**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ   
НАУКИ І ТЕХНОЛОГІЙ**

КАФЕДРА: «Автоматика та телекомунікації»

ЗВІТ   
З ЛАБОРАТОРНОЇ РОБОТИ № 5

ТЕМА: ДОСЛІДЖЕННЯ ВИКОНАННЯ ПРОГРАМИ В РЕЖИМІ ПЕРЕРИВАННЯ.

ВИКОНАВ:   
студент групи СК2311 (919)  
Герман Лендель   
шифр – 230387

ПЕРЕВІРИВ:   
доцент  
Роман РИБАЛКА

*Мета* лабораторної роботи: дослідити виконання програми в режимі переривання на прикладі задачі обчислення величини струму в опорі; отримати уявлення про Visual Studio, будову і функціонування консольного додатку та додатку Windows Forms.

Пункти порядку виконання лабораторної роботи

Пункт 3

Відкривши папку Lr1 у файловому менеджері операційної системи я побачив такі елементи: папку під назвою «ConsoleApp» та файл, який має назву «Lr1.sln».

Пункт 5.2

Скомпілювавши рішення через меню Build\ Build solution я отримав у вікні Error List помилку –

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Severity | Code | Description | Project | File | Line | Suppression State |
| Error | CS0029 | Cannot implicitly convert type 'string' to 'double' | ConsoleApp | D:\СК2311\_Гончаренко\ЛР\_5\Lr1\ConsoleApp\Program.cs | 14 | Active |

Припускаю, що ця помилка відобразилась через те, що 'string' неможливо конвертувати в 'double'.

Пункт 5.4

Коли ми замінили Console.ReadLine() на Convert.ToDouble() і в якості його параметру додали Console.ReadLine(), тобто вийшло – double U = Convert.ToDouble(Console.ReadLine()); Через ці дії у вікні Error List після компіляції не було помилок, у відмінності від того, коли я робив компіляцію з рядком double U = Console.ReadLine(); (тобто у попередній компіляції була помилка, а у цій – ні).

Пункт 5.7

Запустивши проект в режимі налагодження я ввів значення для напруги – 21 і натиснув Enter, після цього мені треба було ввести значення для опору, я ввів – 10 і знову натиснув Enter, але після цього консоль закрилася, це було досить неочікувано, бо я думав, що треба буде ввести ще значення для І, А.

**Пункт 5.8**

Запустивши проект в режимі без налагодження я ввів такі ж самі значення для напруги і опору, як і в **пункті 5.7,** але тепер в мене не закрилася консоль після цього, а вивелось значення - «І, А: 2,1» та текст «Для продовження натисніть будь-яку клавішу. . .», після цього я натиснув Enter і консоль закрилася, на це я і очікував.

Пункт 5.10

Знову запустивши проект в режимі налагодження та ввівши ті ж самі значення для напруги і опору, вивелось значення - «І, А: 2,1» і текст «Для виходу натисн?ть Enter», припускаю, що в слові «натисніть», замість «і» стоїть знак питання тому, що програма не сприймає букви, які є в українському алфавіті, звісно, можна замінити українську «і» на англійську і цього знака питання більше не буде.

Пункт 5.12

Відкривши папку «**Lr1**»у файловому менеджері операційної системи я побачив папку під назвою «ConsoleApp» і файл рішення, який має назву – «**Lr1**», а далі, відкривши папку «ConsoleApp» я побачив потрібні мені назву файлу проекту – «ConsoleApp» та первинного коду – «Program.cs».

Пункт 6.2

Замінивши double U на int U в тексті програми, а після цього скомпілювавши рішення, я побачив помилку у вікні **Error List** -

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Severity | Code | Description | Project | File | Line | Suppression State |
| Error | CS0266 | Cannot implicitly convert type 'double' to 'int'. An explicit conversion exists (are you missing a cast?) | ConsoleApp | D:\СК2311\_Гончаренко\ЛР\_5\Lr1\ConsoleApp\Program.cs | 14 | Active |

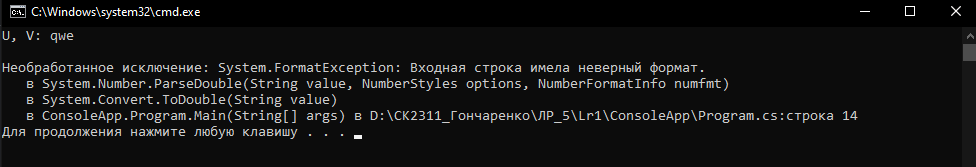
Я думаю, що ця помилка сталася через те, що int займає 32 біта, а double – 64 і ми намагаємось втиснути число з більшою множиною в іншу, яка менше, тому і відбувається цей конфлікт.

