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 КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ»**

БГУИР 6-05-0612-02 124

Выполнил студент группы 453504

Ярцев Александр Александрович

(дата, подпись студента)

Проверил ассистент каф. Информатики

Романюк Максим Валерьевич

(дата, подпись преподавателя)

Минск 2025

# ЗАДАНИЕ

**Задание 1.** Создать консольное приложение. Название проекта:

\_НомерГруппы\_Фамилия.

**Задание 2.** Ознакомиться со структурой проекта.

**Задание 3.** Найти файл \*.csproj. Найдите в нем указание на целевую

платформу и тип приложения.

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

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

сохранять в отдельных переменных.

**Задание 6.** Запустить проект в двух режимах: с отладкой и без отладки.

Ввести сначала корректные, а затем заведомо неправильные значение чисел (строка вместо числа, ноль в качестве делителя). Сравнить результат в режиме отладки и выпуска.

**Задание 7.** Найти скомпилированные файла (для режима отладки и режима выпуска).

**Задание 8.** Поставить точки останова. Запустить проект в двух режимах: с отладкой и без отладки. Сравнить результат. Прочитать в отладчике значения переменных. Провести пошаговое выполнение программы**.**

# ВЫПОЛНЕНИЕ РАБОТЫ

Для первого задания создано консольное приложение с названием проекта: \_453504\_Ярцев. Для этого использовалась среда разработки Visual Studio. В процессе создания проекта был выполнен шаблон для консольного приложения на C#.

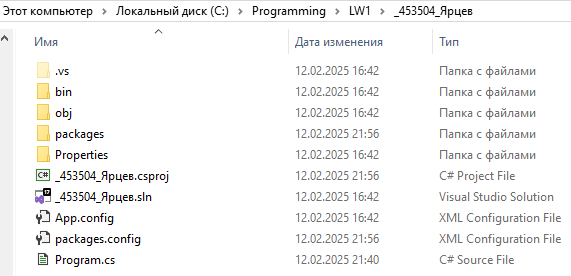


Рисунок 1 – Название проекта (на предыдущей странице)

Для второго задания было знакомство со структурой папок и файлов:

bin/Debug/: данная папка содержит скомпилированные файлы приложения для разных режимов.

obj/Debug/: папка с временными файлами сборки.

Program.cs: основной файл проекта, содержащий код консольного приложения.

.csproj: файл проекта, описывающий его настройки и зависимости.

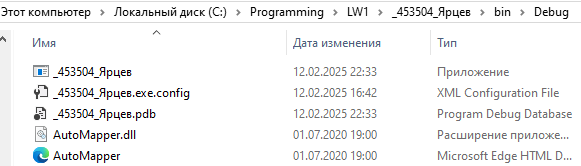


Рисунок 2 – Содержание папки bin\Debug\

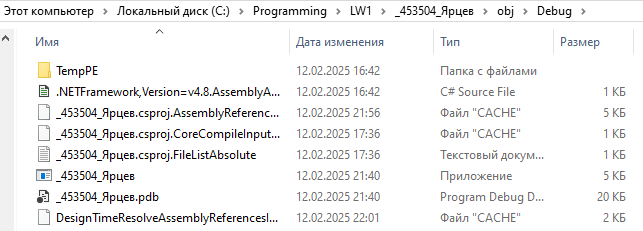


Рисунок 3 – Содержание папки obj\Debug\

Для третьего задания посмотрим проект. В проекте находится файл с расширением .csproj, который содержит указание на целевую платформу и тип приложения. Содержание элемента App.config указано ниже:

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />

</startup>

</configuration>

Целевая платформа: .NET Framework 4.8 (указано в элементе App.config). Тип приложения:Exe. (Расположение: C:\Programming\LW1\\_453504\_Ярцев\obj\Debug).

Для четвертого задания в проект был добавлен NuGet пакет AutoMapper. При изменении конфигурация файла \_453504\_Ярцев.csproj имеется возможность управлять пакетами NuGet и там же установить пакет AutoMapper, после установки появится элемент packages.config в файле \_453504\_Ярцев.csproj:

<?xml version="1.0" encoding="utf-8"?>

<packages>

<package id="AutoMapper" version="10.0.0" targetFramework="net48" />

</packages>

Для пятого задания в файл Program.cs добавим программу, которая запрашивает ввод двух чисел, выполняет их деление в функции Devide (при этом имеется проверка на деление на ноль) и выводит результат в консоль. Код программы:

using System;

namespace \_453504\_Ярцев

{

class Program

{

static void Main()

{

int a, b, c;

Console.WriteLine("Введите делимое: ");

a = int.Parse(Console.ReadLine() ?? "0");

Console.WriteLine("Введите делитель: ");

b = int.Parse(Console.ReadLine() ?? "0");

// Проверка на деление на ноль

if (b == 0)

{

Console.WriteLine("Деление на ноль недопустимо.");

return; // Завершение программы в случае, при условии b == 0

}

c = Divide(a, b);

Console.WriteLine("Частное: ");

Console.WriteLine(c);

}

public static int Divide(int a, int b)

{

return a / b;

}

}

}

Для шестого задания запустим проект в двух режимах: с отладкой и без отладки.

