

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 2

На тему: «Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Змінні. Константи. Типи даних. Розмір типів даних. Ввід вивід. Базові операції та вбудовані функції. Коментарі.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 1
Алготестер Лабораторної Роботи № 1
Практичних Робіт до блоку № 2

Виконав:

Студент групи ШІ-12
Михальчук Антон Євгенійович

Тема роботи:

Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Змінні. Константи. Типи даних. Розмір типів даних. Ввід вивід. Базові операції та вбудовані функції. Коментарі.

Мета роботи:

Закріпити та розвинути набуті знання з програмування на мові C++ через виконання практичних завдань, зосереджених на базових концепціях та конструкціях мови. Зокрема, метою є навчитися застосовувати лінійні та розгалужені алгоритми, ефективно використовувати умовні та логічні оператори, працювати зі змінними, константами та різними типами даних, розуміти їх розмір, організовувати ввід і вивід інформації, застосовувати базові операції та вбудовані функції, а також писати коментарі для підвищення читабельності коду.

Теоретичні відомості:

- 1) Теоретичні відомості з переліком важливих тем:
 - Тема №*.1: C++ Basics
- 2) Індивідуальний план опрацювання теорії:
 - Тема №*.1: C++ Basics
 - o Джерела Інформації
 - Відео. <https://www.youtube.com/watch?v=2UDMGCCRCjo>
 - Стаття. <https://www.w3schools.com/cpp/>
 -
 - o Що опрацьовано:
 - Вивчив базовий синтаксис та семантику мови C++.
 - Особливу увагу приділяв таким темам, як оголошення змінних, типи даних (цілі, з плаваючою точкою, логічні, символи), оператори, умови та цикли.
 - o Статус: Ознайомлений
 - o Початок опрацювання теми: 15.09.2024
 - o Звершення опрацювання теми: 26.10.2024

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 VNS Lab 1 - Task 1

- Деталі завдання:
 - o Обчислити значення виразу при різних дійсних типах даних (float й double).
 - o Пояснити отримані результати.
- Важливі деталі для врахування в імплементації програми:
 - o Обчислення варто виконувати з використанням проміжних змінних.

Завдання №2 VNS Lab 1 - Task 2

- Деталі завдання:

- o Обчислити значення виразу при різних дійсних типах даних (float й double).
- o Пояснити отримані результати.
- Важливі деталі для врахування в імплементації програми:
 - o Обчислення варто виконувати з використанням проміжних змінних.

Завдання №3 Lab 1v1

- Деталі завдання:
 - o У вашого персонажа є Н хітпойнтів та М мани.
 - o Персонаж 3 рази використовує закляття, кожне з яких може використати хітпойнти та ману одночасно.
 - o Якщо якесь закляття забирає і хітпойнти і ману - ваш персонаж програє, отже для виграшу треба використовувати при одному заклинанні АБО хітпойнти, АБО ману.
 - o Якщо в кінці персонаж буде мати додатню кількість хітпойнтів та мани (Н, М>0) - він виграє, в іншому випадку програє.
 - o Ваше завдання у випадку виграшу персонажа вивести YES, вивести NO у іншому випадку.
- Важливі деталі для врахування в імплементації програми:
 - o Input: 2 цілих числа Н та М - хітпойнти та мана персонажа 3 рядки по 2 цілих числа, hі та mі - кількість хітпойнтів та мани, які ваш персонаж потратить за хід на І заклинання
 - o Output YES - якщо ваш персонаж виграє, NO - у всіх інших випадках
 - o Обмеження: цілі числа менше 10^{12} , всі змінні будуть типу.

Завдання №4 Особистий порадижник

- Деталі завдання:
 - o Навчитися користуватися операторами галуження для структурування логіки програм.
 - o Якщо користувач вводить будь-яку іншу умову, запропонуйте йому ввести дійсну умову.
- Важливі деталі для врахування в імплементації програми:
 - o if else - щоб вирішити, чи повинен користувач взяти куртку чи ні.
 - o if, else if - щоб надати рекомендацію щодо активності (прогулянка, футбол, настільні ігри, etc).
 - o switch case - для визначення типу рекомендованого взуття.

