Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №1

на тему:

**«Простое компьютерное приложение»**

БГУИР 6-05-0612-02 113

|  |
| --- |
| Выполнил студент группы 453503  ХАЛАМОВ Николай Андреевич |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2025

**1 ЗАДАНИЕ**

1 Создать консольное приложение. Название проекта: \_НомерГруппы\_Фамилия.

2 Ознакомиться со структурой проекта.

3 Найти файл \*.scproj. Найдите в нем указание на целевую платформу и тип приложения.

4 Добавьте в проект NuGet пакет AutoMapper. Найдите ссылку на установленный пакет в файле \*.csproj.

5 Написать программу, которая выводит в консоль частное от деления чисел, введенных с клавиатуры. Введенные данные и результат сохранять в отдельных переменных.

6 Запустить проект в двух режимах: с отладкой и без отладки. Ввести сначала корректные, а затем заведомо неправильные значения чисел (строка вместо числа, ноль в качестве делителя). Сравнить результат в режиме отладки и выпуска.

7 Найти скомпилированные файлы (для режима отладки и режима выпуска)

8 Поставить точки останова. Запустить проект в двух режимах: с отладкой и без отладки. Сравнить результат. Прочитать в отладчике

**2 ВЫПОЛНЕНИЕ РАБОТЫ**

В начале выполнения работы было создано консольное приложение в Visual Studio, названое по примеру, данному в условии задания (см. рисунок 1).

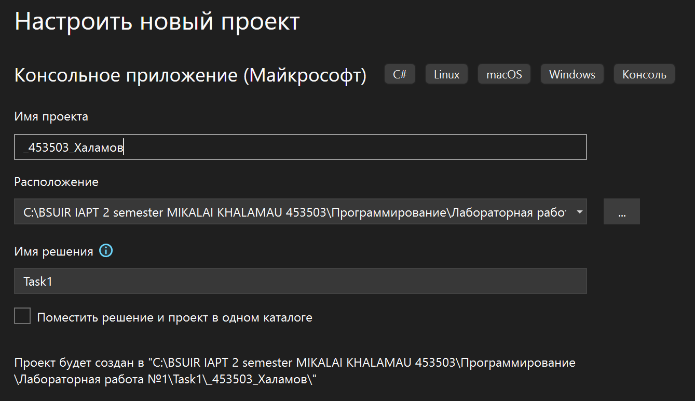


Рисунок 1 – Создание нового проекта (консольное приложение) с названием \_453503\_Халамов

После создания нового проекта в Visual Studio 2022, его структура отобразилась в обозревателе решений. Это файлы "Program.cs" (весь код программы на языке C# помещается в файлы с расширением .cs) и "\_453503\_Халамов.csproj" (открывается при двойном нажатии на название проекта и отвечает за общую конфигурацию проекта), которые являются основными файлами проекта (см.рисунок 2).

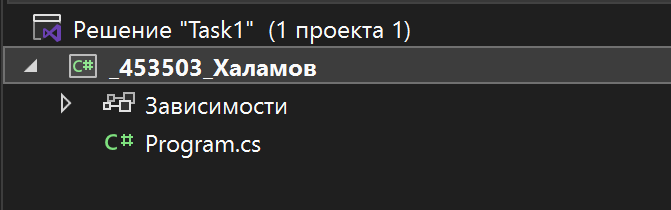


Рисунок 2 – Структура проекта \_453503\_Халамов

Для того, чтобы в файле \*.csproj найти указание на целевую платформу и тип приложения, нужно было открыть его двойным нажатием на название проекта. Внутри файла \*.csproj в начале была указана информация о целевой платформе и типе приложения, в элементе "PropertyGroup". В данном случае значение <OutputType>Exe</OutputType> указывает, что это приложение является исполняемым файлом (executable), а <TargetFramework>net8.0</TargetFramework> указывает на то, что проект предназначен для работы на платформе .NET 8.0 (см. рисунок 3).

