Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №1

на тему:

**«ПРОСТОЕ КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ»**

БГУИР 6-05-0612-02 05

|  |
| --- |
| Выполнила студент группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 задание

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

2 Ознакомиться со структурой проекта.

3 Найти файл \*.csproj. 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 значения переменных. Провести пошаговое выполнение программы.

# Выполнение работы

Для выполнения этой и последующих лабораторных работ необходимо установить специальную IDE — Rider. Для создания приложения необходимо выбрать пункт “New Solution” (см. рис. 1).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1 – Создание проекта в Rider

Затем следует обратить внимание на выбор необходимых параметров приложения (см. рис. 2): “.NET Framework”, “Console Application”.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2 – Выбор необходимых параметров приложения в Rider

Также необходимо указать имя проекта в виде \_НомерГруппы\_Фамилия и выбрать пункт “Create” (см. рис. 3).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 3 – Название проекта в Rider

Структура проекта на языке C# в Rider обычно состоит из нескольких основных элементов:

* Solution (Решение) — это корневой элемент, который содержит один или несколько проектов. Solution представляет собой файл с расширением .sln и предоставляет общий контейнер для всех проектов в вашем решении.
* Project (Проект) — это отдельный файл с расширением .csproj, который содержит всю информацию о вашем проекте, такую как настройки компилятора, ссылки на внешние библиотеки и файлы, настройки сборки.
* Source Files (Исходные файлы) — это файлы с кодом на языке C#, которые содержат логику вашего приложения. Обычно имеют расширение .cs.
* References (Ссылки) — здесь вы можете указать внешние библиотеки и пакеты, которые ваш проект использует.
* Dependencies (Зависимости) — это компонент, который отображает зависимости между проектами в вашем решении.
* Solution Explorer (Обозреватель решений) — это панель, которая позволяет вам просматривать и управлять файлами и проектами в вашем решении.

Структура проекта представлена на рисунке 4.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 4 – Структура проекта

Открыв в папке проекта файл вида \_НомерГруппы\_Фамилия.csproj (см. рис. 5), можно узнать целевую платформу и тип приложения.

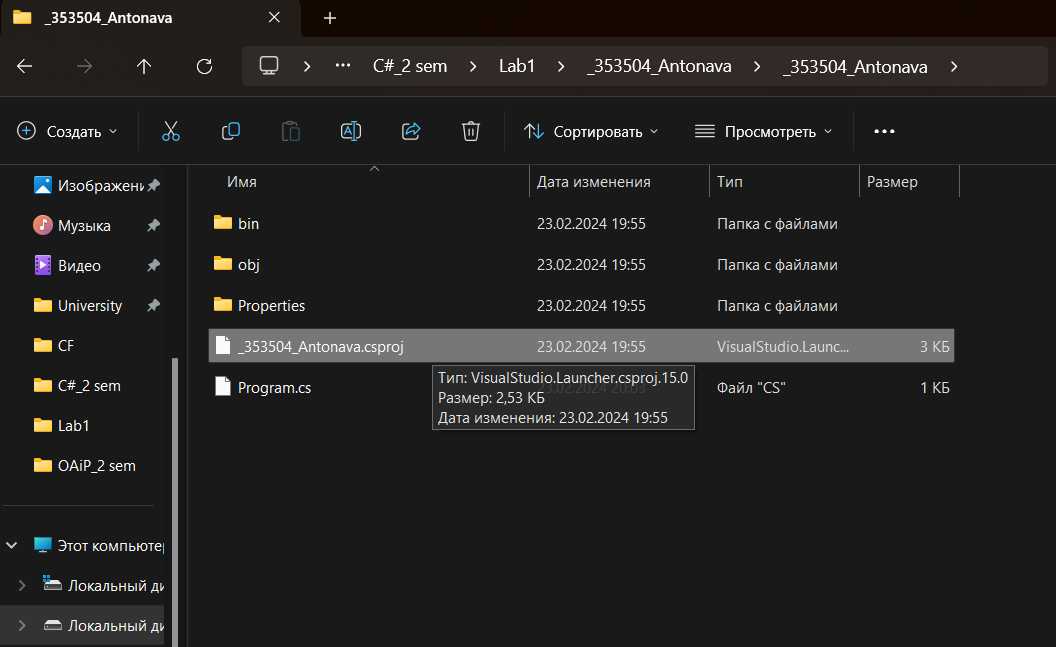


Рисунок 5 – Файл проекта .csproj

Открыв данный файл, можно увидеть следующее (см. рис. 6)

Tег файла <OutputType> содержит указание на тип приложения exe. Тег <PlatformTarget> содержит указание на целевую платформу AnyCPU (выделено на рисунке 6).