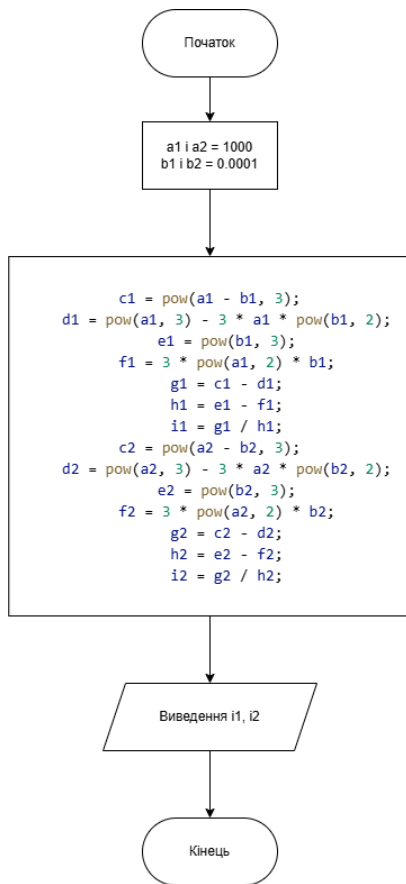
Завдання №5 Торт для Петрика

- Деталі завдання:
 - o Круги накладаються одне на одне.
 - o Неперекруті площі кругів є однаковими
 - o Потрібно знайти площу найменшого круга
 - o Дано кількість кругів і радіуси
- Важливі деталі для врахування в імплементації програми:
 - o Створимо формулу згідно умов задачі $\frac{r^2 \pi}{n} = r_p^2 \pi$
 - o Виведемо формулу для розв'язку: $r_p = \frac{r}{\sqrt{n}}$

2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 VNS Lab 1 - Task 1

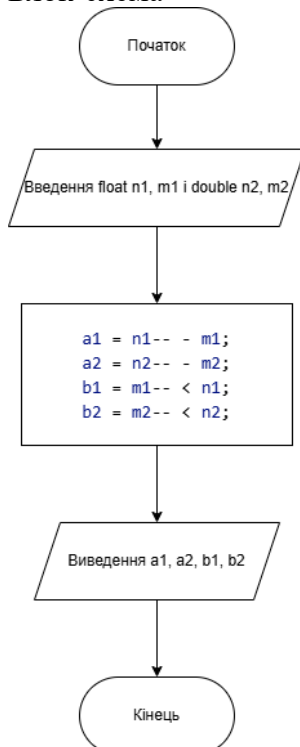
- Блок-схема:



- Планований час на реалізацію: 10 хв

Програма №2 VNS Lab 1 - Task 2

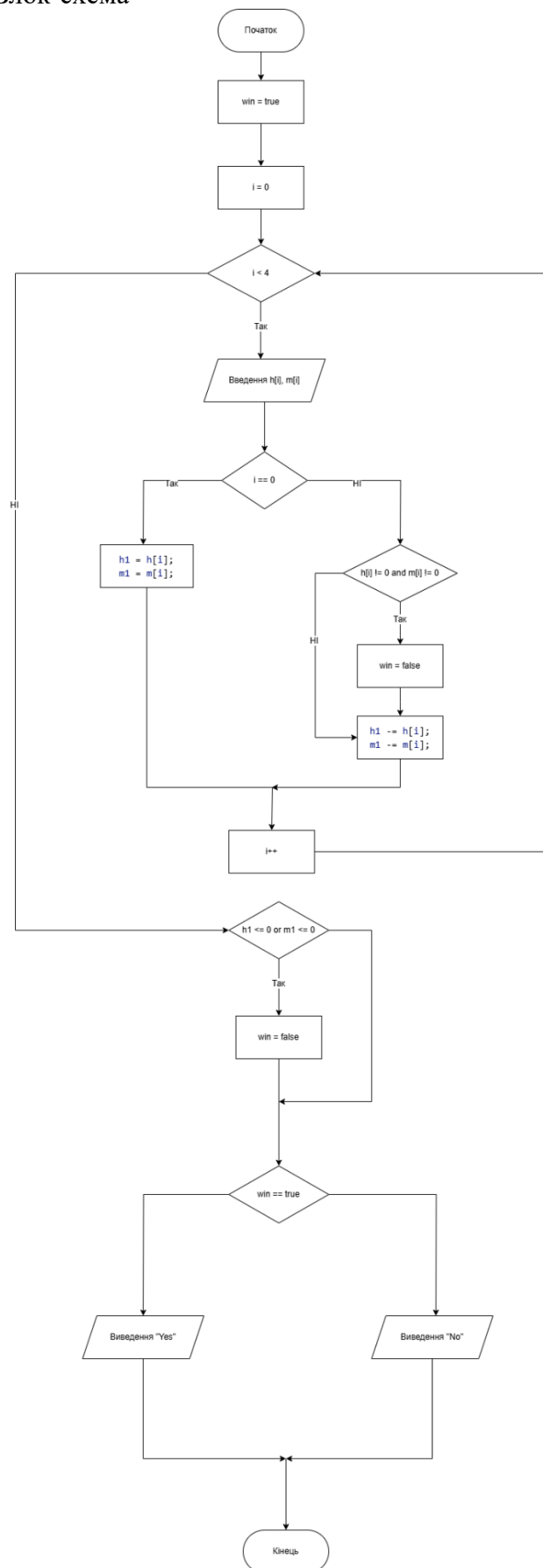
- Блок-схема



- Планований час на реалізацію: 10 хв.