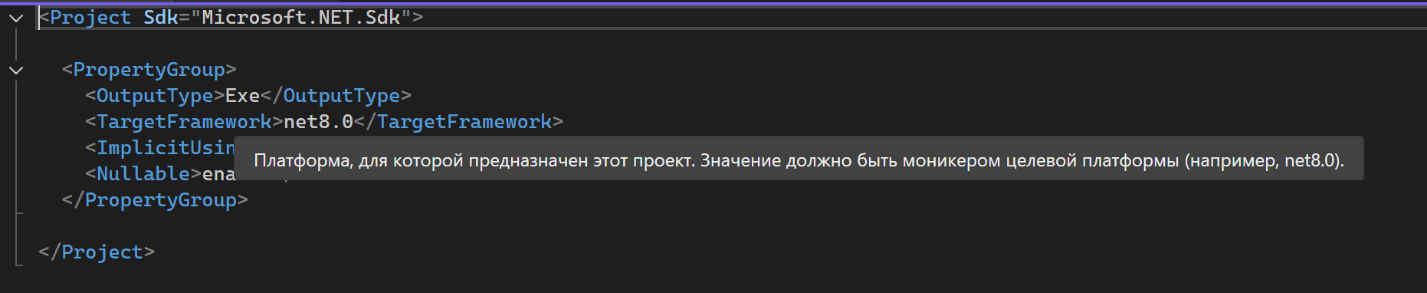


Рисунок 3 – Содержимое файла с расширением .csproj

Затем в проект NuGet был установлен пакет AutoMapper. Для того, чтобы подключить к проекту AutoMapper, надо было зайти в “Управление пакетами NuGet для решения” в обозревателе решений , “Обзор” в появившемся окне , в “Поиск” написать AutoMapper и установить его.

После установки его ссылка была автоматически добавлена в файл \*.csproj (см. рисунок 4).

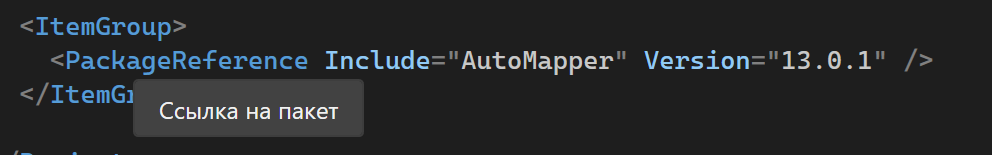


Рисунок 4 – Ссылка на установленный пакет AutoMapper в файле \*.csproj.

Программа, написанная на языке C#, представляет собой консольное приложение на C#, которое вычисляет частное от деления двух чисел, введённых пользователем. Был создан класс с импользованием ключевого слова internal, чтобы скрыть реализацию от внешнего кода и избежать неправильного использования. Был создан основной метод программы Main, с использованием public (модификатор доступа, который указывает, что метод доступен из других классов и сборок. Это необходимо, чтобы среда выполнения могла вызвать этот метод), static (указывает, что метод принадлежит классу, а не экземпляру класса. Это важно, поскольку метод Main вызывается до создания каких-либо объектов данного класса). Метод является void, поскольку не возвращает никакого значения. Для написания нужной программы были использованы базовые методы ввода/вывода класса Console и класс конвертирования Convert. Ввод данных осуществляется через метод Console.ReadLine(), а вывод — с помощью Console.WriteLine(). Код организован в пространстве имён \_453503\_Халамов, что является хорошей практикой для структурирования программ (см. рисунок 5).

