**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа энергетики

Отделение электроэнергетики и электротехники

Направление: 09.04.03 Прикладная информатика

Дисциплина: Программирование и основы алгоритмизации

Отчет

по лабораторной работе №1

**Знакомство со средой Visual Studio 2022**

Вариант 12

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. О-5КМ41 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Рустамов Р.А. |
| Проверил: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) | Гончаров А.С. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (балл) |  |

Томск-2025

# ХОД РАБОТЫ

На рисунке 1 показан запуск интегрированной среды разработки Visual Studio 2022. После открытия программы пользователю предоставляется стартовое окно, где можно создать новый проект, открыть уже существующий или подключиться к системе контроля версий.

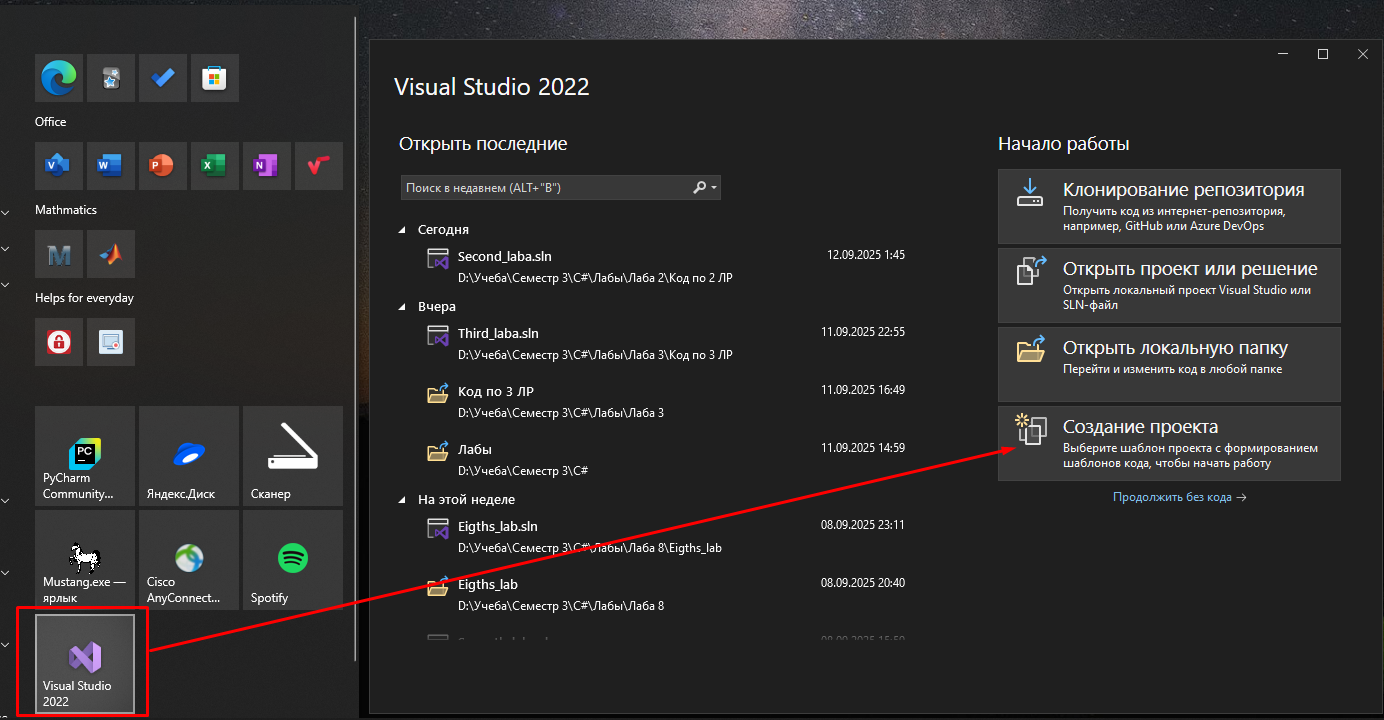


Рисунок –Запуск Visual Studio 2022

На рисунке 2 отображается выбор шаблона проекта. Для выполнения лабораторной работы был выбран тип «Консольное приложение (NET Framework)», предназначенный для создания консольных программ на языке C#.

