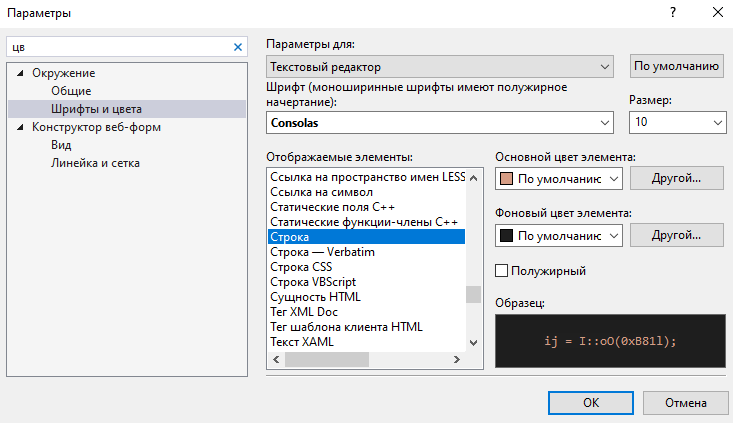
****

**Изменить параметр строк в текстовом редакторе**

**До:**

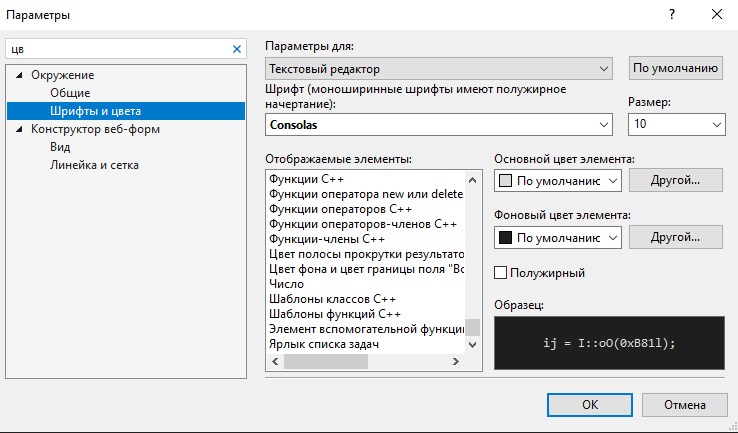
****

**После:**

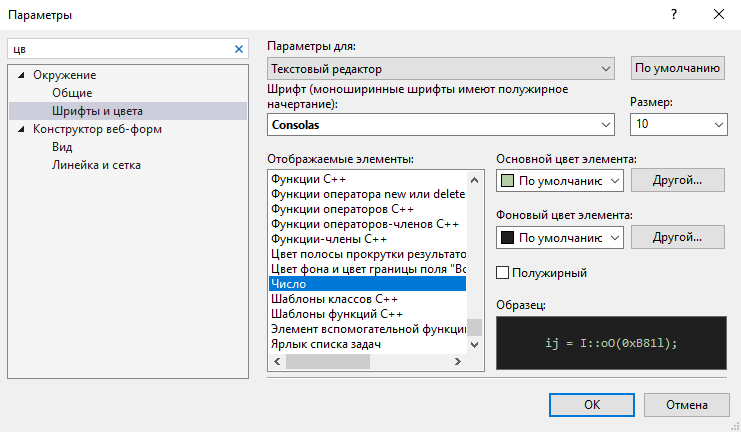
****

**Изменить параметр чисел в текстовом редакторе**

**До:**

****

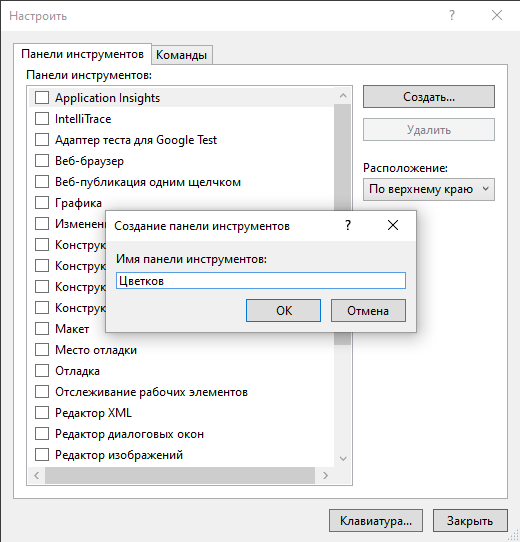
**После:**

****

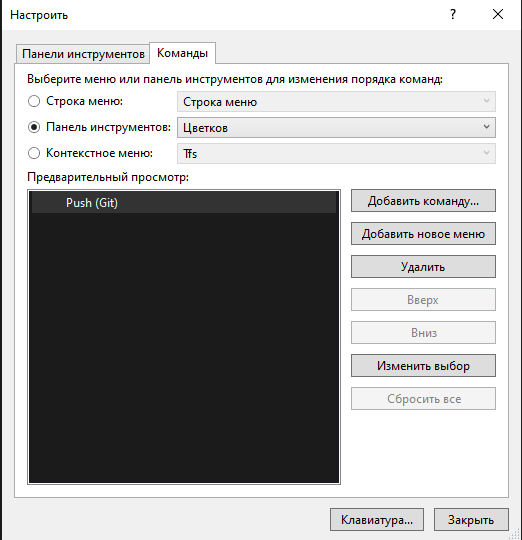
**1.3. Настройка меню и панели инструментов**

Создать свою панель инструментов, назвать ее своей Фамилией. Разместить на ней несколько команд. Создать горячие клавиши.