Корректный ввод и некорректный ввод (рис.4,5)

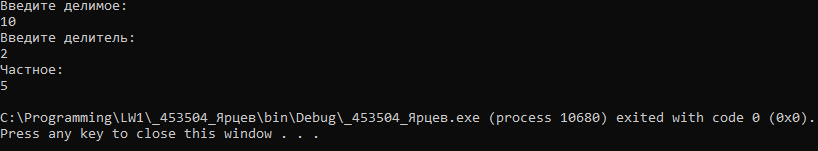


Рисунок 4 – Запуск файла Program.cs без отладки с корректным вводом. Аналогично будет и для режима с отладкой.

Программа выдала правильный результат при введении чисел, например, 10 и 2 (частное: 5).

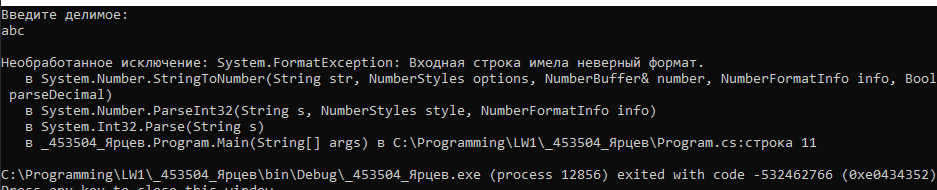


Рисунок 5 – Запуск файла Program.cs без отладки с некорректным вводом

(ввод строки).

При введение первого значение – строки (abc) программа сразу выдала предупреждение и не дала ввести второе значение. В случае запуска файла с отладкой в явном виде в приложение Visual Studio укажет на строку с исключением.

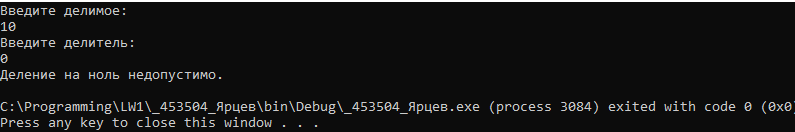


Рисунок 6 – Запуск файла Program.cs без отладки с некорректным  
вводом(ввод нуля в качестве делителя).

В режиме с отладкой выведет такой же текст, так как это исключение было обработано в исходном коде.

Для седьмого задания скомпилированные файлы находятся в папке Debug (Путь к папке смотреть на рис.7).

Для режима отладки и режима выпуска там будут следующие файлы:

Exe файл: \_453504\_Ярцев.exe (в режиме выпуска)

Pdb файл: \_453504\_Ярцев.pdb (в режиме отладки)

Dll файл: AutoMapper.dll и другие файлы

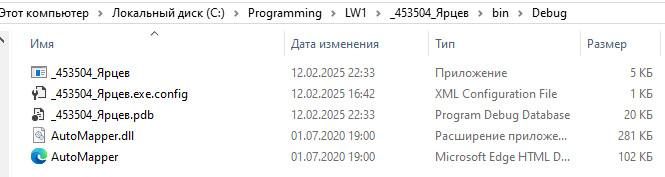


Рисунок 7 – Содержание файла Debug

Для восьмого задания были установлены точки останова в коде программы. Запущены оба режима:

Режим отладки: программа останавливалась на точках останова, и значения переменных могли быть проанализированы.

Режим выпуска: программа выполнялась без остановок.

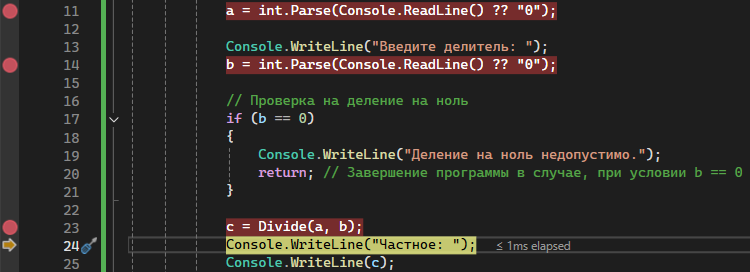


Рисунок 8 – Установка точек останова (Красные точки слева).



Рисунок 9 – Значение переменных в режиме отладки во время выполнения программы.

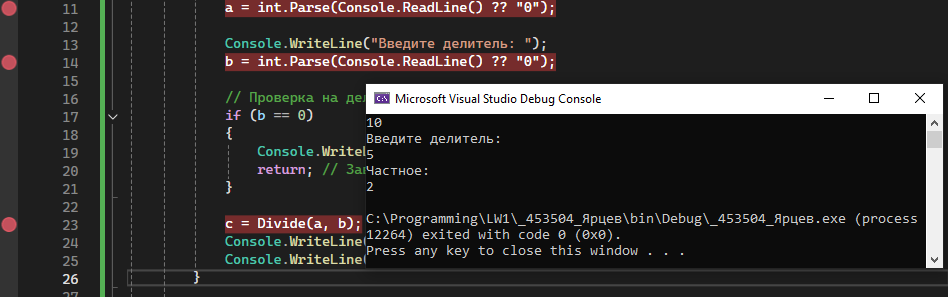


Рисунок 10 – Запуск программы с точками останова в режиме без отладки.

**ВЫВОД**

В ходе лабораторной работы были изучены основы создания и настройки консольного приложения на языке программирования C# в среде разработки Visual Studio, работа с NuGet пакетами(в частности с пакетом AutoMapper), а также практические навыки отладки и анализа кода. Также был изучен базовый синтаксис языка программирования С# для решения задания под номером пять (смотреть выше).