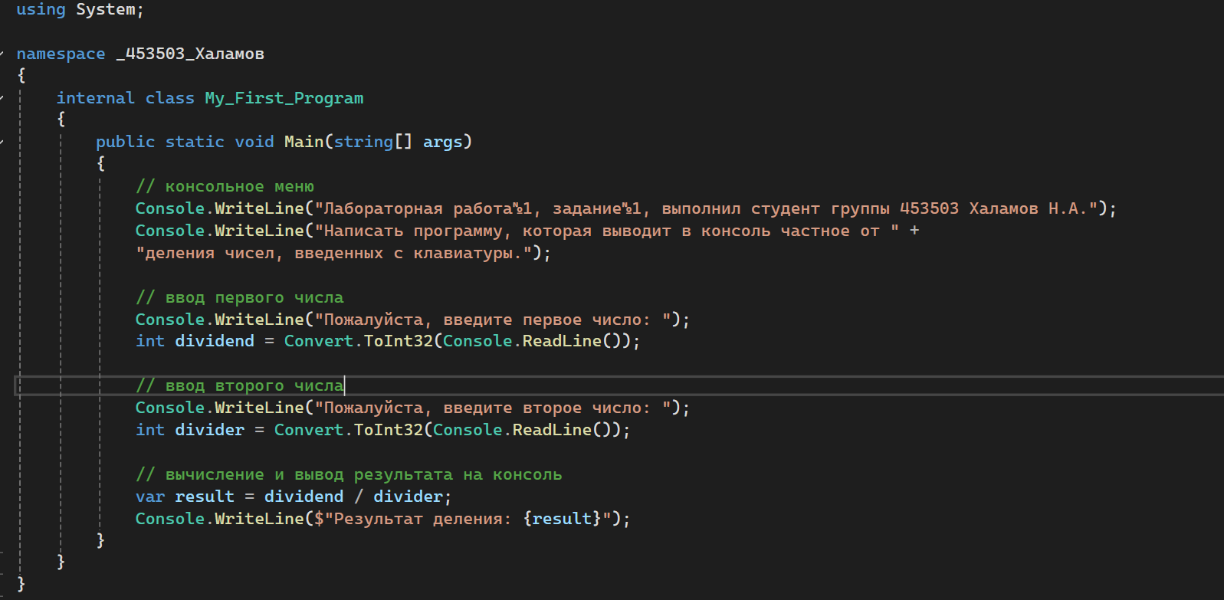


Рисунок 5 – Созданная программа

После запуска без отладки и ввода корректных входных данных программа вывела в консоль результат операции деления (см. рисунок 6).

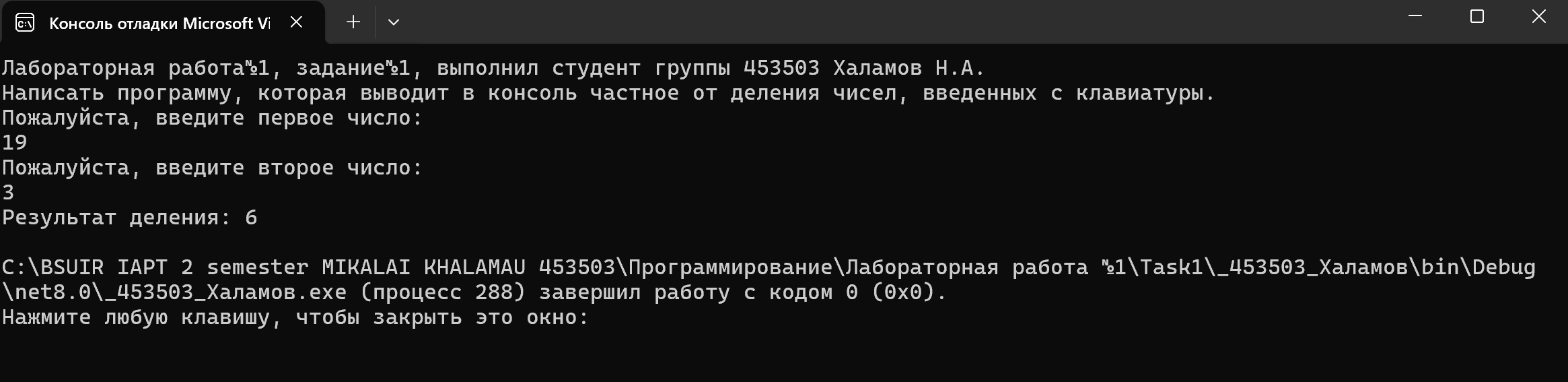


Рисунок 6 – Результат выполнения программы без отладки

После запуска с отладкой и ввода корректных входных данных программа идентично вывела корректные значения (см. рисунок 7).