Пункт 7.2

Ввевши в якості значення напруги довільний рядок нечислових даних, а саме я ввів – «qwe» та натиснувши клавішу Enter, фокус з консолі перекинуло на Visual Studio і там в коді підсвічувався рядок, в якому була помилка і описано саму помилку, а саме: «System.FormatException: Вхідна строка мала невірний формат.'»

Я думаю, що ця помилка сталася через те, що метод ToDouble конвертує в числа, а ми ввели текст.

Пункт 7.5

Запустивши проект в режимі без налагодження та ввівши в якості значення напруги рядок нечислових данних, я ввів такий же, як і в **пункті 7.2** та натиснувши клавішу Enter, після цих дій я побачив перед собою таку картину: 

Та ж сама інформація, що надав нам Visual Studio, яка була в попередньому пункті та додатково висвітлюється в якому виклику методу сталась проблема і те, на якому рядку Visual Studio минулого разу зупинило своє виконання.

Пункт 8

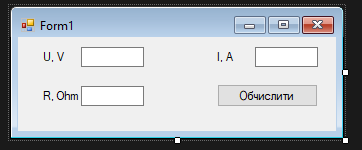
Скріншот остаточного тексту програми прикріпив у додаток 1.

Пункт 10

Відкривши папку «**Lr1**»у файловому менеджері операційної системи я побачив щойно додану папку, яка має назву – «**WinApp**». Я думаю, що ця папка створилася через те, що ми створили новий проект **WinApp** у Visual Studio.

Пункт 12.3

Скріншот форми на наступній сторінці(4).



2

6

5

7

3

1

8

Вікно проекту WinApp з елементами керування: 1, 3, 5 – Label; 2, 4, 8 – TextBox; 6 – Form;

7 – Button.

4

**Таблиця 5.1**

**Елементи керування на формі проекту WinApp**

|  |  |  |  |
| --- | --- | --- | --- |
| № на формі | Назва | Властивість | Значення |
| **1** | label1 | Text | U, V |
| **2** | textBox1 | Name | TextBoxU |
| **3** | label2 | Text | R, Ohm |
| **4** | textBox2 | Name | TextBoxR |
| **5** | label3 | Text | I, A |
| **6** | Form1 | Text | Закон Ома |
| **7** | button1 | Name  Text | button4ik  Обчислити |
| **8** | textBox3 | Name  ReadOnly | TextBoxI  True |

Пункт 12.5

Ввівши у вікні мого додатку довільні цілочислові значення, а саме: для напруги – 21, а для опору я ввів – 10. Натиснувши після цього на кнопку **Обчислити** в результаті кнопка змінила свій вигляд трохи посинівши у контурі, також у першому і другому textBox залишилися ті ж самі значення, після натискання на кнопку обчислення я думав, що у третьому textBox ми отримаємо результат, але там нічого не було, просто пустий елемент керування textBox. Припускаю, що це через те, що ми не дали елементу button1 ніякого призначення, тобто, що саме цій кнопці треба робити.

Пункт 15

Скомпілювавши та запустивши проект в режимі налагодження, а також ввівши довільні цілочислові значення, такі ж самі, як і в **пункті 12.5,** а після цього натиснувши на кнопку **Обчислити,** ми отримали результат у елементі форми textBox3(TextBoxI), а саме – 2,1. Це і є тим самим результатом обчислення значень, які ми вводили до цього.

Пункт 17.1

Скомпілював проект та після цього запустив його в режимі налагодження, у textBox1(TextBoxU) вже було значення – 7,5(з комою, а не крапкою, як це було прописано в коді), а у textBox2(TextBoxR) теж вже було вписане значення, але інше, а саме – 25, тобто, це ті ж самі значення, які ми вводили у коді, коли доповнювали тіло обробника **Form1\_Load.**

Пункт 17.3

У вікні додатку у textBox1(TextBoxU) замінивши цілочисловий розділовий знак з коми на крапку і після цього натиснувши на кнопку **Обчислити** фокус перекинуло з додатку на Visual Studio та виділило рядок, в моєму випадку, 23, як помилку, а саме: «System.FormatException: Вхідна строка мала невірний формат.'». Ми вже стикалися з подібною проблемою. Я думаю, що саме ця проблема виникла через те, що на моїй ОС стоїть українська локалізація, а у нас, в Україні, роздільним знаком є саме кома. Тому, будь-що, окрім коми, буде викликати помилку.