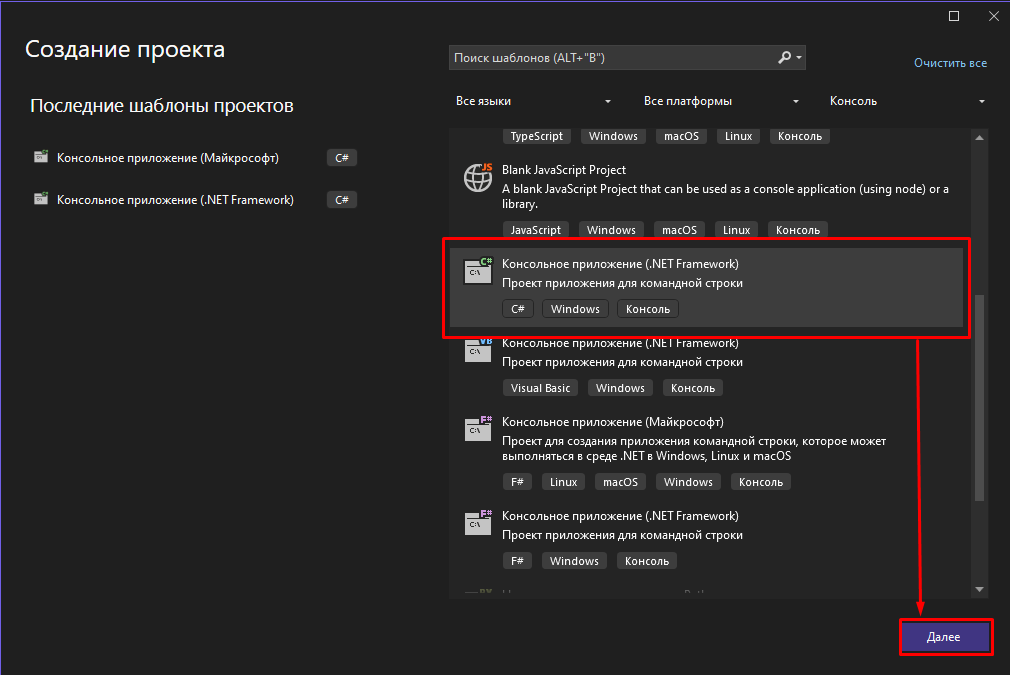


Рисунок –Окно создания проекта

На рисунке 3 представлено окно настройки нового проекта. Здесь задаются имя проекта (FirstLab), его расположение на диске и используемая платформа (*.*NET Framework 4.8). Также можно объединять несколько проектов в одно решение или хранить их отдельно.

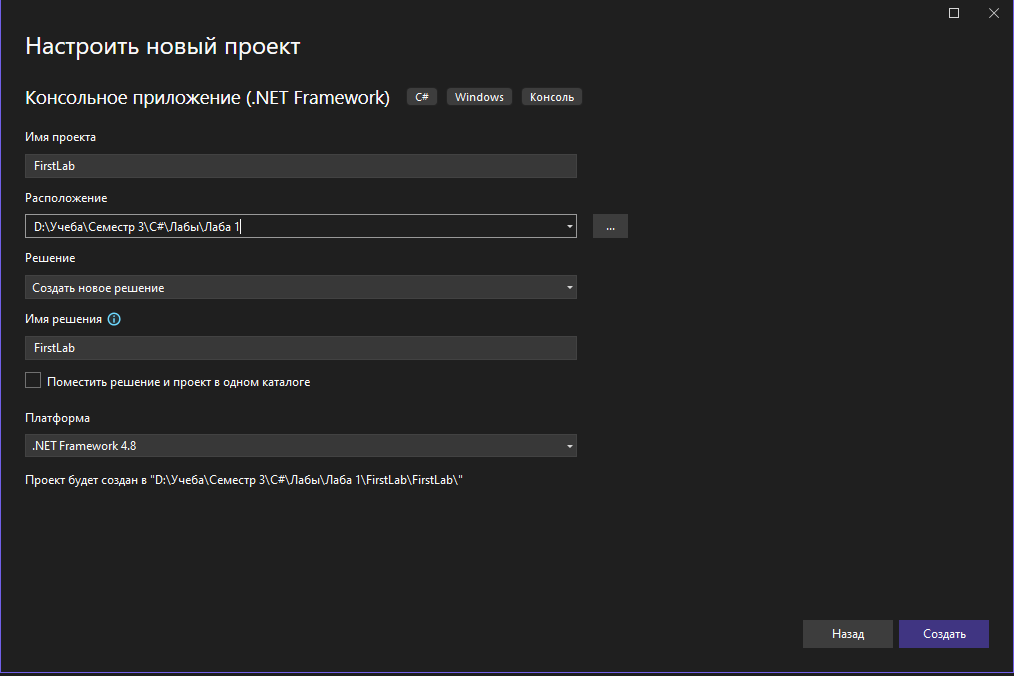


Рисунок –Настройка проекта

На рисунке 4 показан интерфейс Visual Studio после создания консольного приложения. В центральной области открыт файл Program.cs с исходным кодом на языке C#. В правой части окна отображается структура проекта в Обозревателе решений, где содержатся все файлы и настройки. В верхней панели расположена кнопка запуска программы, с помощью которой осуществляется компиляция и выполнение кода. В нижней части отображается окно вывода и ошибок, где выводятся сообщения системы. Данный интерфейс позволяет удобно разрабатывать, отлаживать и запускать программы на C#. »