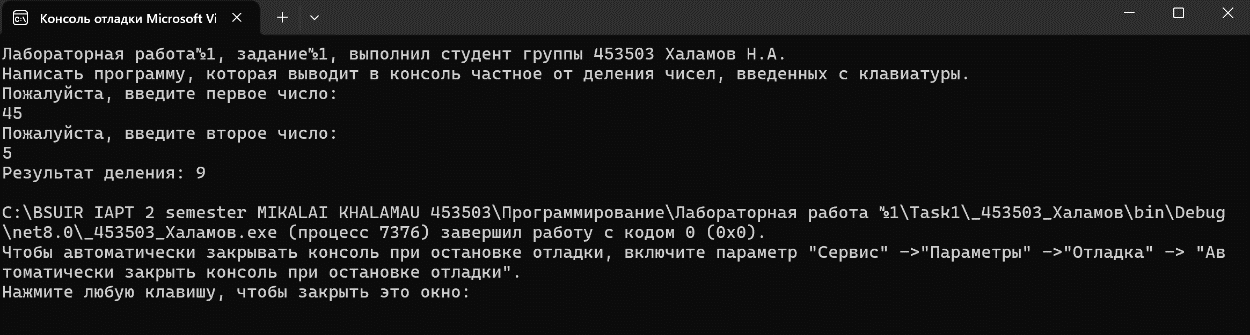


Рисунок 7 – Запуск проекта с отладкой, корректные данные

После запуска программы без отладки с некорректными входными данными (ноль в качестве делителя) программа закончилась с указанием ошибки «деление на 0» и ненулевым кодом, однако автоматического приостановления выполнения программы и открытия окна "Исключение" не произошло (см. рисунок 8).

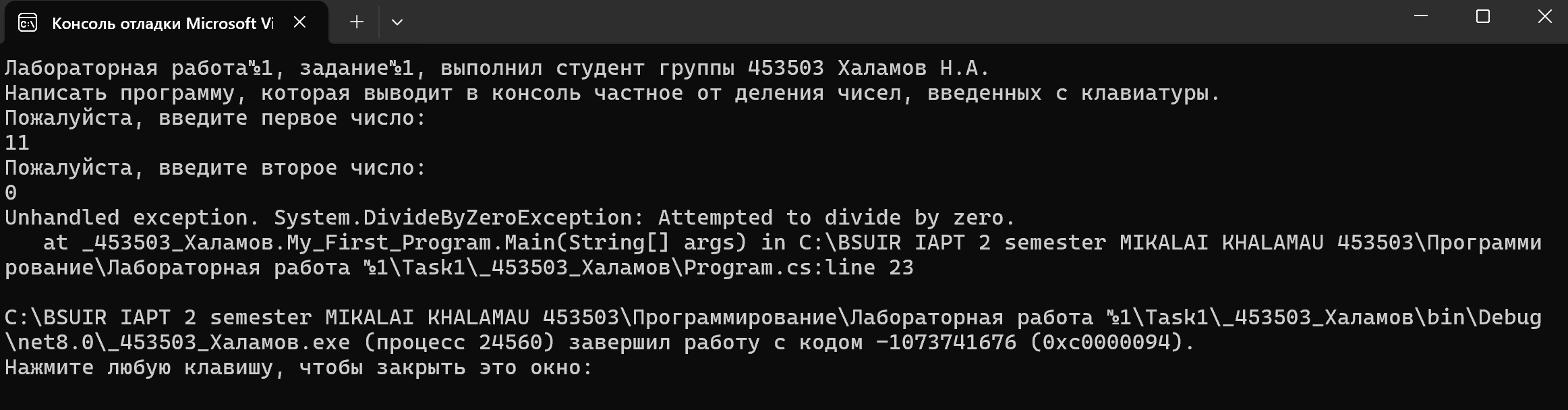


Рисунок 8 – Результат попытки выполнения программы

без отладки при введении нуля в качестве делителя

После запуска программы с отладкой и некорректными входными данными (строка вместа числа) программа остановилась и перешла к месту ошибки с некорректный форматом (см. рисунок 9).