**До:**

****

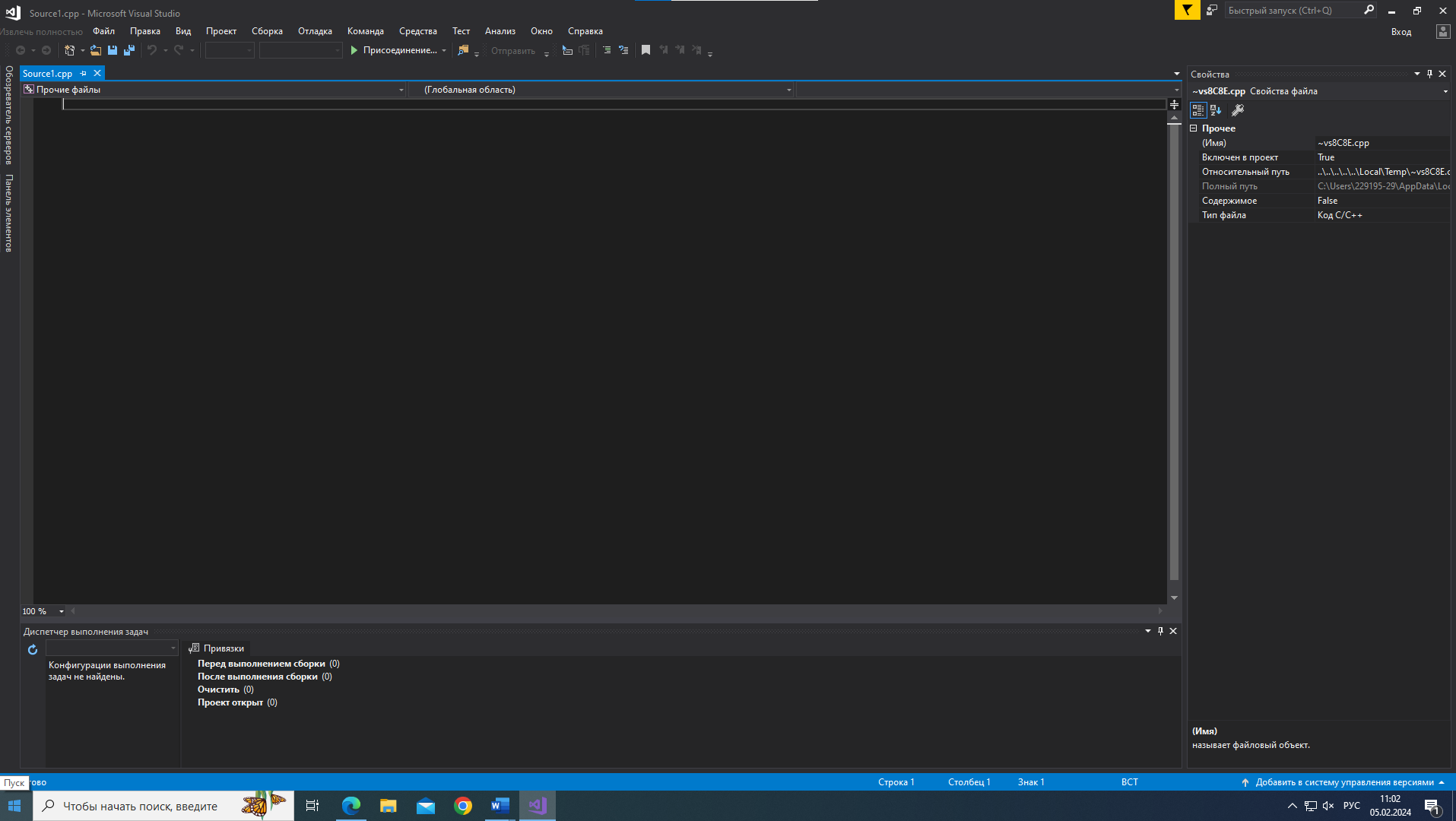
**После:**

****

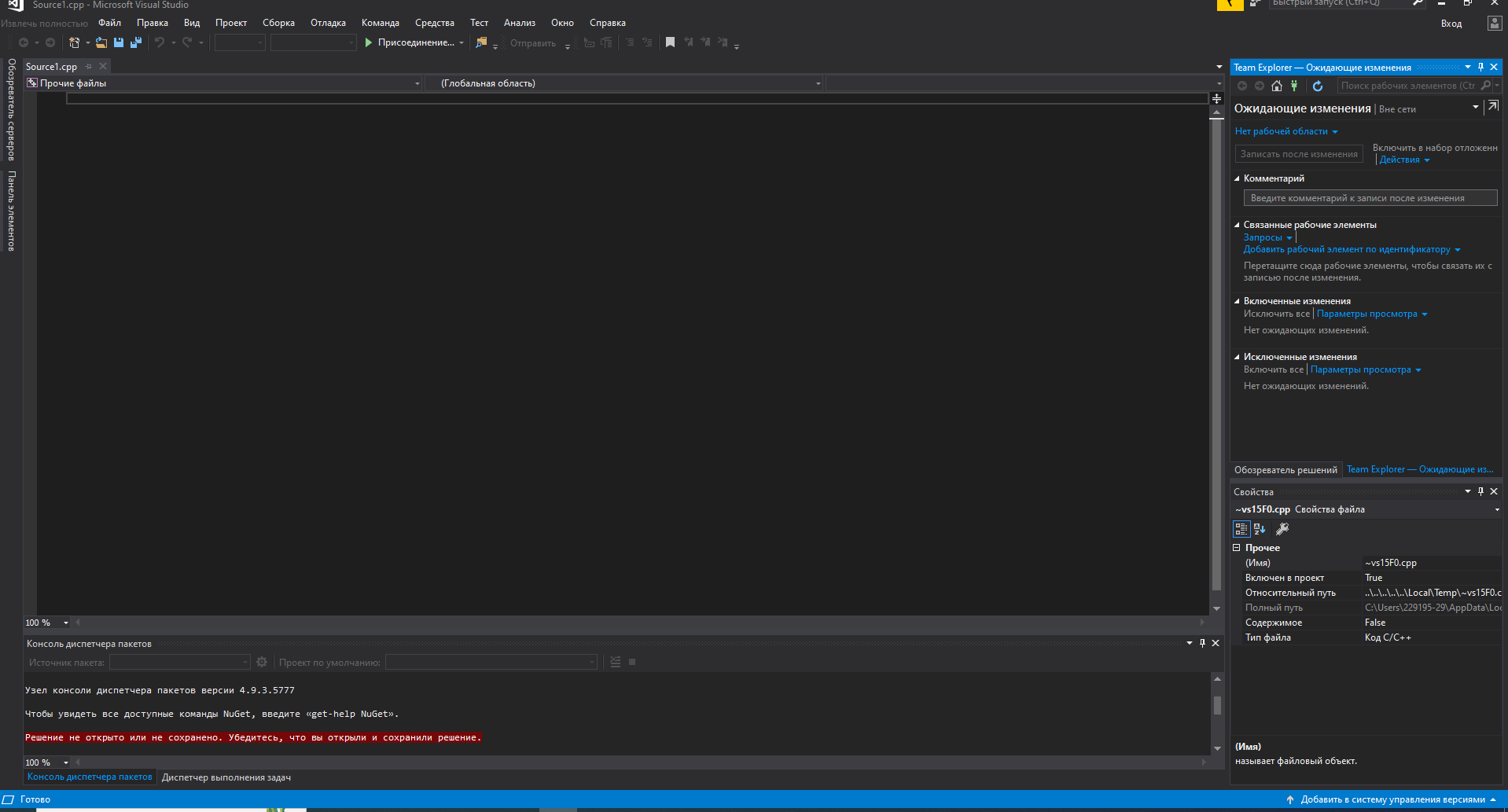
**1.4. Настройка макетов окон**

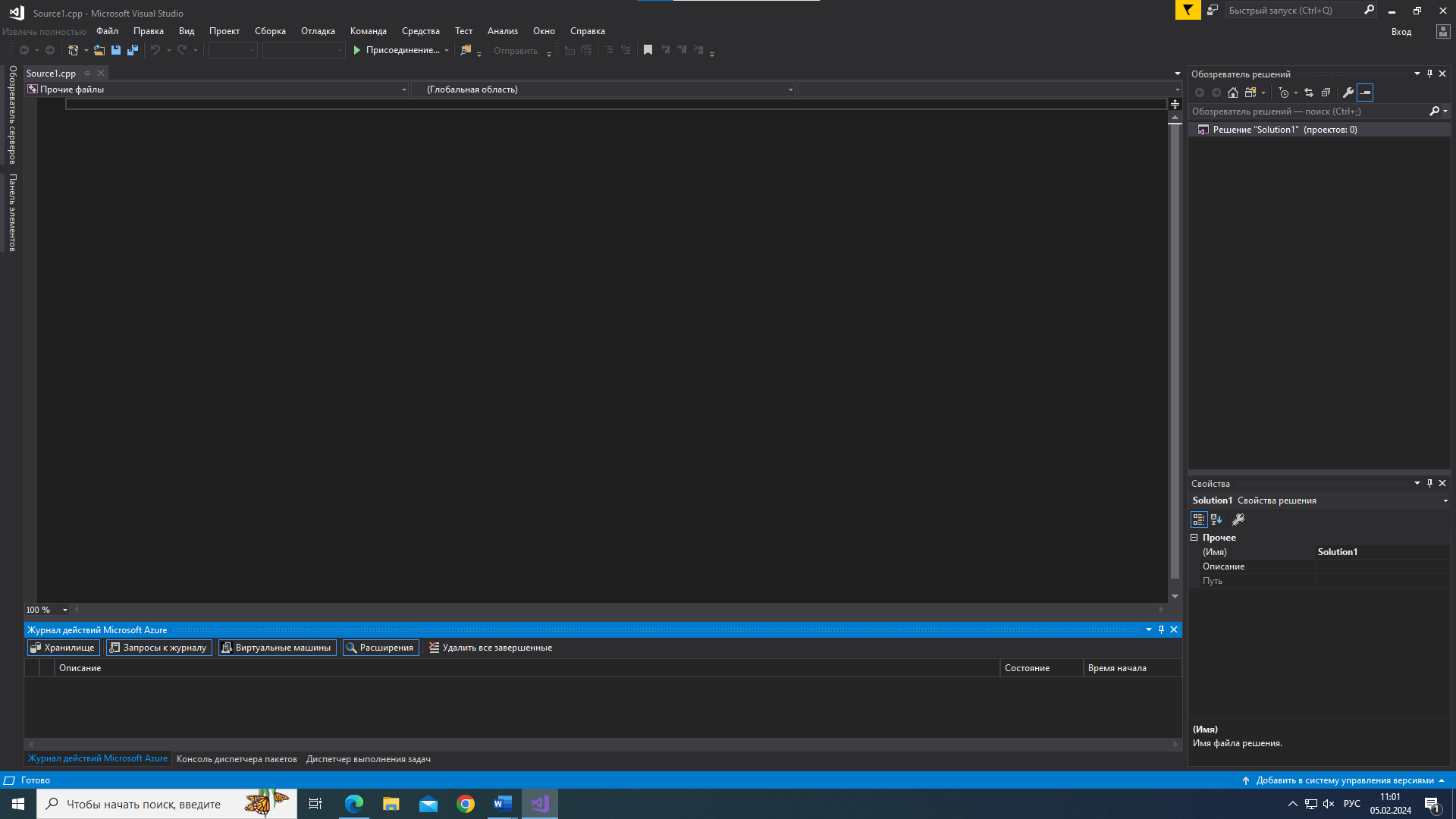
В меню «вид» -> «другие окна» добавить несколько окон. Сделать свой макет, назвать своей фамилией. Продемонстрировать переключение между макетами и управление макетами.

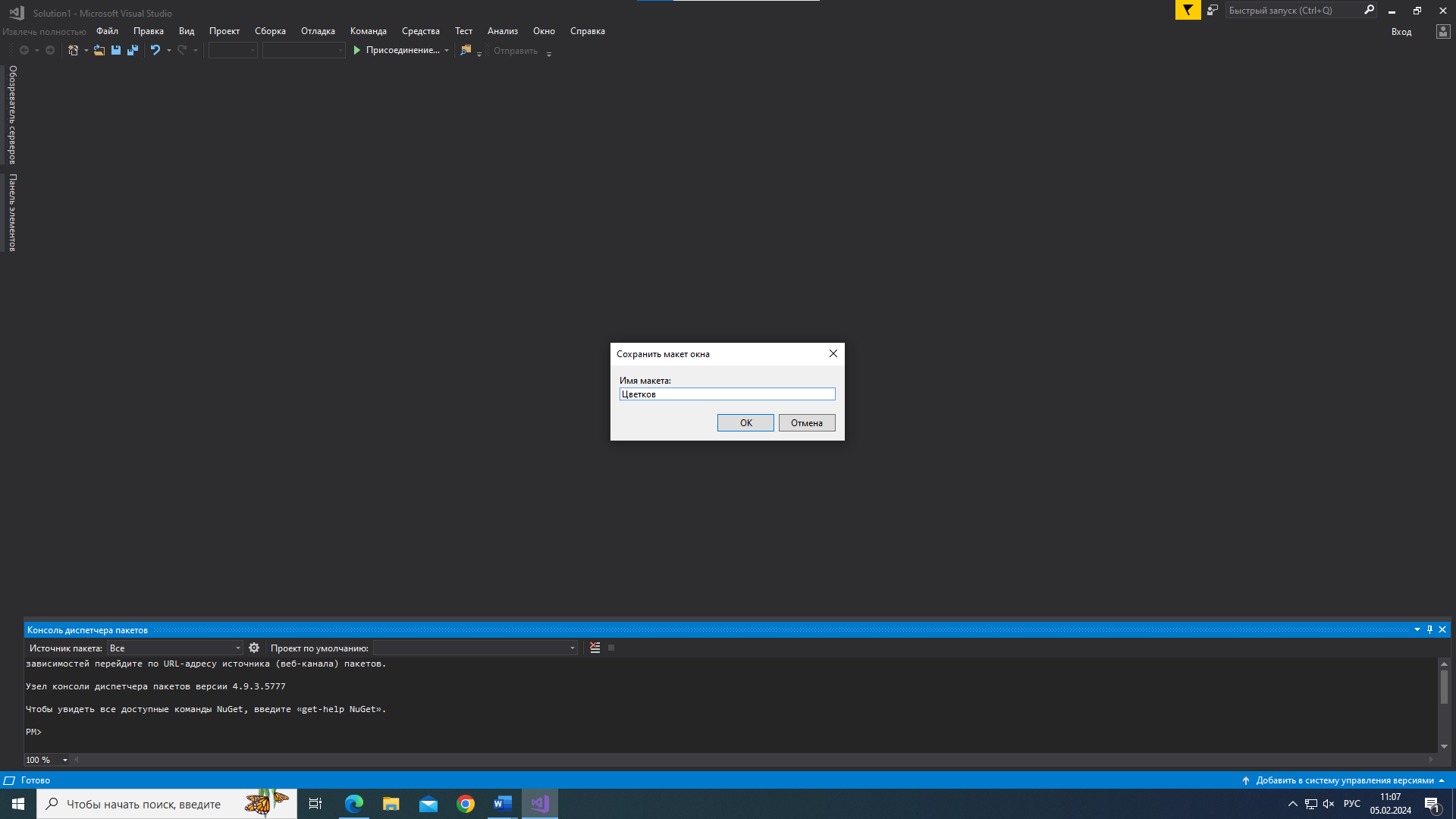
**До:**

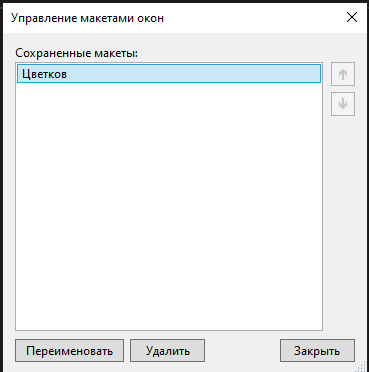


**После:**





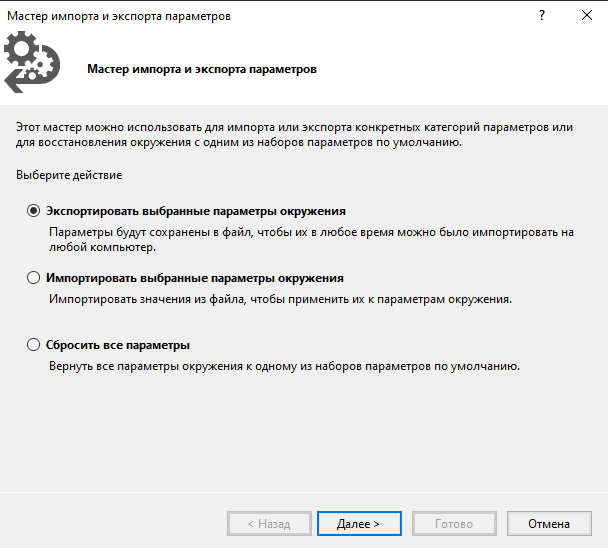


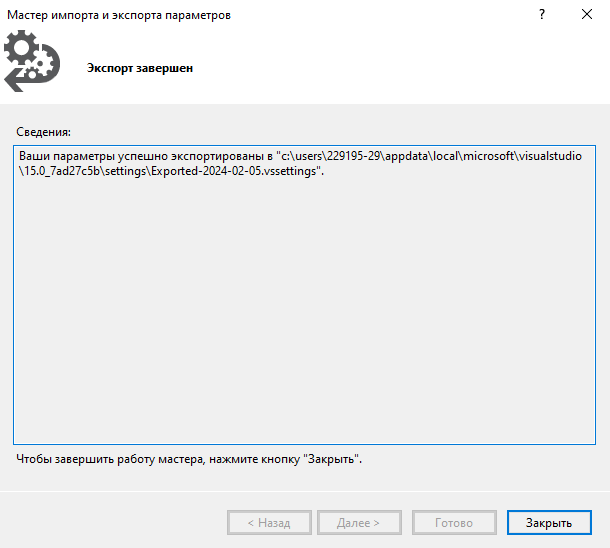


* 1. **Экспорт настроек**

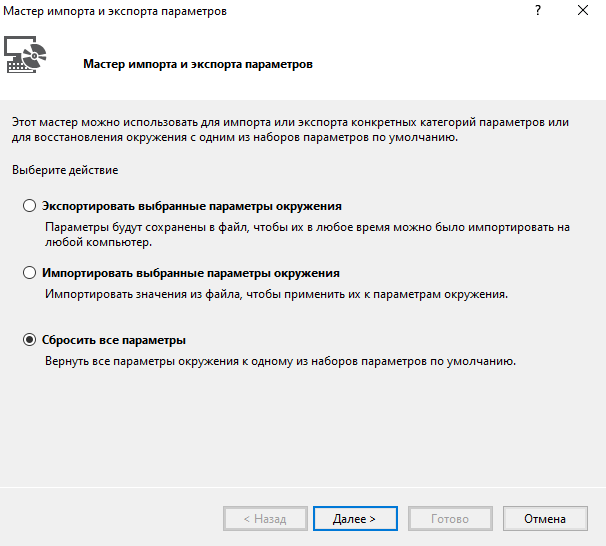
С помощью меню «Импорт и экспорт параметров»