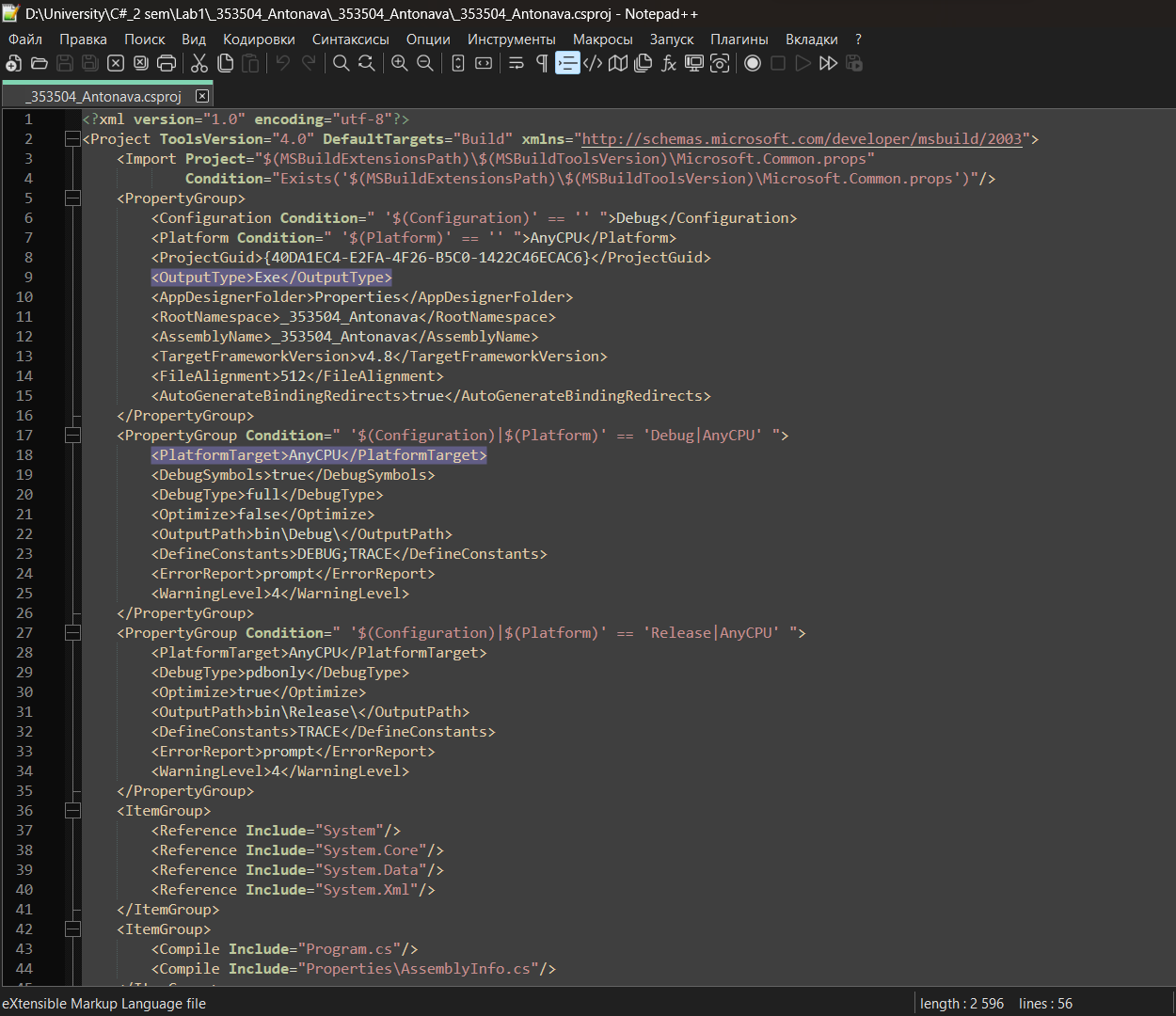


Рисунок 6 – Содержимое файла .csproj

Для того, чтобы добавить в проект пакет AutuMapper необходимо открыть файл \*.csproj в Rider, а затем выбрать необходимый пакет из перечня (см. рис. 7).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 7 – Добавление пакета AutoMapper