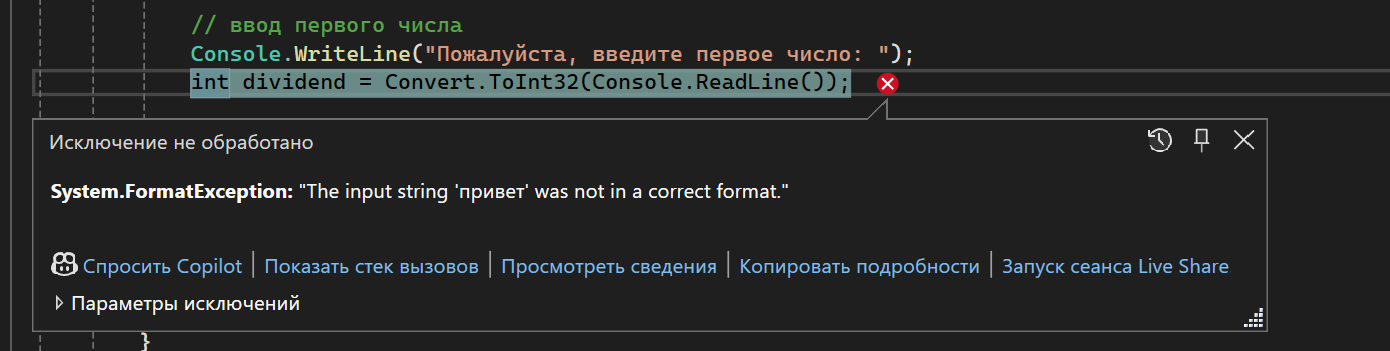


Рисунок 9 – Остановка программы в режиме отладки

при введении строки вместо числа

После запуска программы с отладкой и некорректными входными данными (ноль в качестве делителя) программа остановилась и перешла к месту ошибки с делением на ноль (см. рисунок 10).

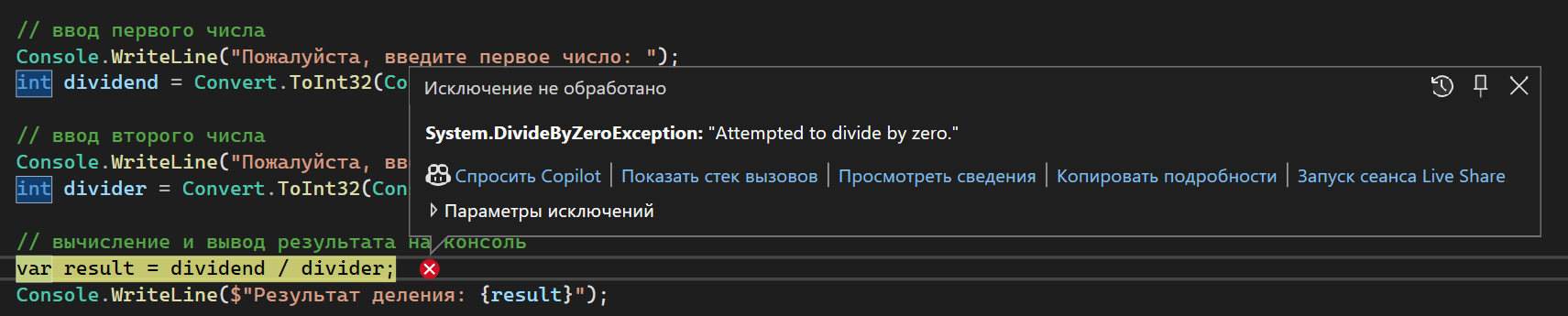


Рисунок 10 – Остановка программы в режиме отладки

при введении нуля в качестве делителя

Скомпилированные файлы для режима отладки находятся по адресу C:\...\\_453503\_Халамов\bin\Debug\net8.0\\_453503\_Халамов.exe, а для режима выпуска – по адресу C:\...\\_453503\_Халамов\bin\Release\net8.0\\_453503\_Халамов.exe (см. рисунок 11).