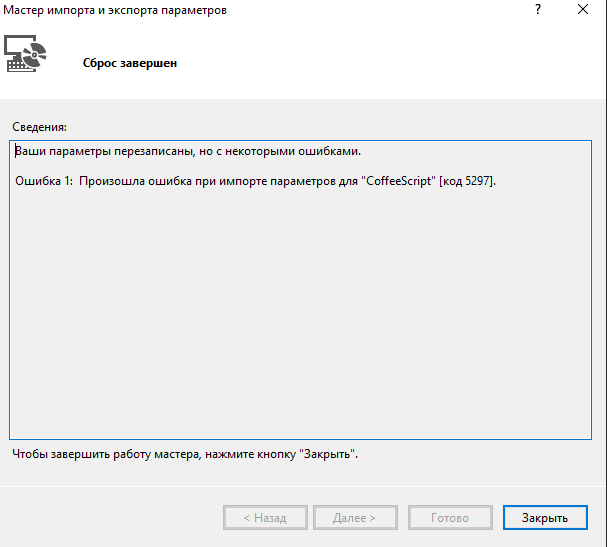
* **Экспортировать настройки**



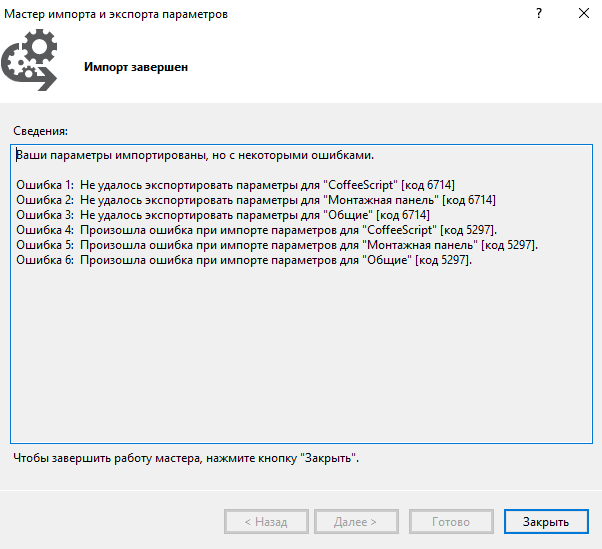


* **Сбросить настройки**



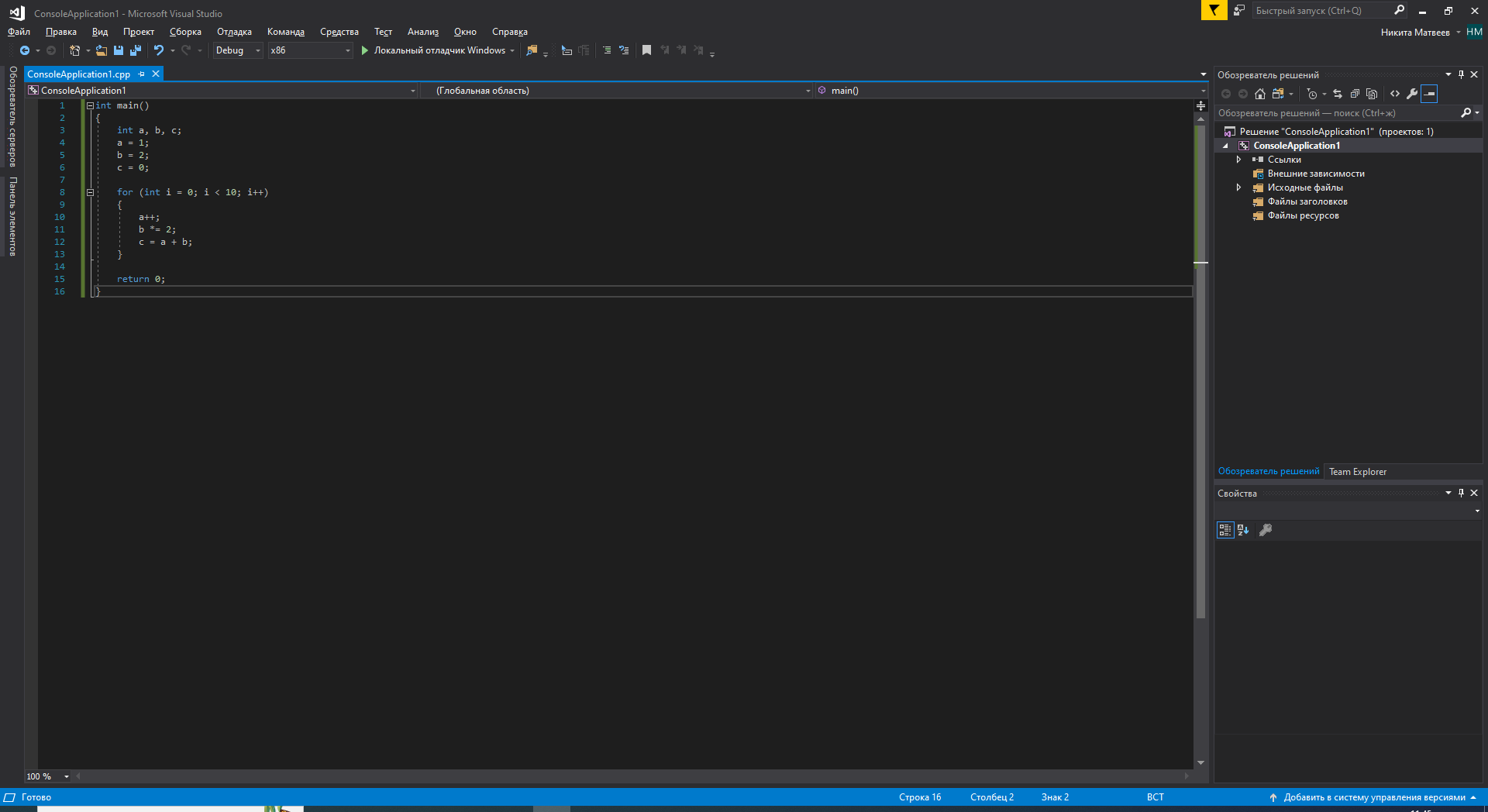


* **Импортировать выполненные ранее настройки**

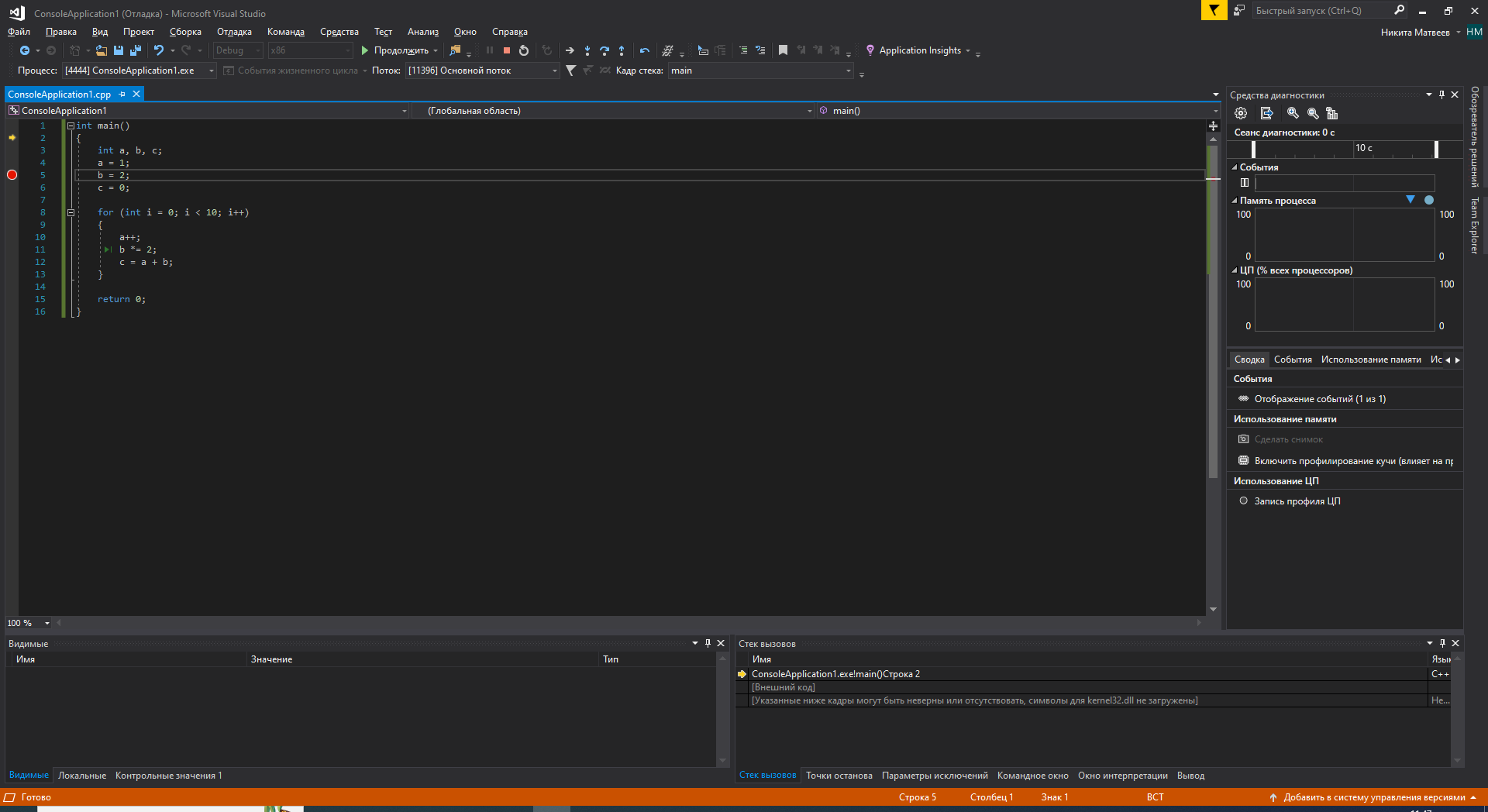


# **2.1. Навигация по коду с помощью отладчика**

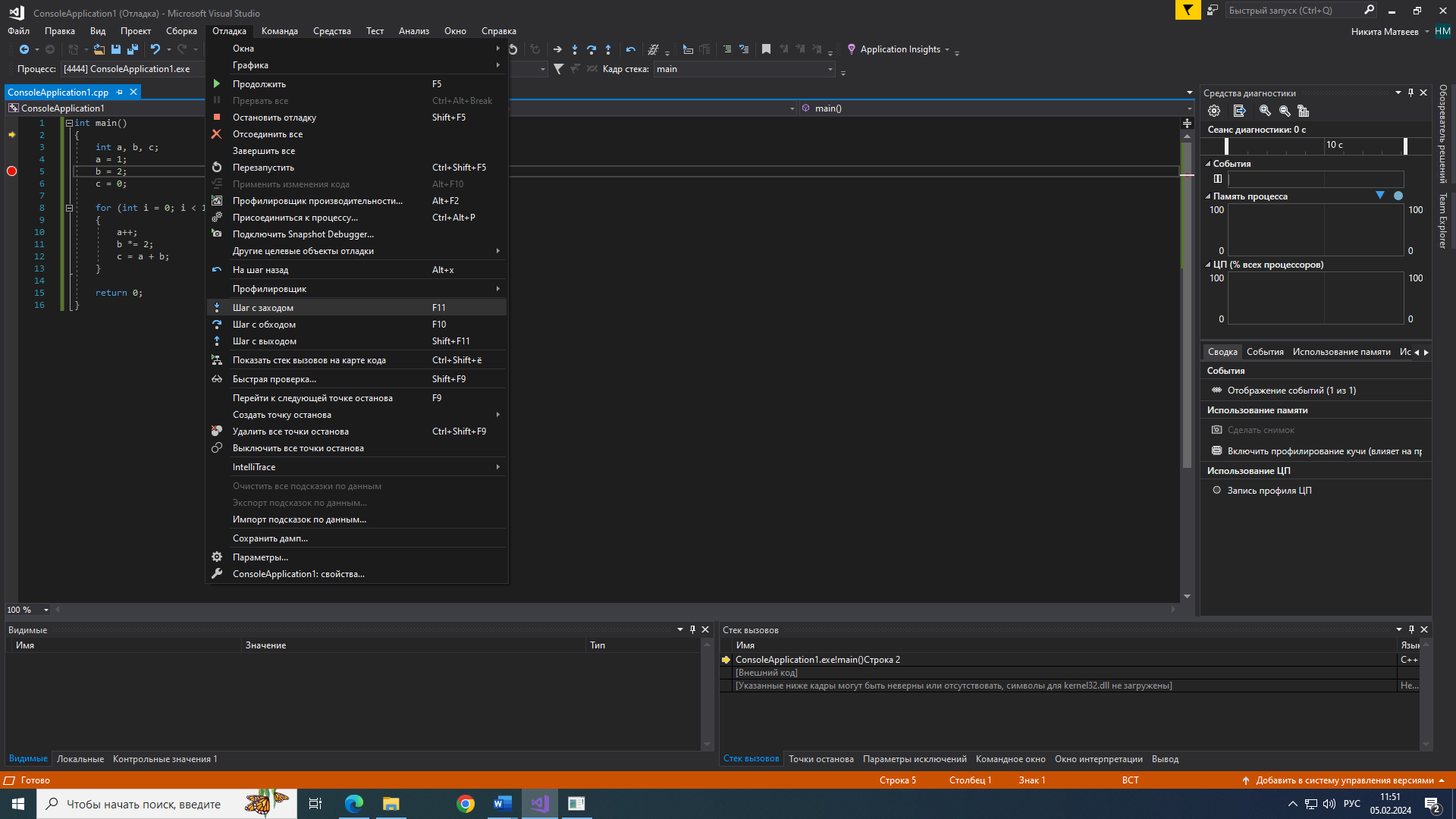
1. Проект:



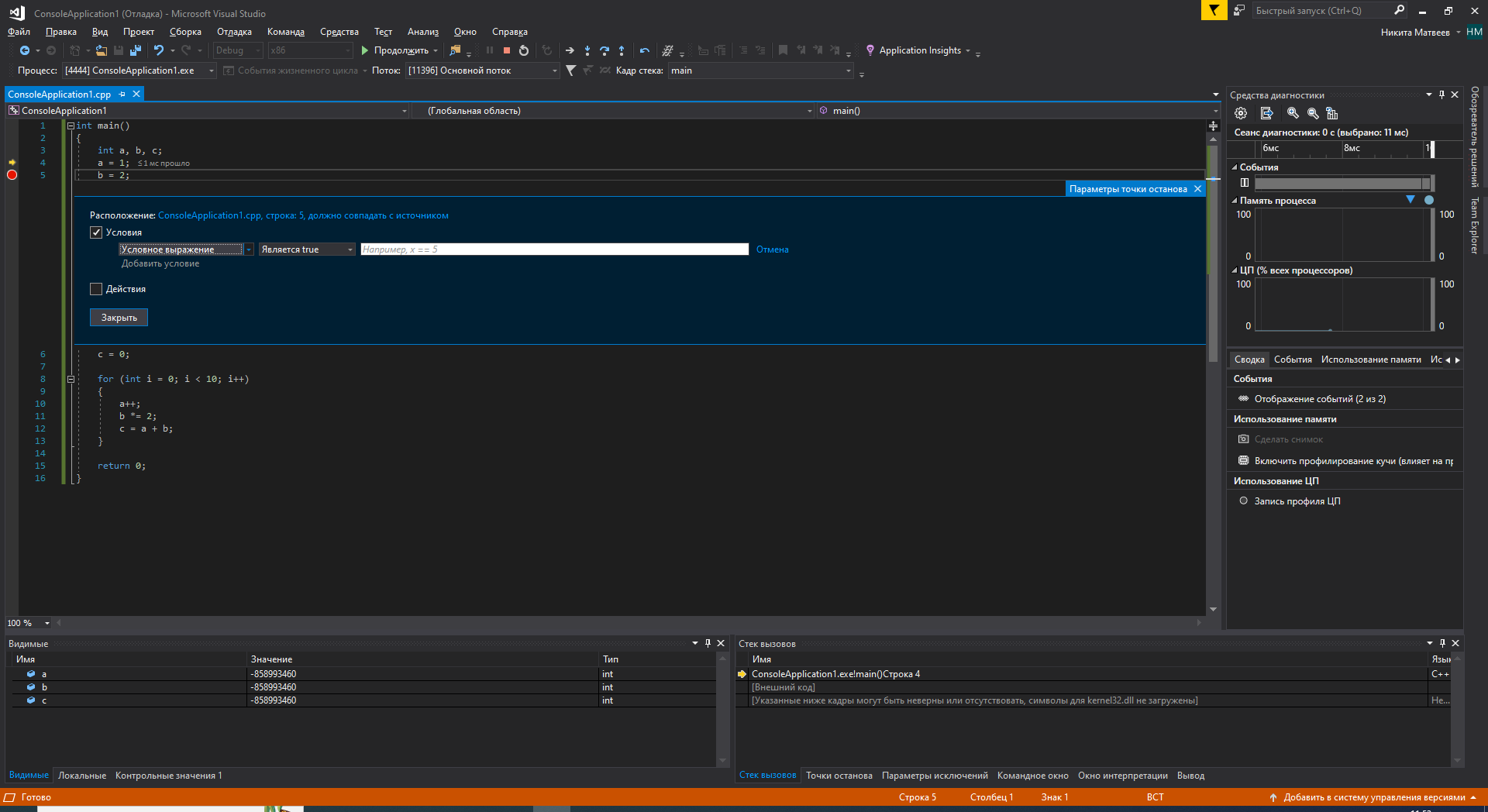
1. Установка точки остановки:



1. Пройти программу по шагам

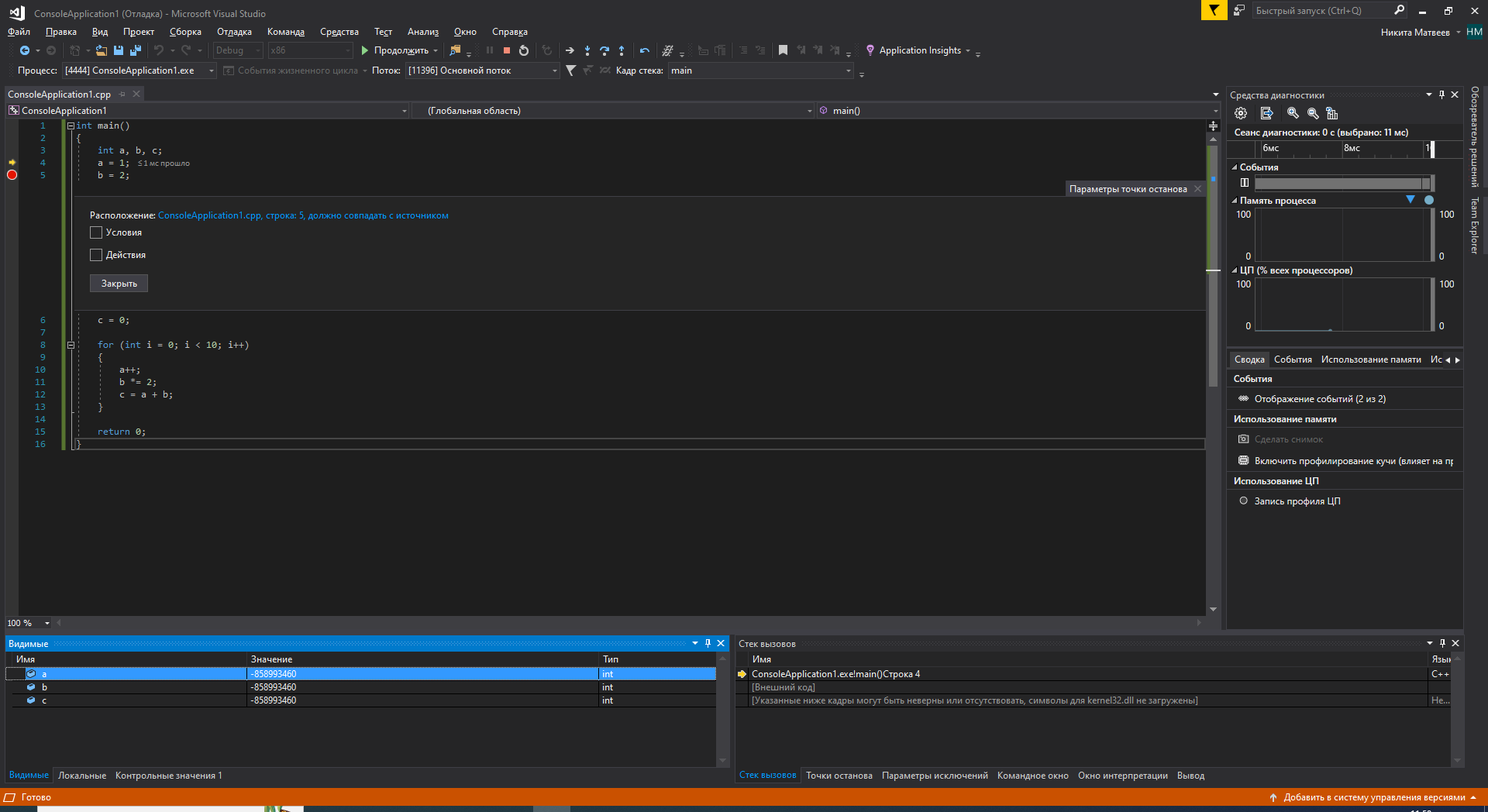


1. В цикле выполнить «точку остановки с условием»

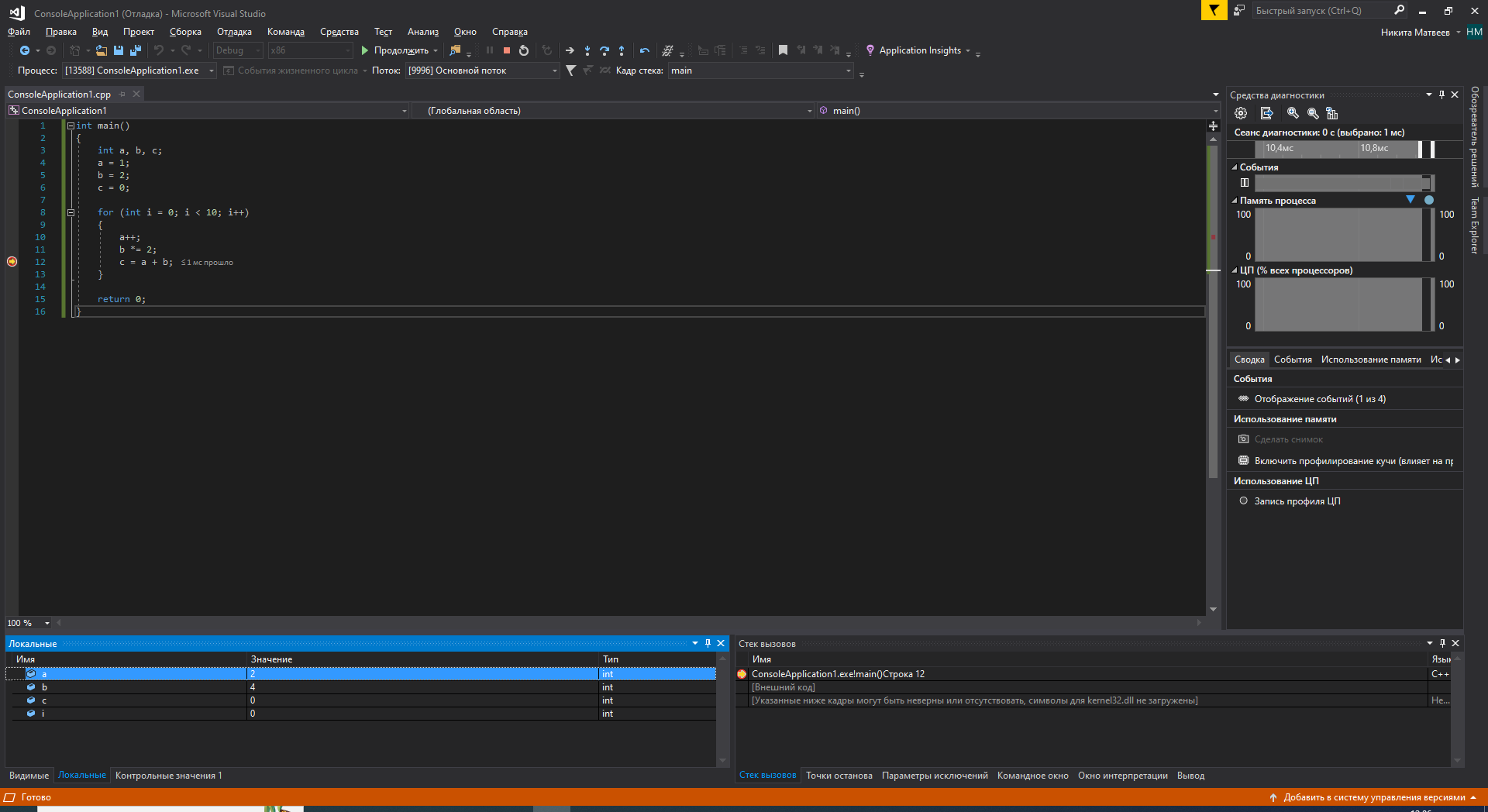


1. Исследовать изменения переменных в окнах "Видимые" и "Локальные"

