Міністерство освіти України

Національний технічний університет "ХПІ"

кафедра “ННІКНІТ"

**Звіт**

**Лабораторна робота 1**

з дисципліни "Програмна технологія .NET"

Виконав: студент групи КН-921Г

Верещак А. Г.

Перевірив:

Івашко А.В.

Харків 2023

**Зміст**

[Завдання 3](#__RefHeading___Toc1950_2028602607)

[Хід роботи 6](#__RefHeading___Toc1952_2028602607)

[Завдання 1: 6](#__RefHeading___Toc1954_2028602607)

[Завдання 2: 7](#__RefHeading___Toc1956_2028602607)

[Завдання 3: 8](#__RefHeading___Toc1958_2028602607)

[Завдання 5: 10](#__RefHeading___Toc1960_2028602607)

[Завдання 6: 11](#__RefHeading___Toc1962_2028602607)

[Завдання 7: 12](#__RefHeading___Toc1964_2028602607)

[Завдання 8: 13](#__RefHeading___Toc1966_2028602607)

[Завдання 10: 17](#__RefHeading___Toc1968_2028602607)

[Висновок 19](#__RefHeading___Toc1970_2028602607)

Лабораторна робота №1. **Використання базових типів, консольного введення / виведення та операторів мови C# .**

**Мета роботи**: набути навички в написанні програм з використання базових типів, операторів та засобів введення / виведення.

# Завдання

1. Використовуючи завдання власного варіанта до лабораторної роботи 2 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.
2. Виконати минуле завдання, покращивши зовнішній вигляд таблиці із вихідними результатами за допомогою символів псевдографіки (┐, ┌, ┘, └, ├, ┤, ┬, ┴ та ін.) та налаштування кольорів консолі, наприклад, числа відобразити одним кольором, текст – іншим, таблицю – іншим від попередніх.
3. Використовуючи завдання власного варіанта до лабораторної роботи 3 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.
4. Виконати минуле завдання, покращивши гнучкість програми обчислення виразу, якщо вхідні дані є некоректними із точки зору математичних правил (нуль в знаменнику, від’ємне значення під знаком квадратного кореня та ін.). Для цього використати механізм обробки відповідних виключень (try ... catch ... finally)
5. Використовуючи завдання власного варіанта до лабораторної роботи 5 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.
6. Виконати минуле завдання без використання оператору **if** у будь-яких формах
7. Використовуючи завдання власного варіанта до лабораторної роботи 6 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.
8. Виконати минуле завдання, переробивши алгоритм вираховування суми ряду таким чином:
   1. похибка та кількість членів ряду мають вводитися як вхідні параметри у відповідні змінні;
   2. сума ряду вираховується або при досягненні заданої похибки, або при досягненні заданої кількості членів ряду із видачою відповідного повідомлення.
9. Використовуючи завдання власного варіанта до лабораторної роботи 7 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.
10. Виконати минуле завдання, покращивши зовнішній вигляд побудованого графіку за допомогою символів псевдографіки ⸜, ⸝, ⸺, \, /, ­ та ін.) та налаштування кольорів консолі, наприклад, числа відобразити одним кольором, пояснюючий текст – іншим, графік – іншим від попередніх.

**Що таке базові типи даних в мові С#?**

Базові типи визначають основні види даних, які можна використовувати в мові програмування C#. Деякі з найпоширеніших базових типів включають в себе:

* int: Цілі числа.
* double: Дійсні числа з плаваючою комою.
* bool: Логічний тип, який може приймати значення true або false.
* char: Символьний тип для представлення одного символу.
* string: Рядок, який використовується для представлення текстових даних.

**Що таке консольне введення/виведення (Console Input/Output)?**

Консольне введення та виведення в C# дозволяють взаємодіяти з користувачем через командний рядок. Для виведення даних на консоль використовується Console.WriteLine або Console.Write, а для зчитування введених даних - Console.ReadLine або інші методи.

**Що таке Оператори (Operators)?**

Оператори в мові C# використовуються для виконання різних операцій над даними. Деякі з основних операторів включають в себе:

* Арифметичні оператори (+, -, \*, /, %).
* Оператори порівняння (==, !=, <, >, <=, >=).
* Логічні оператори (&&, ||, !).
* Оператори присвоєння (=, +=, -=, \*=, /=).
* Оператори умовного виразу (тернарний оператор ? :).