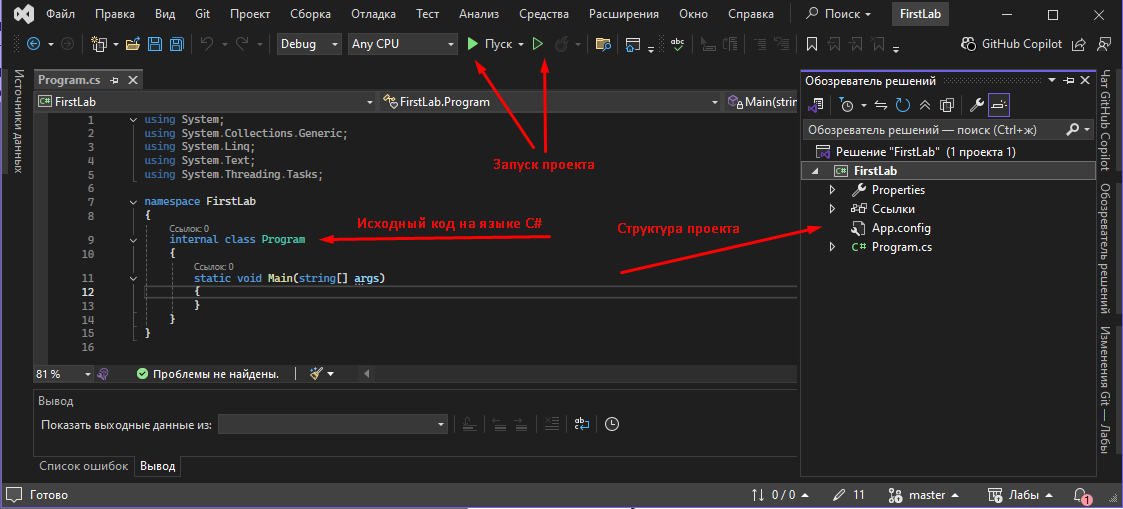


Рисунок – Рабочая среда

Эти строки подключают пространства имён (библиотеки), чтобы можно было использовать встроенные классы и функции без полного пути:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

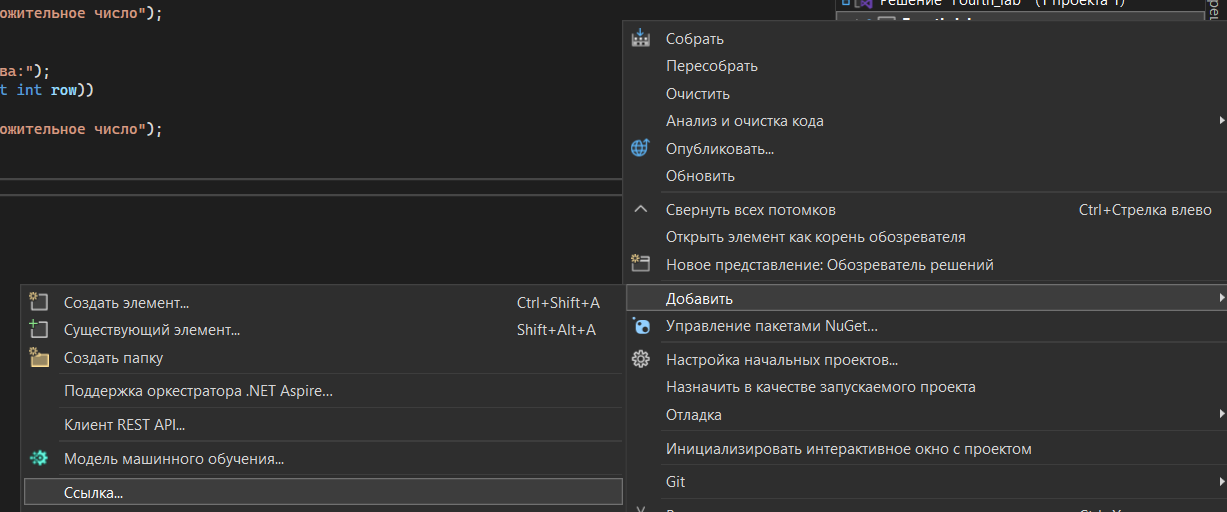


Рисунок - Добавление ссылок на библиотеки

Далее объявляется пространство имен. Это группировка кода. Всё, что написано внутри фигурных скобок, относится к namespace FirstLab**.** Это удобно для организации проектов и чтобы имена классов не конфликтовали между собой.

namespace FirstLab

{

\\ что-то тут есть?!