**Видимые:**

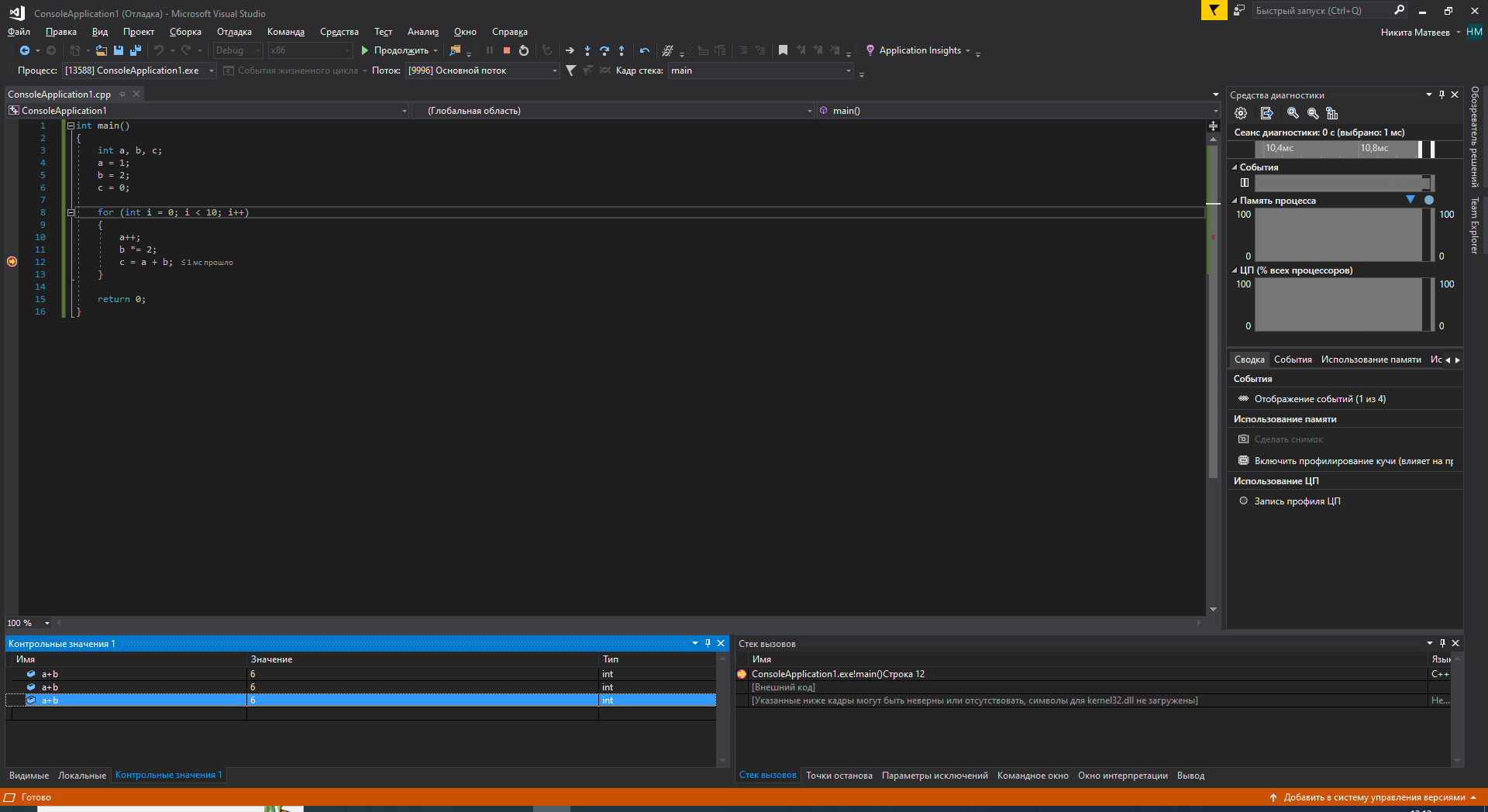


**Локальные:**

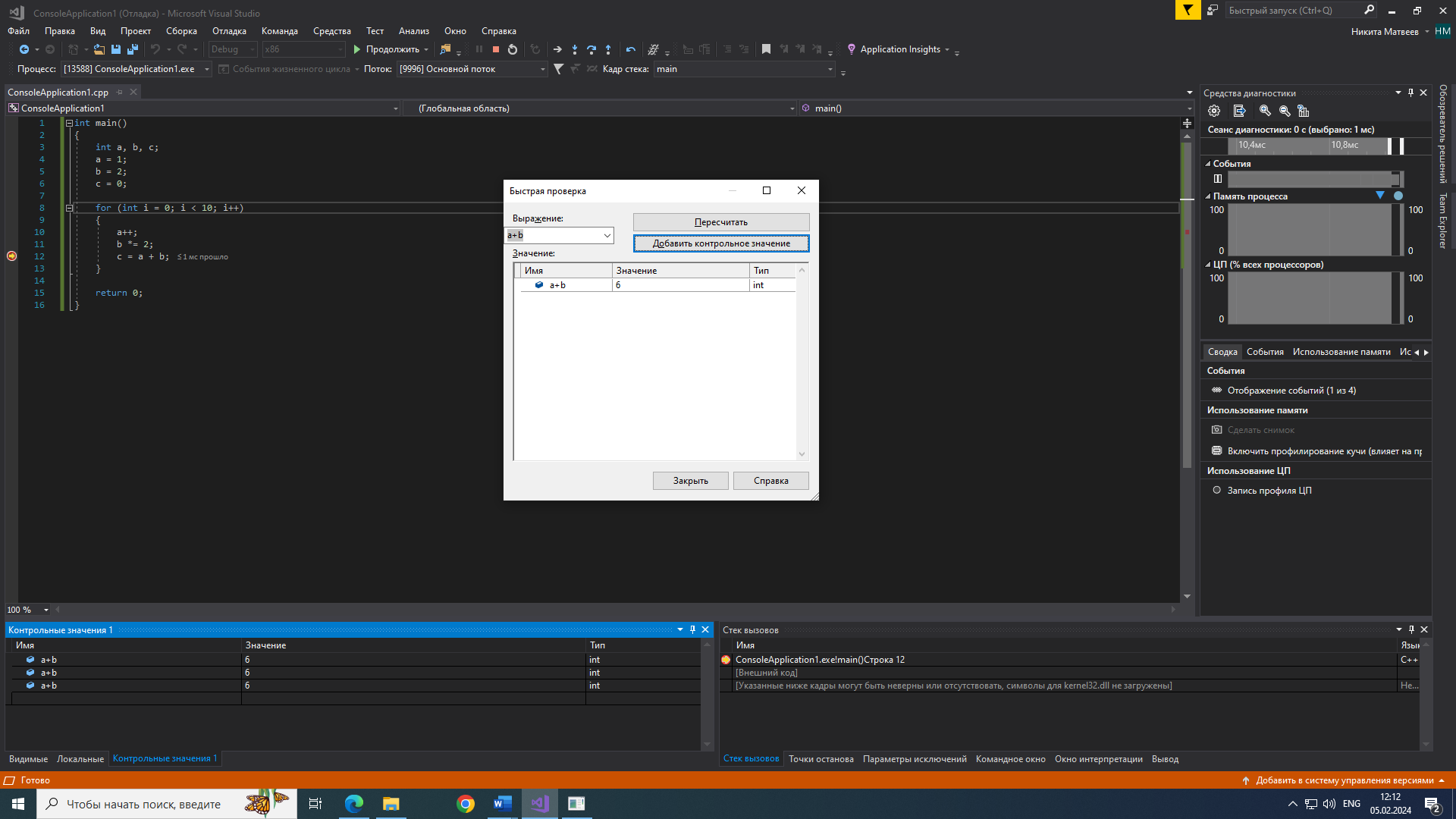


1. Продемонстрировать работу «Контрольных значений» и «Быстрой проверки»