Програма №3 Lab 1v1

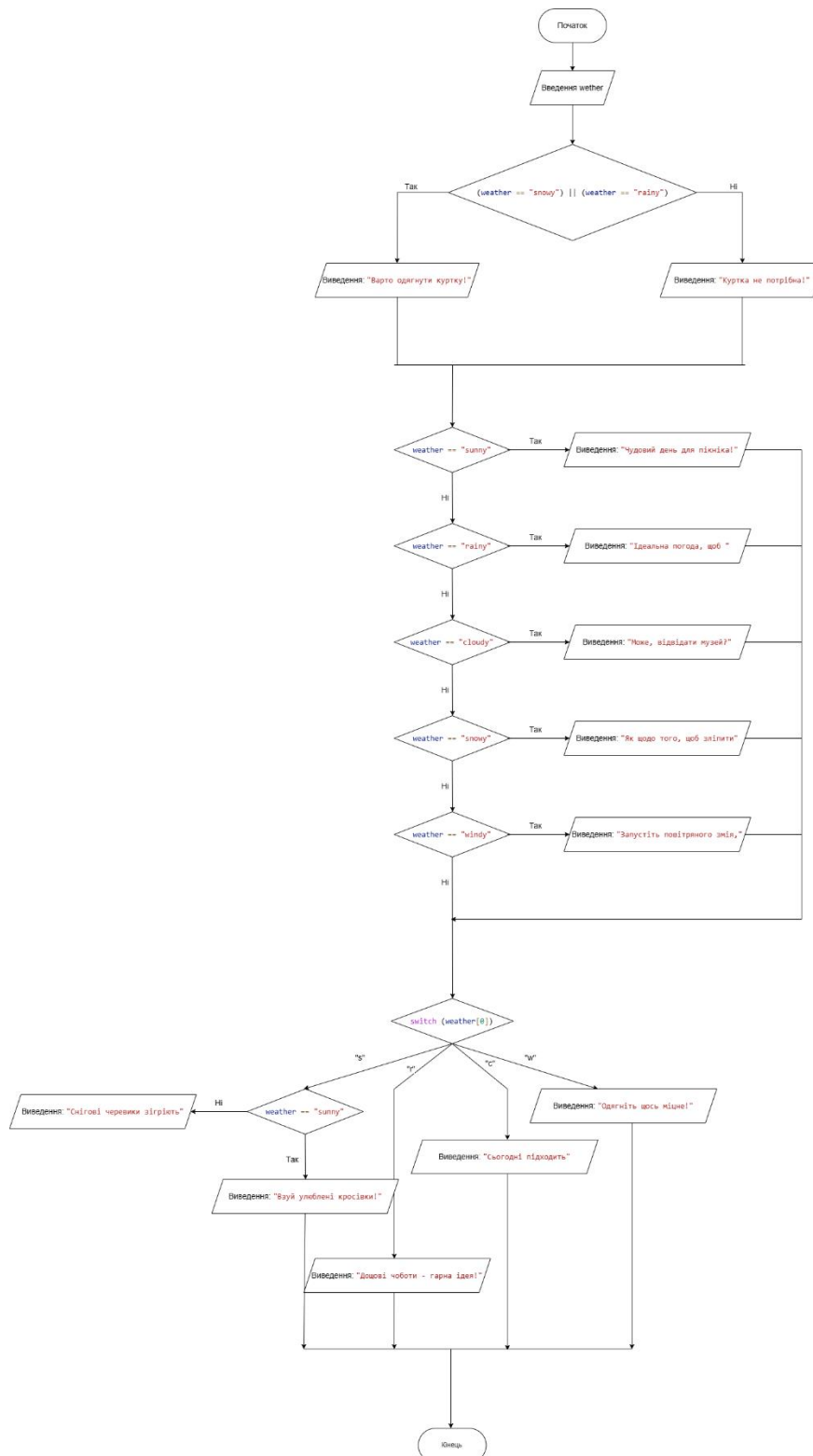
- Блок-схема



- Планований час на реалізацію: 20 хв.