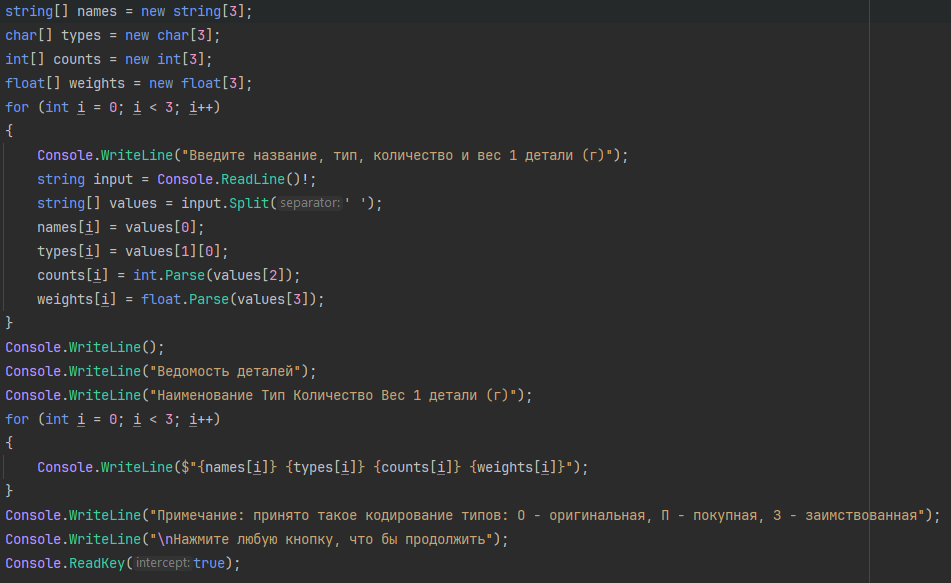
Ці концепції є основою для розробки програм на мові C# і використовуються для обробки даних, взаємодії з користувачем та виконання різних обчислень і логіки в програмах. Вони допомагають створювати різноманітні програми, від простих консольних застосунків до складних програм з графічним інтерфейсом користувача.

# Хід роботи

# Завдання 1:

Створюємо новий проект Task\_1.

Для вирішення поставленого завдання пишемо наступний код:

Рисунок 1.1 — Код проекту Task\_1

При запуску отримуємо наступне:

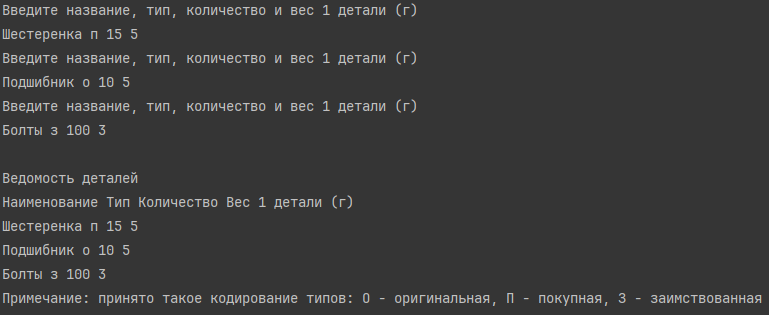


Рисунок 1.2 — Виконання проекту Task\_1

# Завдання 2:

Створюємо новий проект Task\_2.

Змінюємо код, згідно з завданням. Модифікована частина:

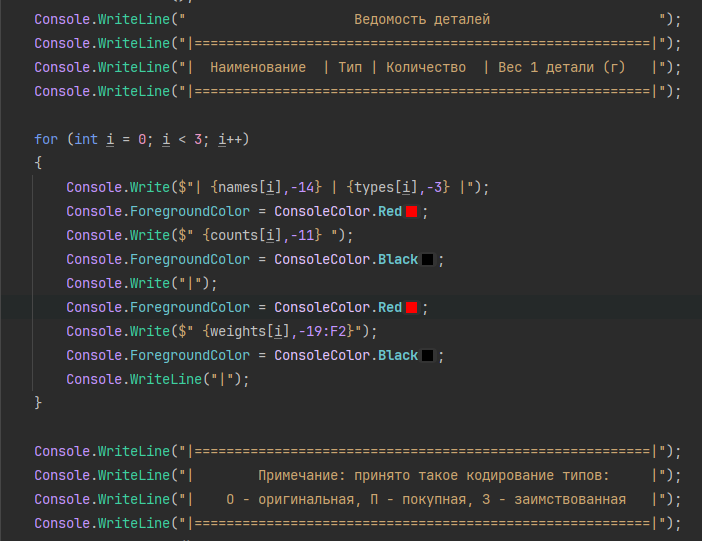


Рисунок 1.3 — Код проекту Task\_2

При запуску отримуємо наступне:

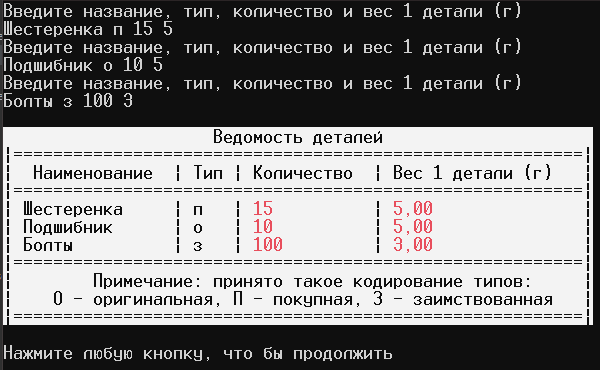
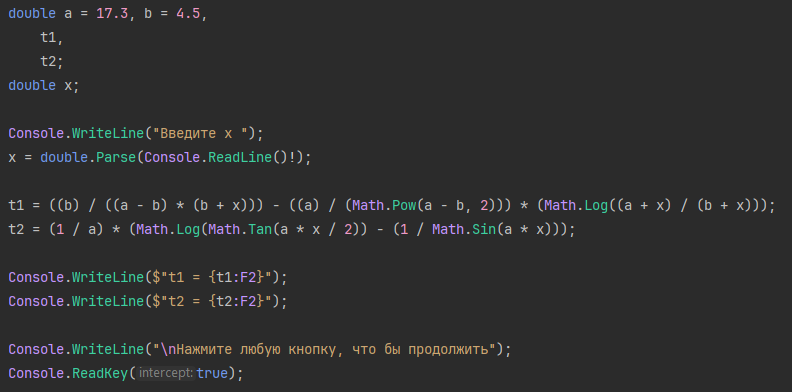


Рисунок 1.4 — Виконання проекту Task\_2

# Завдання 3:

Створюємо новий проект Task\_3.

Для вирішення поставленого завдання пишемо наступний код:

Рисунок 1.5 — Код проекту Task\_3

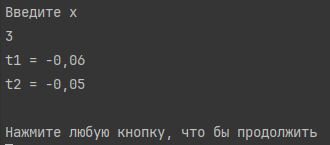
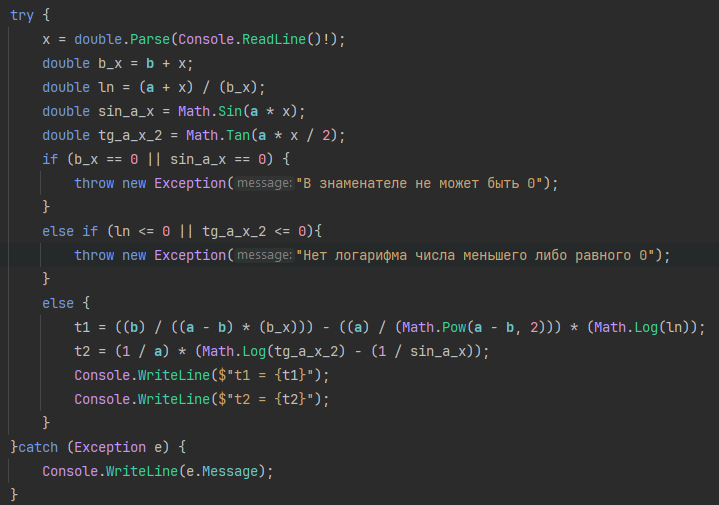
Результат виконання коду:

Рисунок 1.6 — Виконання проекту Task\_3

**Завдання 4**:

Створюємо новий проект Task\_4.

Змінюємо код, згідно з завданням. Модифікована частина:

Рисунок 1.7 — Код проекту Task\_4

Результат виконання програми:

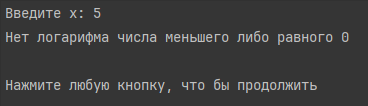


Рисунок 1.8 — Виконання проекту Task\_4

# **Завдання 5**:

Створюємо новий проект Task\_5.