**Контрольные значения:**

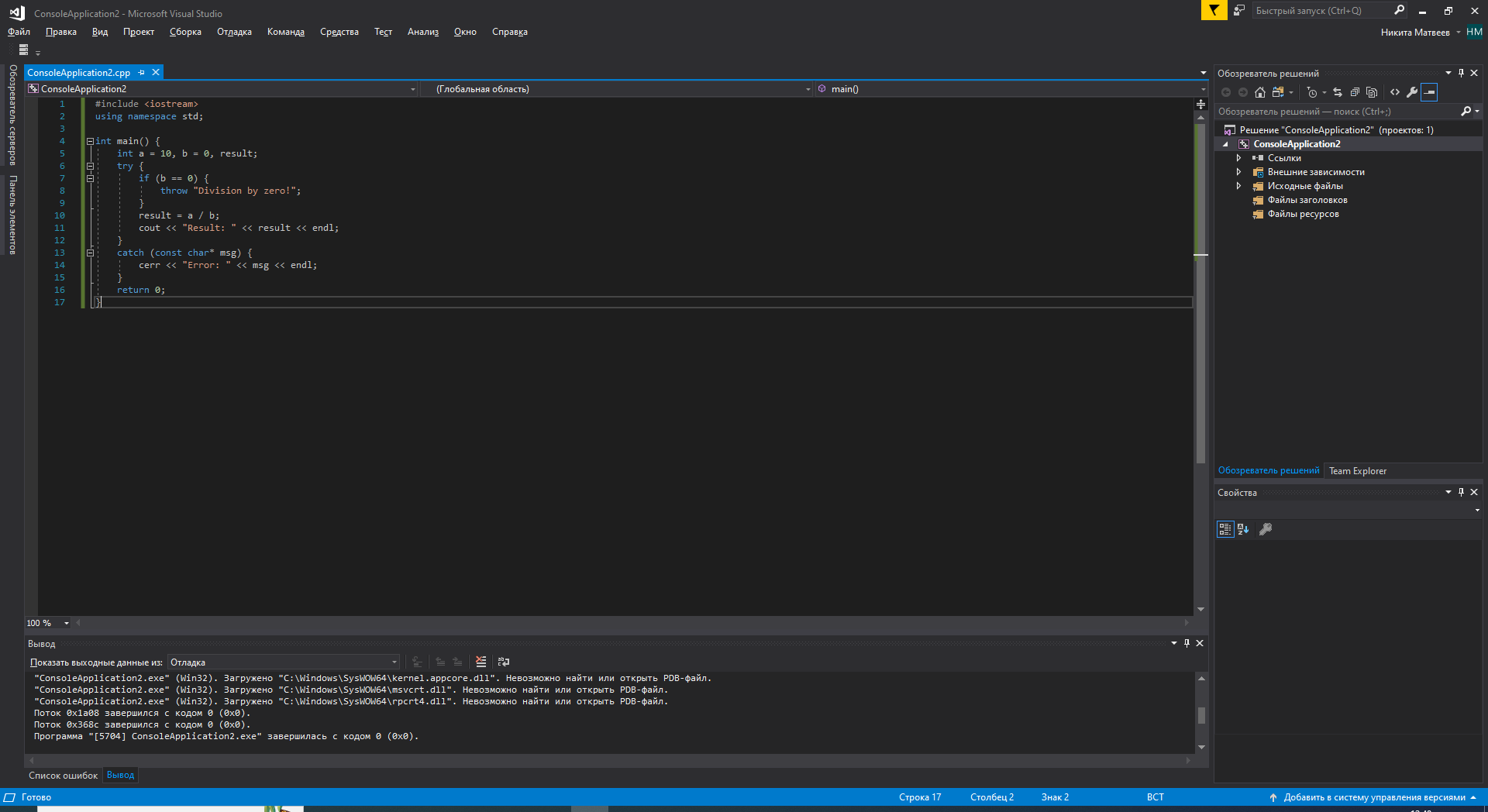


**Быстрая проверка:**

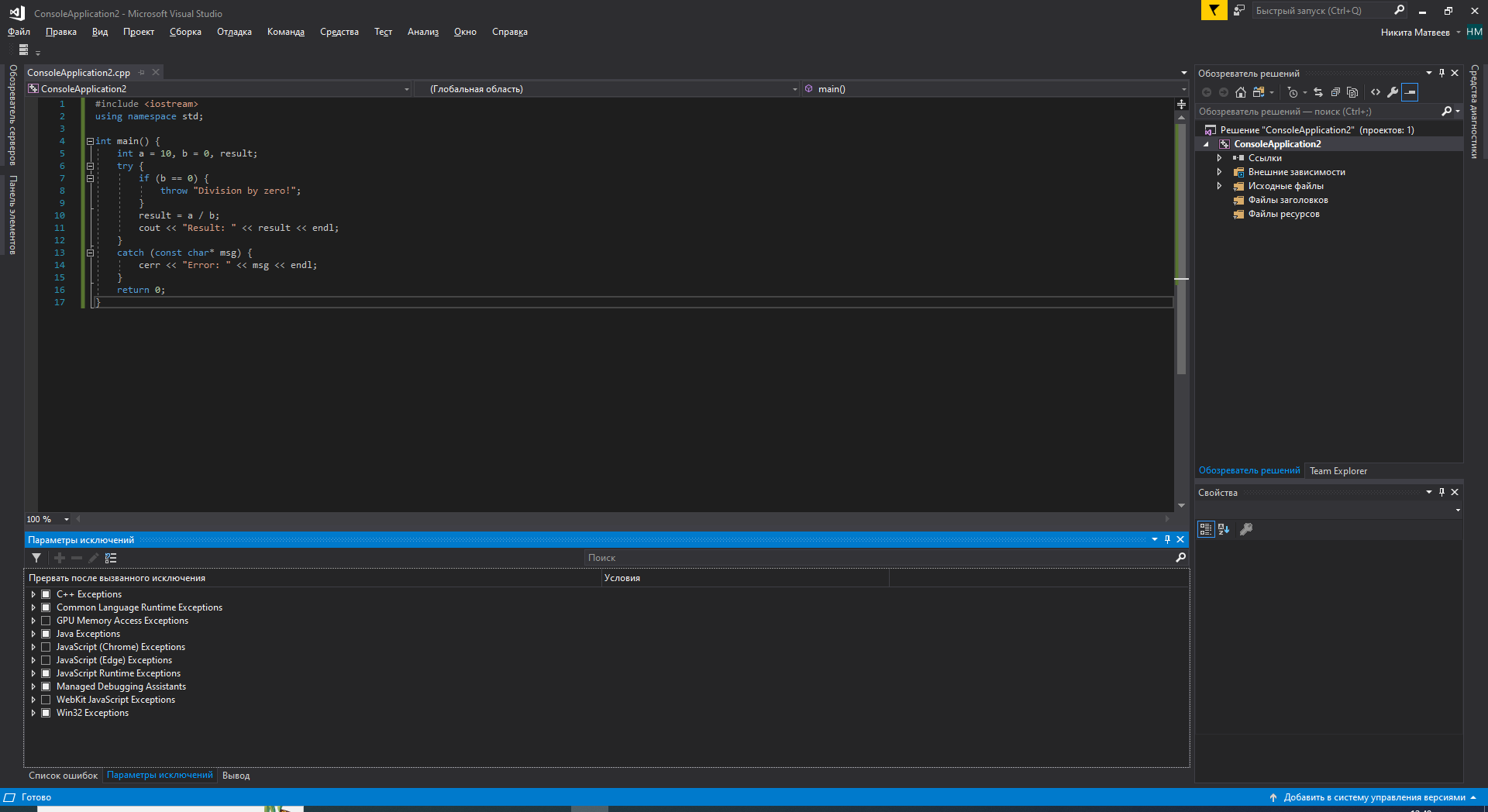


# **2.2 Управление исключениями с помощью отладчика**

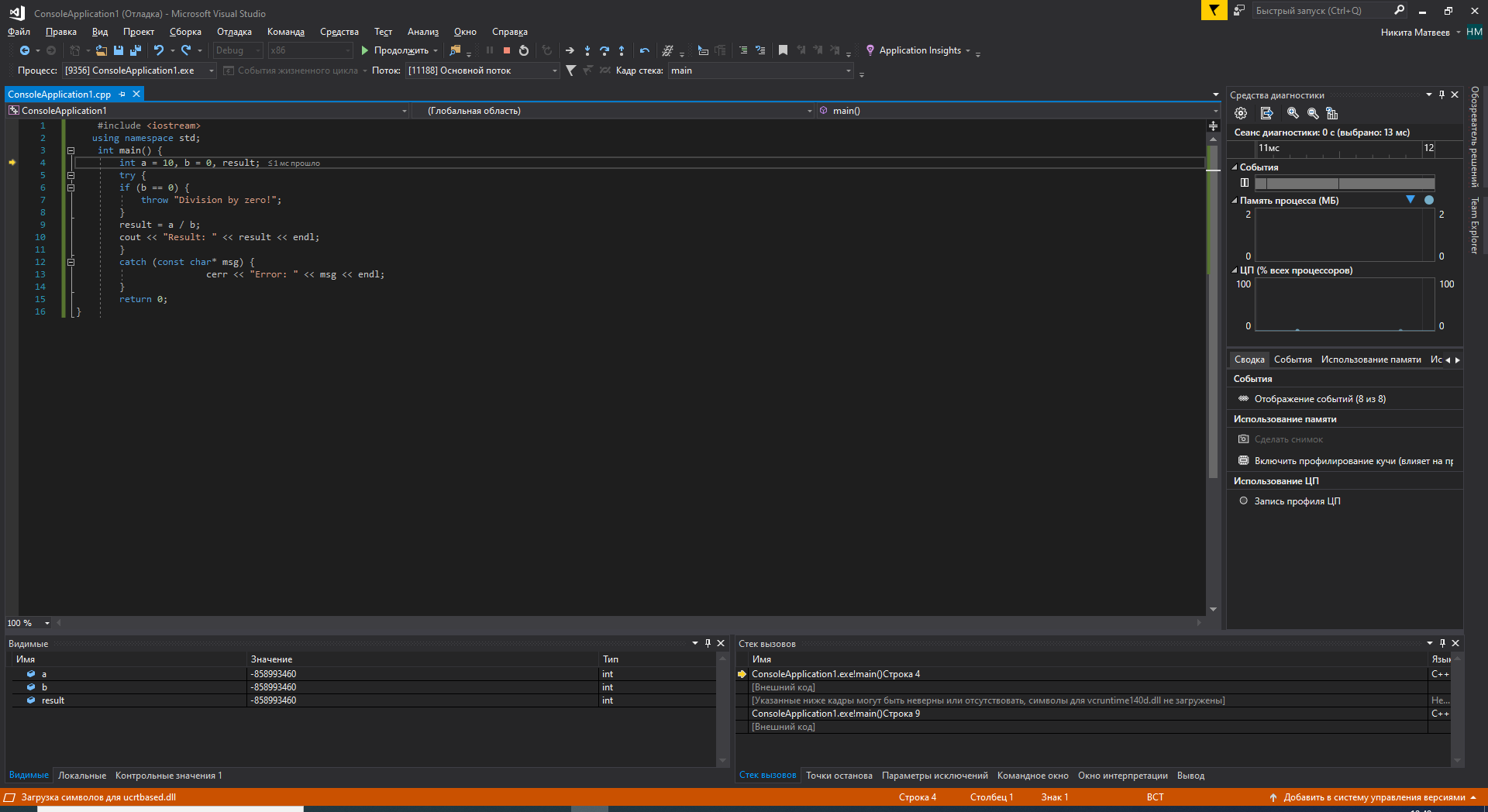
1. Изменить код, добавив в него исключение и обработчик исключения

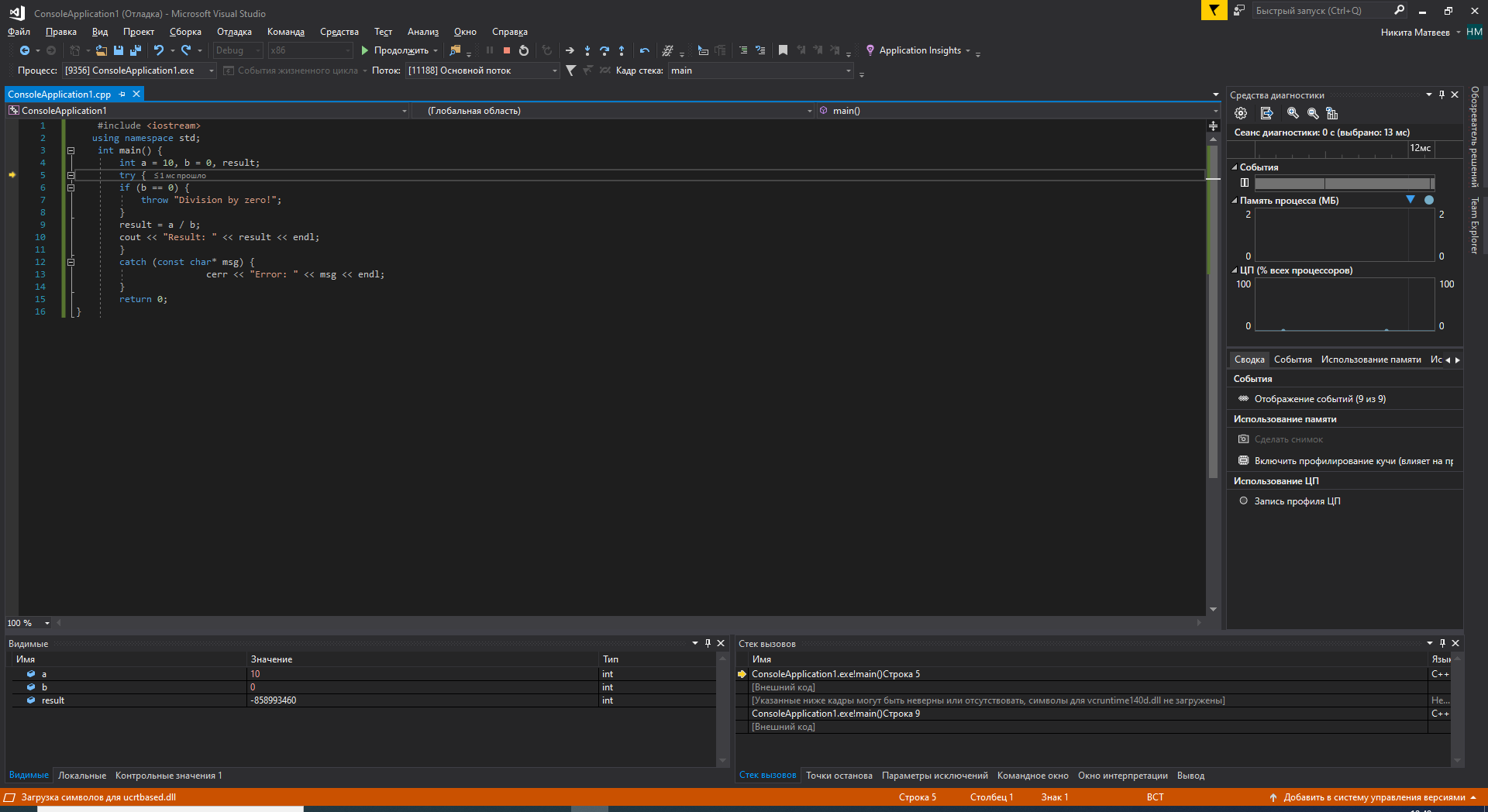


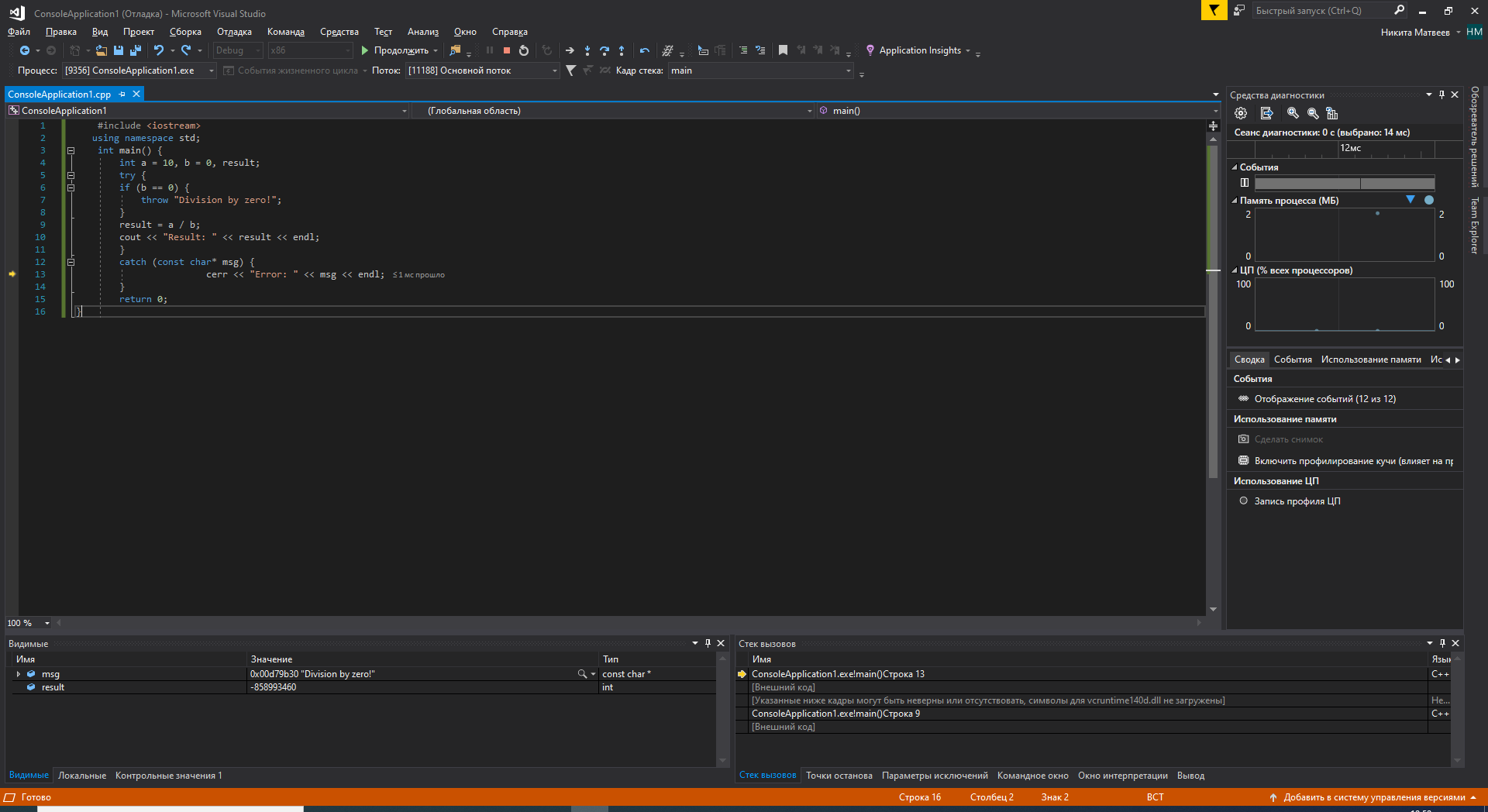
1. Настроить отладчика для прерывания выполнения при создании исключения (Отладка-Окна-Параметры исключений (только когда код запущен!))