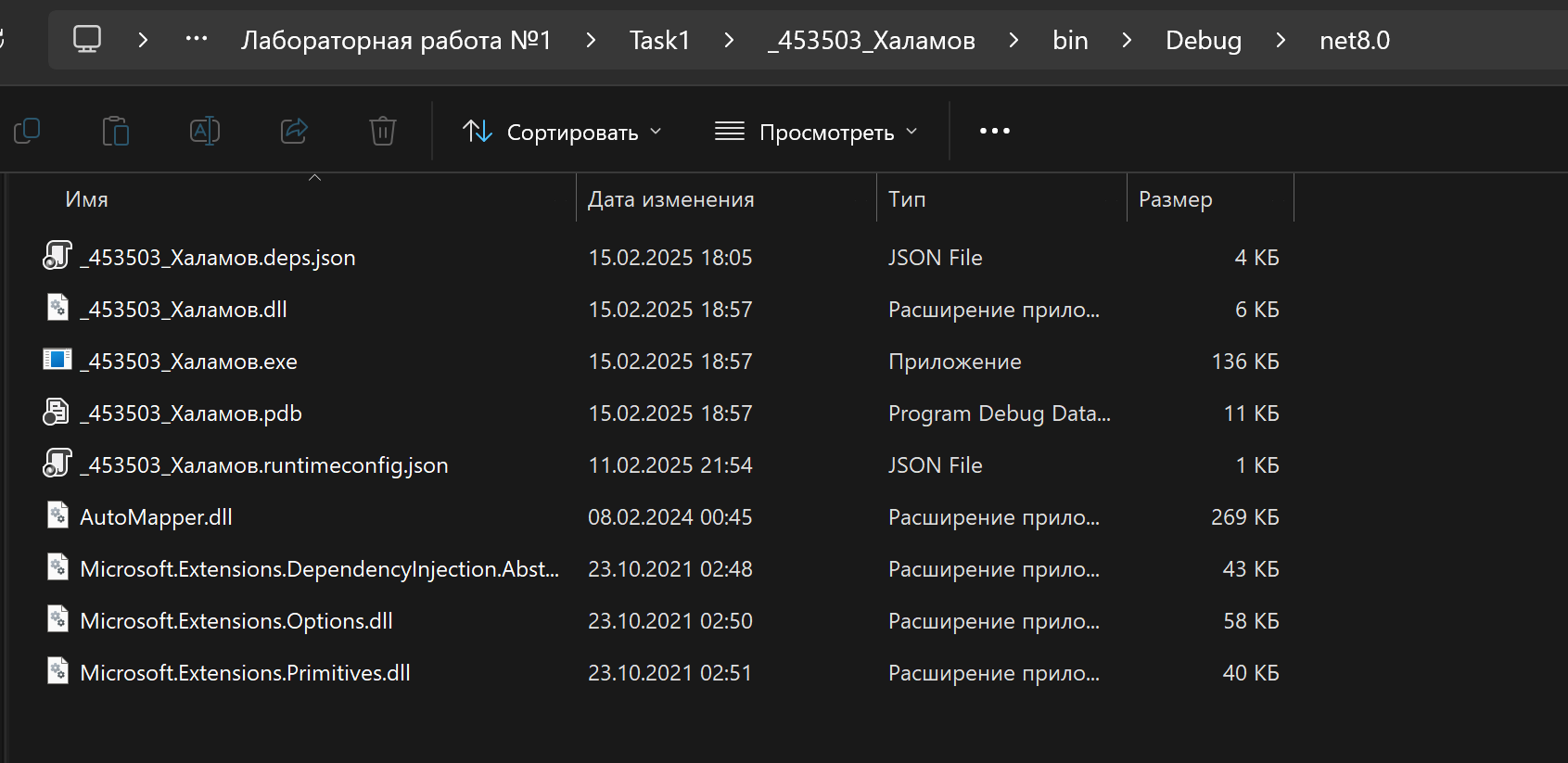


Рисунок 11 – Содержимое папки Debug

После установки точки останова и запуска программы без отладки она вывела результат и завершила работу (см. рисунок 12).