Для вирішення поставленого завдання пишемо наступний код:

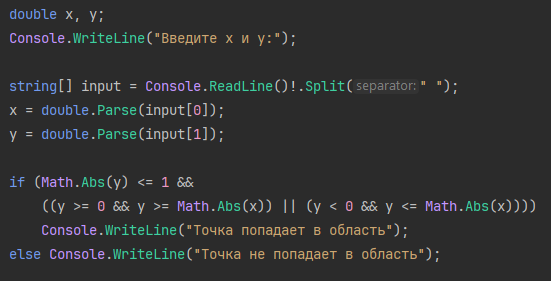


Рисунок 1.9 — Код проекту Task\_5

Результат виконання програми:

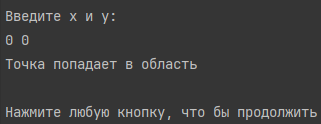


Рисунок 1.10 — Виконання проекту Task\_5

# **Завдання 6**:

Створюємо новий проект Task\_6.

Змінюємо код, згідно з завданням. Модифікована частина:

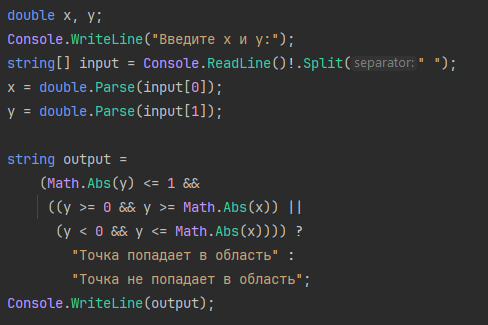


Рисунок 1.11 — Код проекту Task\_6

Результат виконання програми:

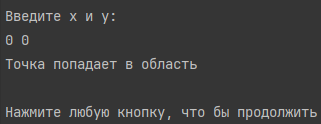
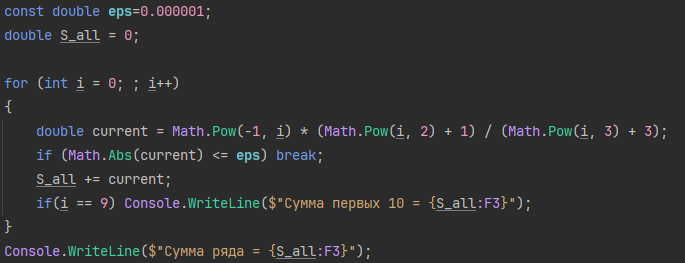


Рисунок 1.12 — Виконання проекту Task\_6

# **Завдання 7**:

Створюємо новий проект Task\_7.

Для вирішення поставленого завдання пишемо наступний код:

Рисунок 1.13 — Код проекту Task\_7

Результат виконання програми:

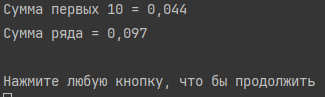
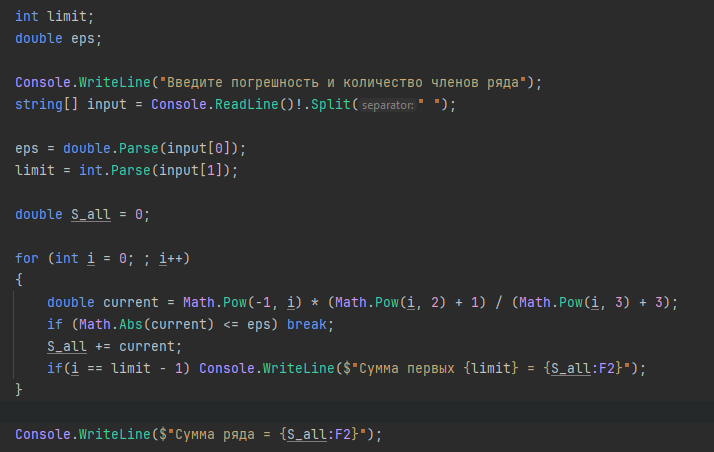


Рисунок 1.14 — Виконання проекту Task\_7

# **Завдання 8**:

Створюємо новий проект Task\_8.

Змінюємо код, згідно з завданням. Модифікована частина:

Рисунок 1.15 — Код проекту Task\_8

Результат виконання програми:

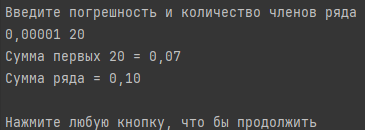


Рисунок 1.16 — Виконання проекту Task\_8

**Завдання 9**:

Створюємо новий проект Task\_9.