Пункт 18.3

Перемістивши фокус введення до редактора коду і розташувавши вказівник миші поверх змінної U я побачив підказку і значення в ній, а саме U=0. Виконавши аналогічні дії зі змінними R та І я побачив, що і ці змінні мають нульові значення, тобто, R = 0; I = 0.

Пункт 18.7

Таблиця 5.3

**Значення змінних**

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Ім’я (ідентифікатор) змінної | | |
| U | R | I |
| Перед присвоюванням | 0 | 0 | 0 |
| Після присвоювання | 7.5 | 25 | 0.3 |

Пункт 19

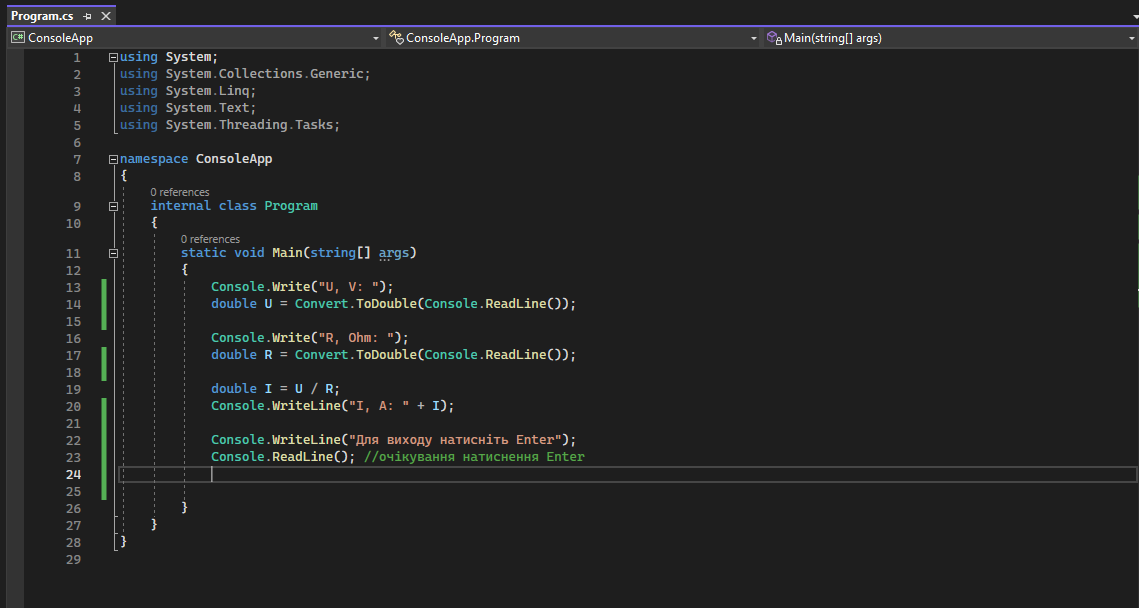
­­­­Висновки

Для досягнення своеї мети я використовував програму Visual Studio Code .На мою думку це найзручніший редактор коду який існує. Тому я з легкістю зробив Windows forms.

1. **Виконання програми в режимі переривання:**
   * Важливим аспектом дослідження було вивчення режиму переривання в програмах. Цей режим дозволяє взаємодіяти з програмою у визначені моменти часу або при виникненні певних подій. У контексті обчислення величини струму в опорі, використання переривань може бути корисним для надання звіту про стан обчислень або для взаємодії з користувачем.
2. **Visual Studio:**
   * Visual Studio є потужним інтегрованим середовищем розробки, яке надає різноманітні інструменти для створення програм для різних платформ. Уперше знайомство з цією платформою дозволило зрозуміти її можливості та ефективність для розробки програм.
3. **Консольний додаток та додаток Windows Forms:**
   * Розробка консольного додатку дозволила ознайомитися з базовими концепціями програмування в C# та роботою з текстовим інтерфейсом. З іншого боку, створення додатку Windows Forms розкрило можливості графічного інтерфейсу, що полегшує взаємодію з користувачем.
4. **Завдання з обчислення величини струму в опорі:**
   * Використання програм для обчислення величини струму в опорі дозволило застосувати отримані знання на практиці. Ця задача може бути реалізована як у консольному додатку, так і у додатку Windows Forms, залежно від конкретних потреб користувача.

Загалом, лабораторна робота дала можливість вивчити та випробувати основні аспекти програмування в середовищі Visual Studio.

Пункт 8



Пункт 19