1. Продемонстрировать работу отладчика

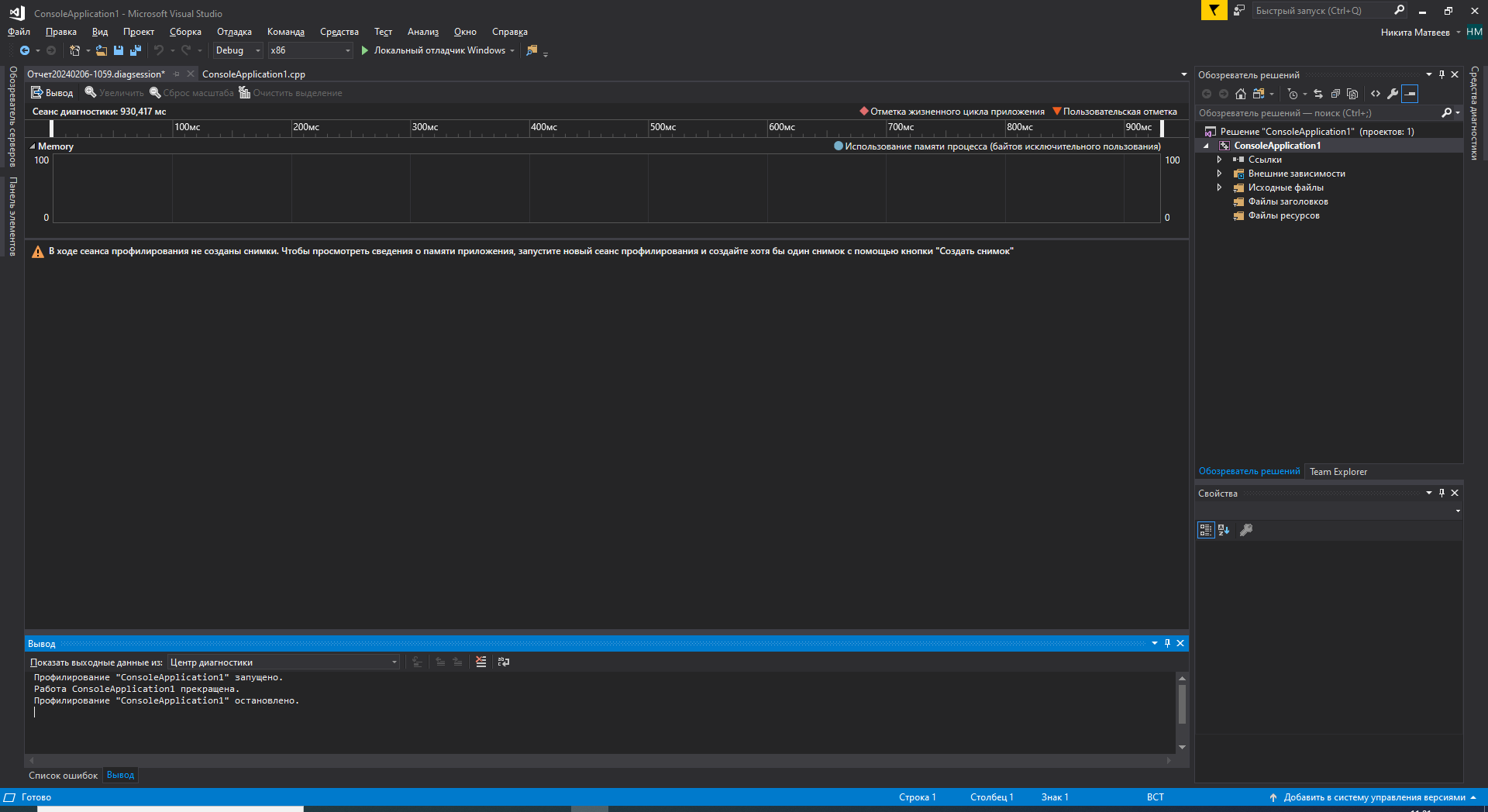




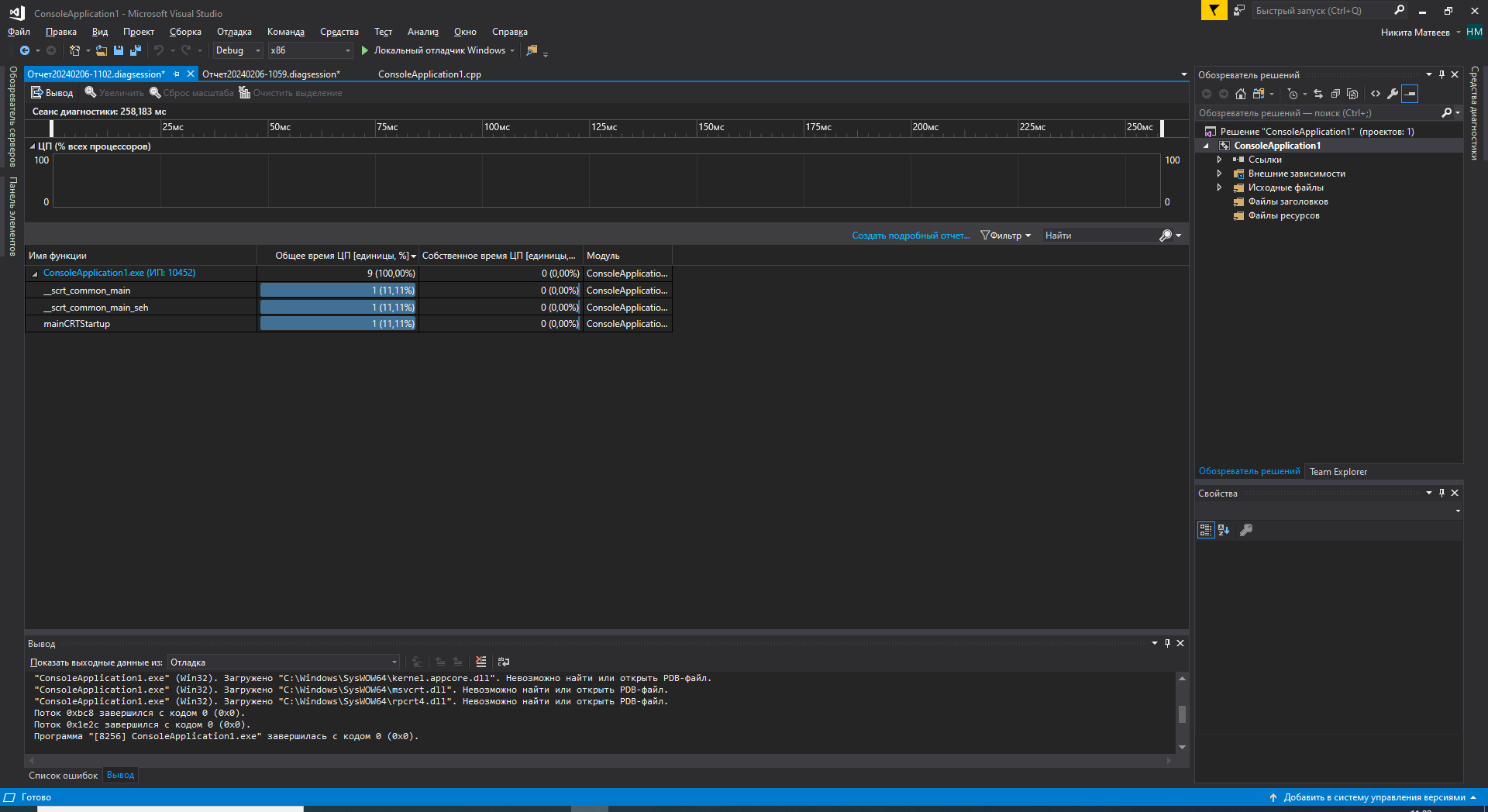


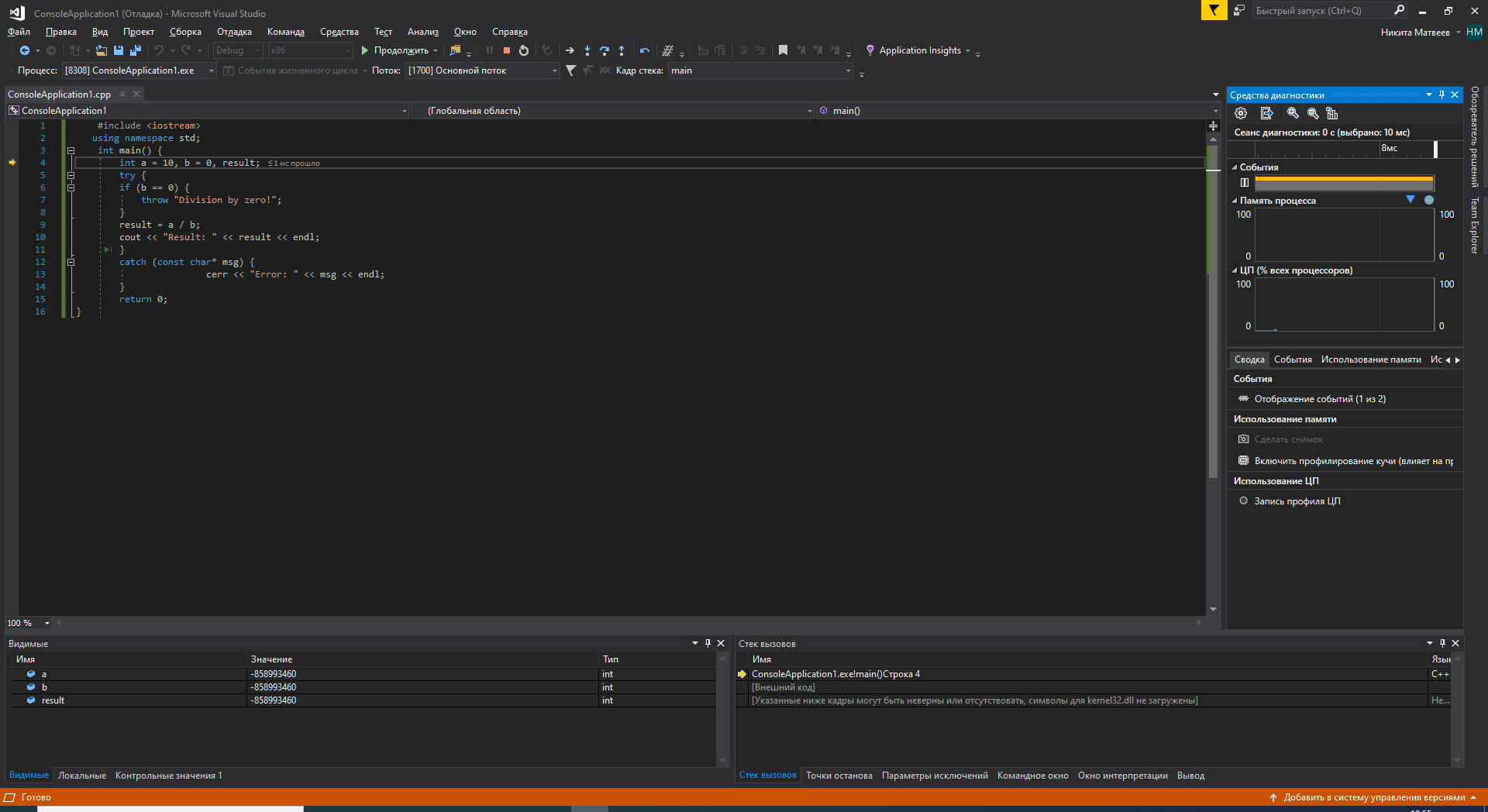
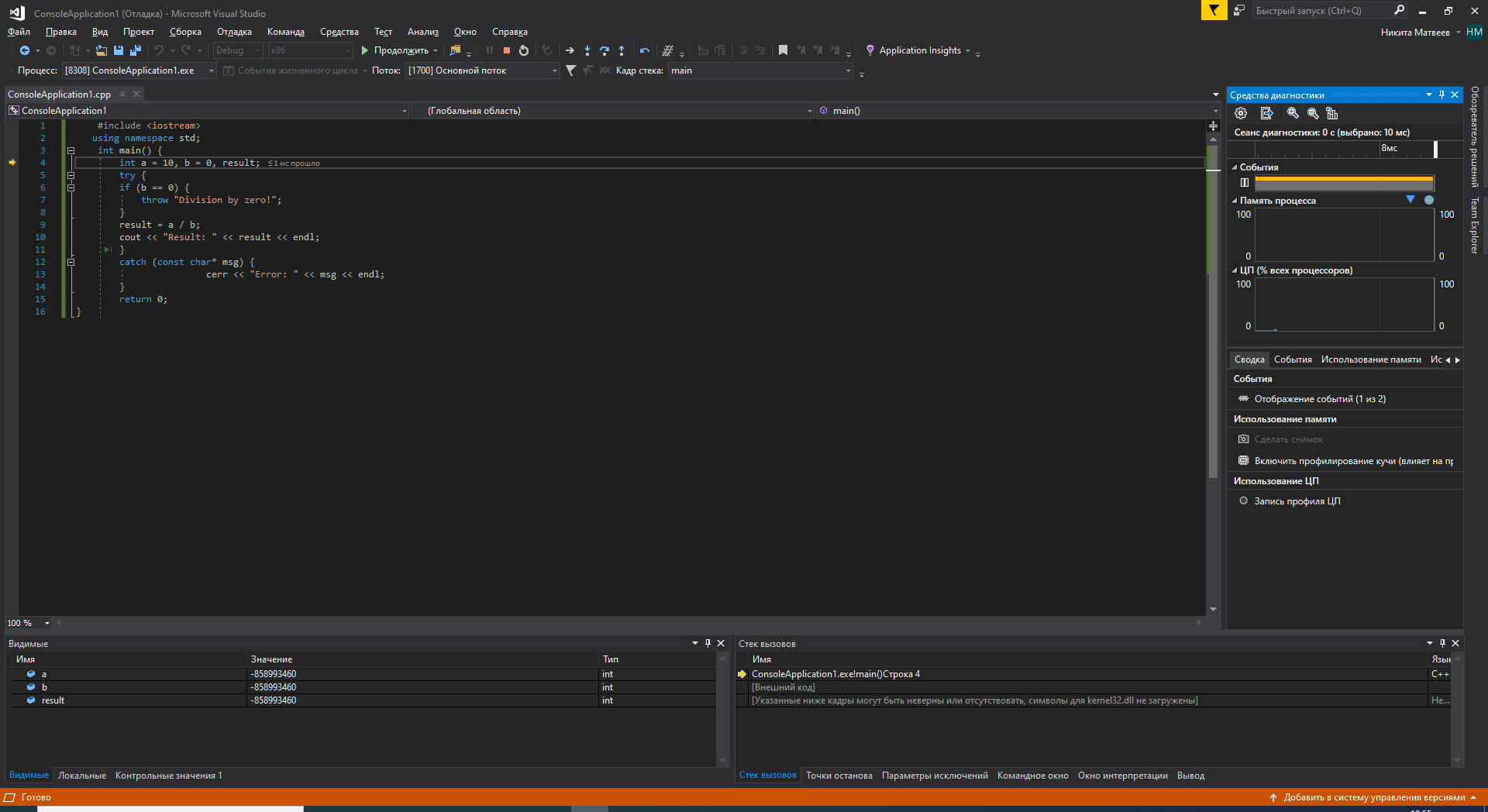
# **2.3. Использование средств профилирования**