Для вирішення поставленого завдання пишемо наступний код:

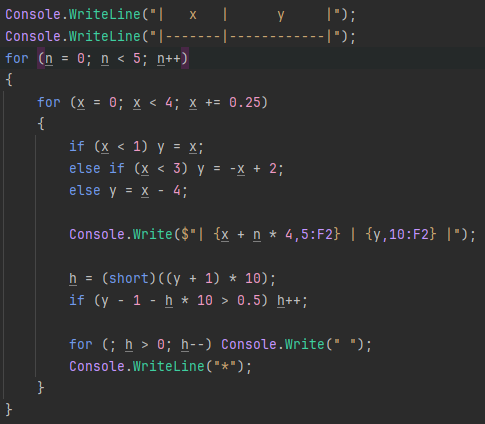


Рисунок 1.17 — Код проекту Task\_9

Результат виконання програми:

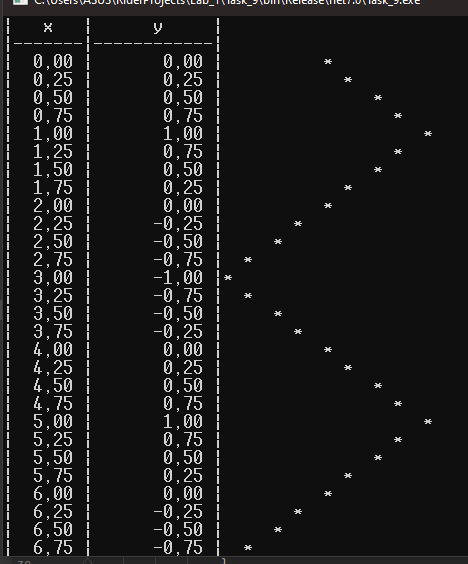


Рисунок 1.18 — Виконання проекту Task\_9

# **Завдання 10**:

Створюємо новий проект Task\_10.

Змінюємо код, згідно з завданням. Модифікована частина:

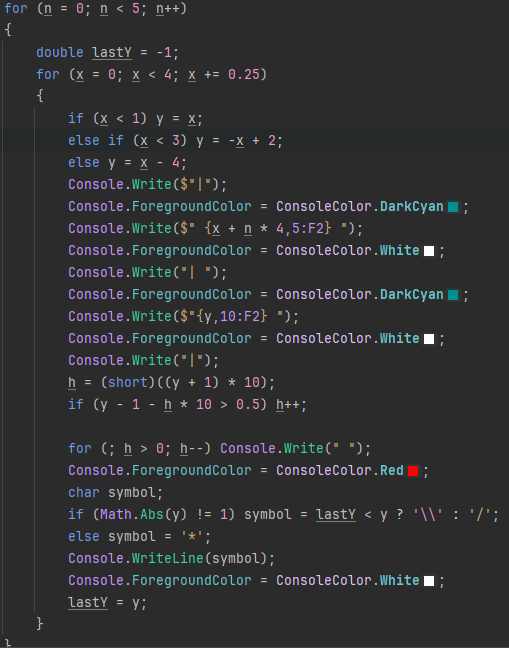
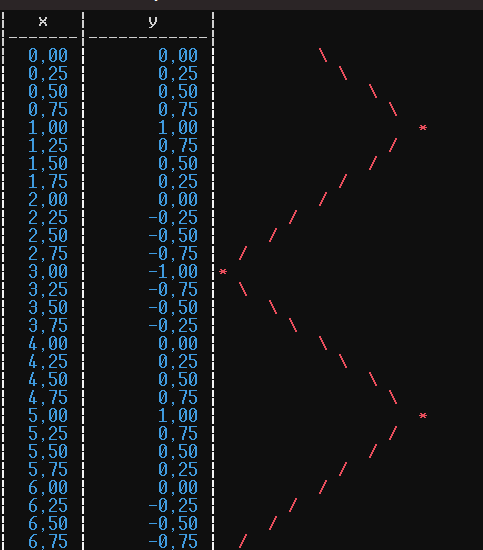


Рисунок 1.19 — Код проекту Task\_10

Результат виконання програми:



\

Рисунок 1.20 — Виконання проекту Task\_10

# Висновок

Було набуто навичок в написанні програм з використання базових типів, операторів та засобів введення / виведення. Використання базових типів даних, консольного введення/виведення та операторів мови C# дозволяє розробникам створювати програми, які можуть взаємодіяти з користувачем, обробляти дані і виконувати різні операції. Це є ключовими складовими для створення функціональних і корисних програм на мові C#.