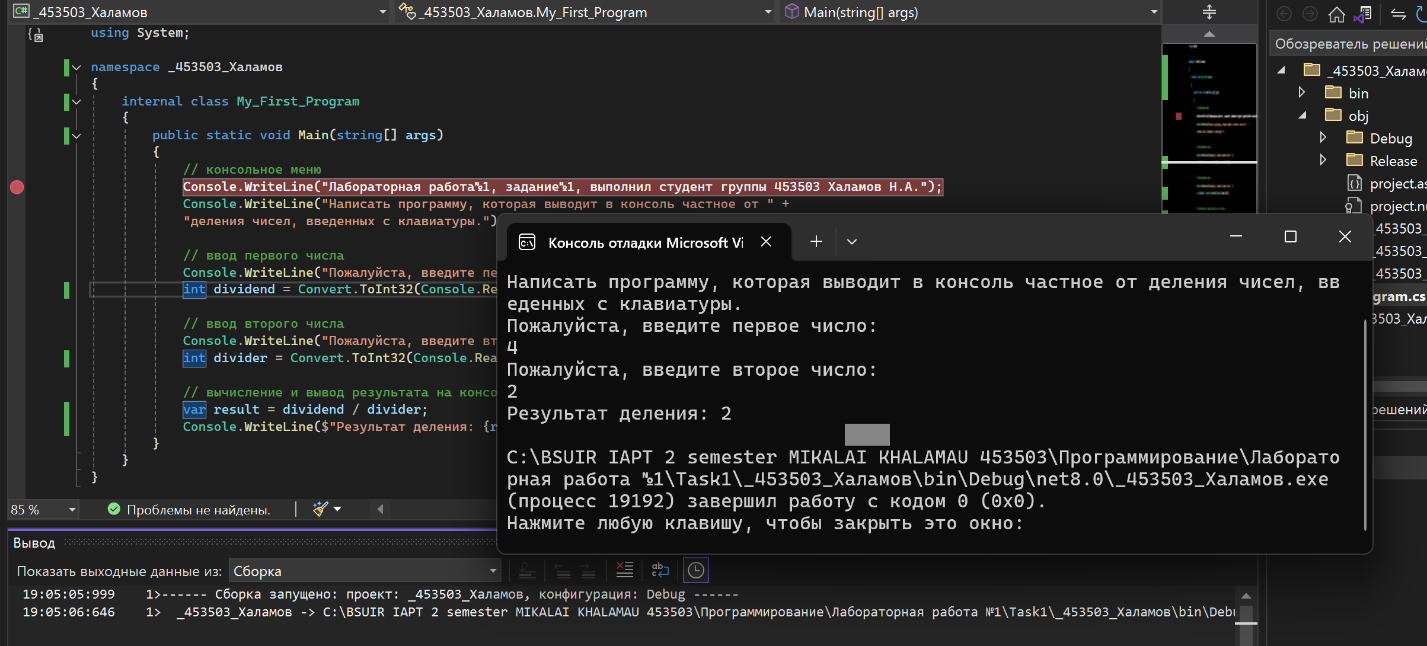


Рисунок 12 – Результат работы программы

с точкой останова без отладки

При запуске в режиме с отладкой с установленными точками останова на строках программа во время своего выполнения остановилась на каждой точке останова. Стало возможным прочитать значения переменных в окне отладчика, а также пошагово выполнить программу, следуя по одной строке кода за разом и наблюдая изменения переменных на каждом шаге (см. рисунок 13).

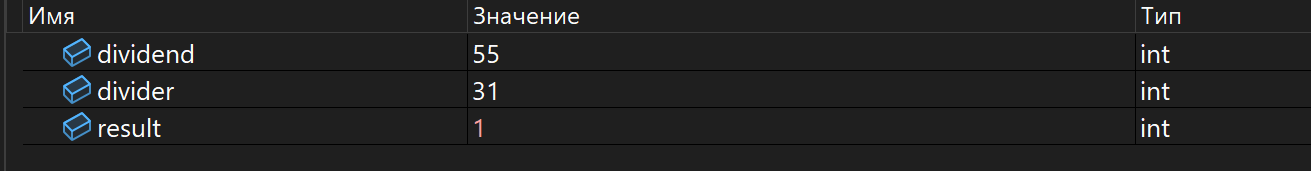


Рисунок 13 – Значения переменных в режиме запуска проекта с точкой останова

Для перехода к следующему шагу была использована комбинация клавиш «Fn + F10». При запуске без отладки программа выполнилась нормально, но прочитать значения переменных во время выполнения уже не было возможности.

**ВЫВОД**

В ходе выполнения лабораторной работы были изучены основы работы с проектами в среде разработки Visual Studio, включая поиск и анализ файлов проекта и установку NuGet пакетов, были изучены устройство консольного приложения С#, структура проекта и нахождение информации о нём (целевая платформа и тип приложения). Были разработаны и проверены основные навыки написания программ на языке C#, а именно нахождение частного от деления чисел. Также были изучены основные возможности отладки в Visual Studio, включая запуск проекта в режимах с отладкой и без отладки, установку точек останова и пошаговое выполнение.