}

Здесь объявлен класс Program, который будет содержать логику программы. internal значит, что этот класс виден только внутри текущей сборки (проекта).

internal class Program

{

\\ что же тут?

}

Это главный метод, с которого начинается выполнение программы.

* static — метод принадлежит классу Program, а не конкретному объекту.
* void — метод ничего не возвращает.
* string[] args — массив строк, куда попадают аргументы командной строки (если программу запускать с параметрами).

static void Main(string[] args)

{

// пусто…

}

Сейчас тело метода пустое, поэтому программа ничего не делает — она просто запускается и сразу завершает работу.

Далее была написана команда, которая выводит текст «Hello, World!» в консоль.

Console.WriteLine("Hello, World!");

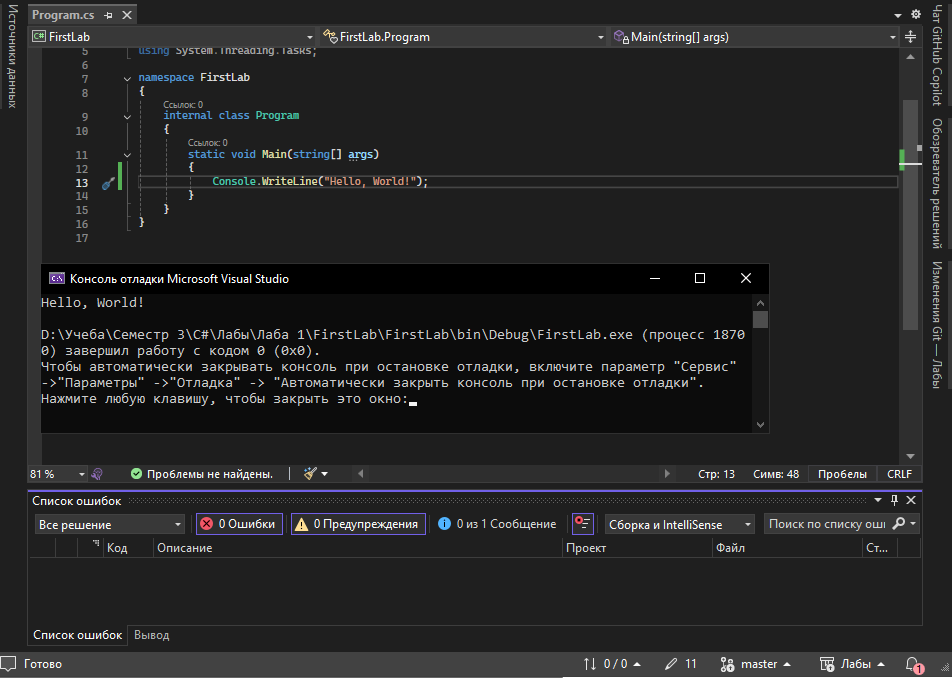


Рисунок – Пример компиляции кода

В окне консоли также указано, где находится скомпилированный файл (…\FirstLab\bin\Debug\FirstLab.exe). Приведены подсказки о том, как автоматически закрывать или оставлять консоль открытой после завершения программы. Внизу видно, что в проекте нет ошибок и предупреждений («0 ошибок, 0 предупреждений»), компиляция прошла успешно.

«На рисунке представлена модифицированная программа, в которую добавлена команда:

Console.ReadLine();.

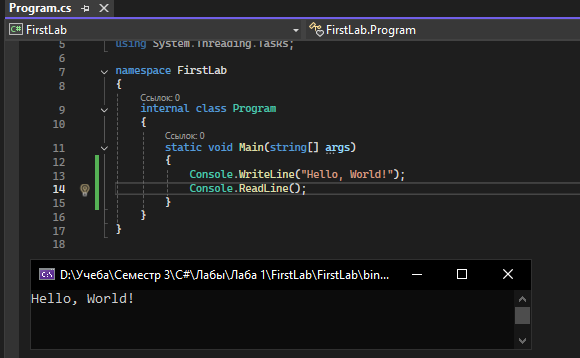


Рисунок – Результат вывода в консоль после добавления команды Console.ReadLine()

Она используется для задержки выполнения и позволяет оставить окно консоли открытым до тех пор, пока пользователь не введёт данные. Таким образом, результат работы программы (Hello, World!) остаётся на экране

# Заключение

В ходе выполнения лабораторной работы № 1 были изучены:

* Запуск среды разработки Visual Studio 2022 и создание консольного проекта.
* Пользовательский интерфейс среды разработки Visual Studio 2022.
* Текстовый редактор и структура программного кода.
* Процесс сборки и запуска программного проекта с выводом текстовой информации на консоль.