После добавления пакета, открыв файл с расширением .csproj, внутри секции <ItemGroup> можно увидеть строку с тегом <Reference Include>. Строка содержит название установленного пакета NuGet и версию пакета (см. рис. 8).

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 8 – Информация об установленном пакете

Простейшая программа, которая выводит в консоль частное от деления чисел, введенных с клавиатуры. Ниже представлен листинг кода.

using System;

namespace \_353504\_Antonava {

internal class Program {

public static void Main (string[] args) {

Console.WriteLine("Enter a: ");

double a = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter b: ");

double b = Convert.ToDouble(Console.ReadLine());

double x = a / b;

double y = b / a;

Console.WriteLine("a/b = " + x + "\nb/a = " + y);

}

}

}

В данном коде используются методы Console.Write() и Console.WriteLine(), которые предназначены для вывода текста на консоль. У них есть небольшое отличие: Console.WriteLine() в отличие от Console.Write() после вывода текста переносит курсор на следующую строку. Для использования методов для ввода-вывода необходимо подключить пространство имен System с помощью ключевого слова using. Также для чтения введенной пользователем информации используется метод Console.ReadLine(). Метод Convert.ToDouble() предназначен для преобразования значений в тип с плавающей точкой (double).

Главный выполняемый код находится в функции Main, которая является точкой входа в программу. Функция располагается в классе class Program, который имеет название файла с расширением .cs. А сам класс находится в пространстве имен с названием данного проекта (namespace \_353504\_Antonava).

В коде программы введенные данные и результат сохраняются в отдельных переменных: 2 числа расположены в переменных a и b, а также их частные – в переменных x и y.

Для запуска программы с отладкой необходимо использовать сочетание клавиш Alt + F5.

Для запуска программы без отладки используется сочетание клавиш Ctrl + F5.

Вывод в консоли при корректных входных данных показан на рис. 9.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Рисунок 9 – Результат выполнения программы при корректных входных данных

Результат ввода строки вместо числа в режиме отладки представлен на рисунке 10. Пользователя информируют о неверном формате входной строки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 10 – Результат выполнения программы с отладкой при вводе строки вместо числа

Результат выполнения программы без представлен на рисунке 11.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Результат выполнения программы без отладки при вводе строки

Результат выполнения программы с отладкой при вводе числа 0 (нуль) представлен на рисунке 12.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 12 – Результат выполнения программы с отладкой при вводе нуля

Результат выполнения программы без отладки при вводе числа 0 (нуль) представлен на рисунке 13.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок 13 – Результат выполнения программы без отладки при вводе 0

Скомпилированные файлы для режима отладки (см. рис. 14) и для режима выпуска располагаются по следующему пути: \_НомерГруппы\_Фамилия –> bin –> Debug\Release (для выпуска).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 14 – Скомпилированные файлы для режима отладки

После установки точек останова требуется запустить отладку как уже было показано ранее. Проход каждой строки осуществляется с помощью опции “Шаг с обходом” или клавиши F10.

Результаты выполнения программы в режимах с отладкой и без отладки идентичны и показаны соответственно на рисунке 15 и на рисунке 16.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 15 – Результат выполнения программы в режиме с отладкой

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Рисунок 16 – Результат выполнения программы в режиме без отладки

# Вывод

В ходе выполнения Лабораторной работы №1 были приобретены навыки создания простого консольного приложения в Rider, изучена структура проекта, добавление NuGet пакетов, написания программы, запуска и отладки проекта в разных режимах. Также были получены навыки использования точек останова для отладки программы.