1. Оценить использование памяти



1. Оценить использование ЦП

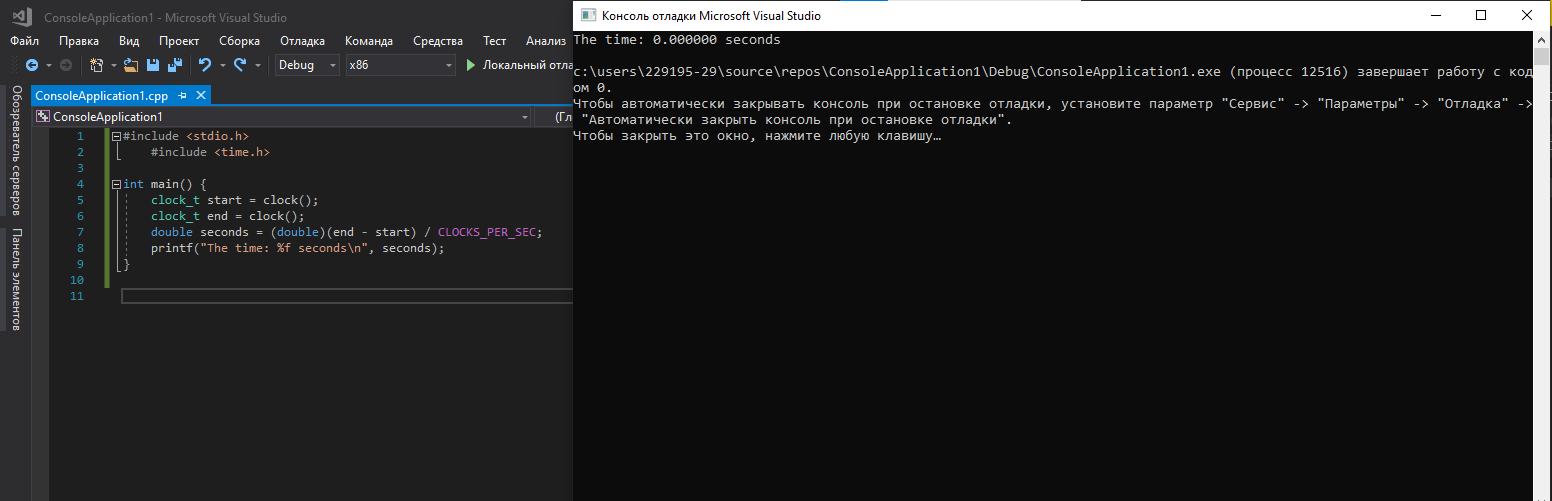




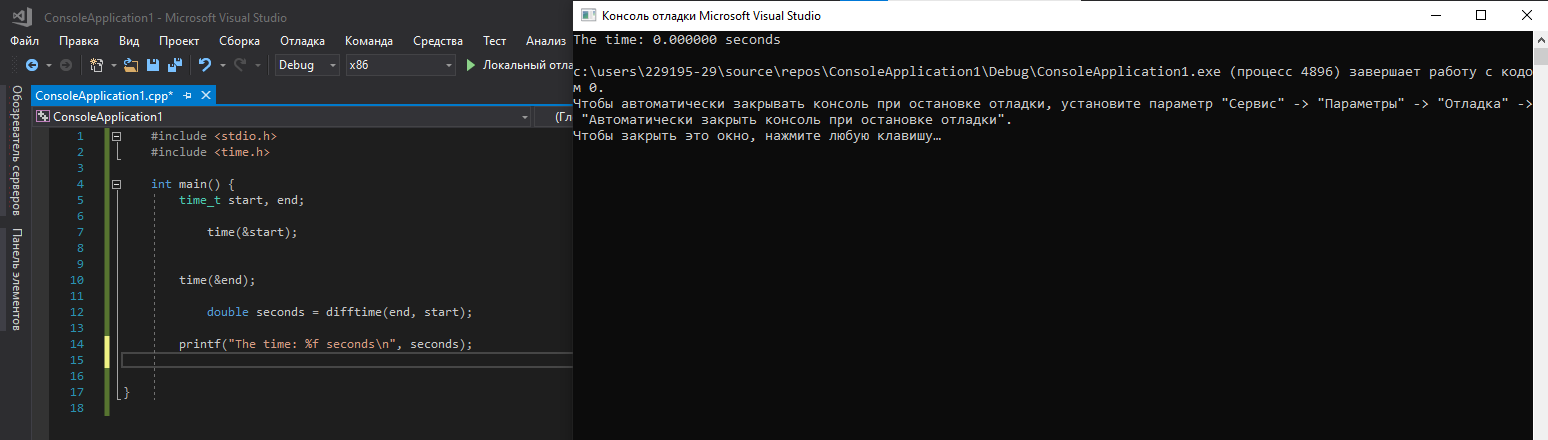
# **2.4. Определение времени работы функций программным способом**

Проанализируйте результаты работы функций clock и time. Сделайте выводы

1. Clock:



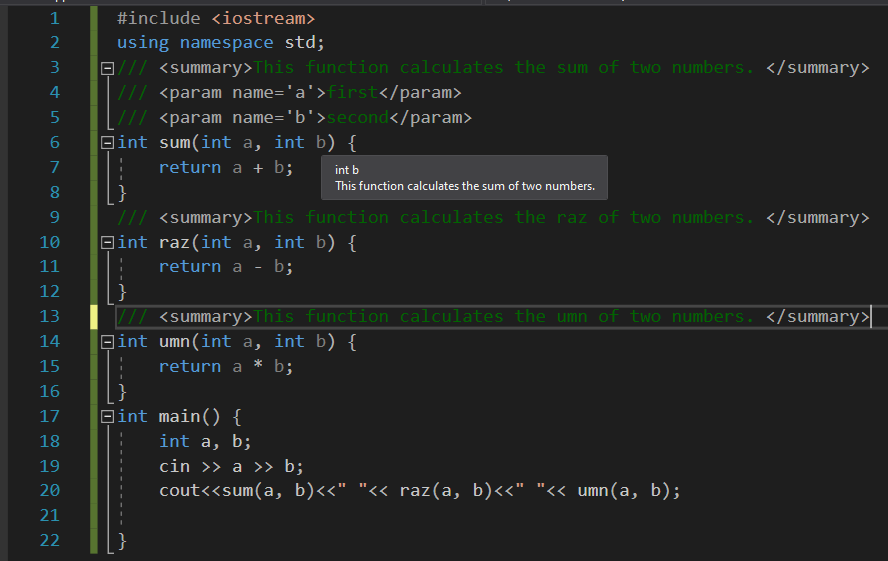
1. Time:



**3. Обеспечение качества кода**

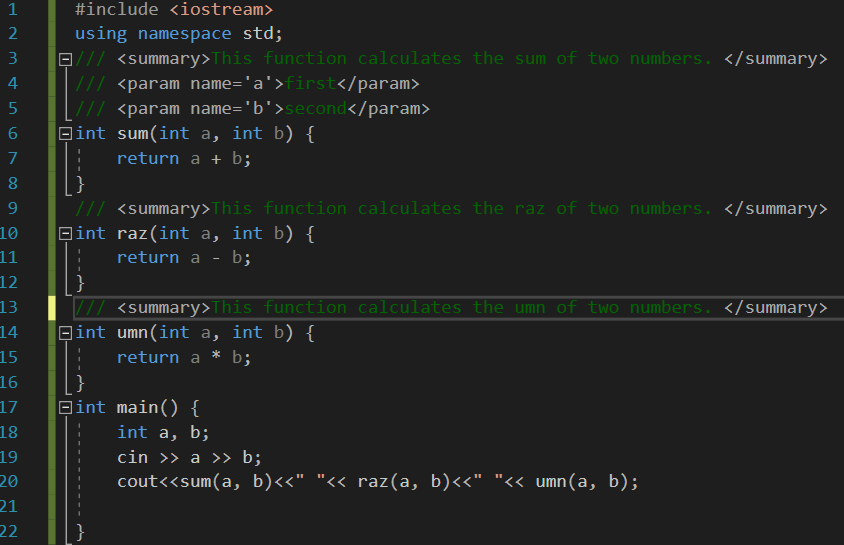
**3.1 Самодокументируемый код**

1. Создать проект
2. Добавить XML-комментарии для всех функций
3. Продемонстрировать всплывающую подсказку при вводе данной функции

****

# **3.2 Соглашение о кодировании**

* По возможности используйте современные языковые функции и версии C#.
* Избегайте устаревших или устаревших конструкций языка.
* Перехват исключений, которые можно правильно обрабатывать; избегайте перехвата универсальных исключений.
* Используйте определенные типы исключений для предоставления значимых сообщений об ошибках.
* Используйте запросы и методы LINQ для обработки коллекций, чтобы улучшить удобочитаемость кода.
* Используйте асинхронное программирование с асинхронным и ожиданием операций с привязкой ввода-вывода.
* Будьте осторожны с взаимоблокировками и используйте [Task.ConfigureAwait](https://learn.microsoft.com/ru-ru/dotnet/api/system.threading.tasks.task.configureawait) при необходимости.
* Используйте языковые ключевое слово для типов данных вместо типов среды выполнения. Например, используйте string вместо [System.String](https://learn.microsoft.com/ru-ru/dotnet/api/system.string) нее или int вместо [System.Int32](https://learn.microsoft.com/ru-ru/dotnet/api/system.int32).
* Используйте int вместо неподписанных типов. Использование int часто используется на C#, и при использовании int проще взаимодействовать с другими библиотеками. Исключения предназначены для документации, конкретной для типов данных без знака.
* Используйте var только в том случае, если средство чтения может вывести тип из выражения. Читатели просматривают наши примеры на платформе документов. У них нет подсказок по наведении указателя мыши или инструментов, отображающих тип переменных.
* Напишите код с четкостью и простотой.
* Избегайте чрезмерно сложной и запутанной логики кода.

****

# **3.3 Анализ качества кода**

4 if else- L if

0 операторов цикла – L lops

0 модулей и подсистем- L mod

Т.к. число операторов условия = 4, абсолютная сложность (CL) программы по формуле = 4

Относительная сложность программы равна: cl=CL/L=4/92=0,0435

(L)- общее число операторов

**4. Работа с реестром ОС Windows**

1. Создать программу Windows Forms
2. Используя системный реестр сохранить пользовательские изменения свойств программы (цвет фона, введенный текст, размер окна,)
3. Проверить сохранение данного свойства при перезапуске программы