Програма №4 Особистий поради́ник

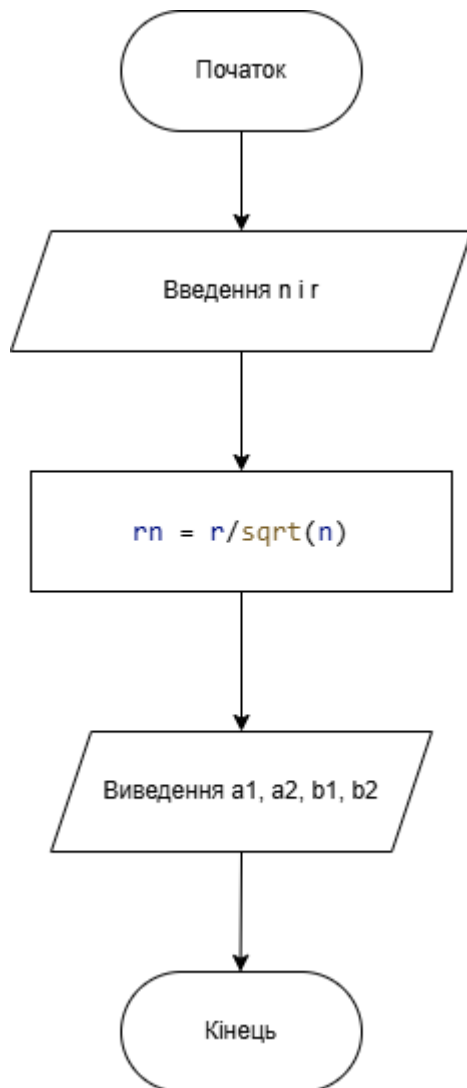
- Блок-схема



- Планований час на реалізацію: 30 хв.

Програма №5 Торт для петрика

- Блок-схема



- Планований час на реалізацію: 20 хв.

4. Код програм з посиланням на зовнішні ресурси:

Завдання №1

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/14/commits/62fe4ab1eb381339e6521f327fd5cd30fb3b7f36#diff-46264ad872955422e4e34210db3f058f44ee114be3c319dc80407b01af437fef

```
#include <iostream>
#include <cmath>
using namespace std;
```

```

int main()
{
    float a1, b1, c1, d1, e1, f1, g1, h1, i1;
    double a2, b2, c2, d2, e2, f2, g2, h2, i2;

    a1 = 1000;
    b1 = 0.0001;

    c1 = pow(a1 - b1, 3);
    d1 = pow(a1, 3) - 3 * a1 * pow(b1, 2);
    e1 = pow(b1, 3);
    f1 = 3 * pow(a1, 2) * b1;
    g1 = c1 - d1;
    h1 = e1 - f1;
    i1 = g1 / h1;

    cout << i1 << endl;

    a2 = 1000;
    b2 = 0.0001;

    c2 = pow(a2 - b2, 3);
    d2 = pow(a2, 3) - 3 * a2 * pow(b2, 2);
    e2 = pow(b2, 3);
    f2 = 3 * pow(a2, 2) * b2;
    g2 = c2 - d2;
    h2 = e2 - f2;
    i2 = g2 / h2;

    cout << i2;

    return 0;
}

```

Завдання №2

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/14/commits/62fe4ab1eb381339e6521f327fd5cd30fb3b7f36#diff-b2798507a541c29cad04f26b69e4665c0cce6ee3e917dc6c7e01d5cca50182a0

```

#include <iostream>

using namespace std;

int main()

```



```

{
    float n1, m1, a1;
    double n2, m2, a2;
    bool b1, b2;

    cout << "Enter value for n (in float): ";
    cin >> n1;
    cout << "Enter value for m (in float): ";
    cin >> m1;

    cout << "Enter value for n (in double): ";
    cin >> n2;
    cout << "Enter value for m (in double): ";
    cin >> m2;

    a1 = n1-- - m1;
    a2 = n2-- - m2;

    cout << "Task2.1 output (from float values): " << a1 << endl;
    cout << "Task2.1 output (from double values): " << a2 << endl;

    b1 = m1-- < n1;
    b2 = m2-- < n2;

    cout << "Task2.2 output (from float values): " << b1 << endl;
    cout << "Task2.2 output (from double values): " << b2 << endl;

    return 0;
}

```

Завдання №3

```

#include <iostream>
using namespace std;

int main()
{
    long long h[4], m[4], h1, m1;
    bool win = true;

    for (int i = 0; i < 4; i++)
    {
        cin >> h[i] >> m[i];
    }
}

```

```

    if (i == 0)
    {
        h1 = h[i];
        m1 = m[i];
    }

    else
    {
        if ((h[i] != 0) && (m[i] != 0))
        {
            win = false;
        }
        h1 -= h[i];
        m1 -= m[i];
    }
}

if ((h1 <= 0) || (m1 <= 0))
{
    win = false;
}

if (win) {
    cout << "YES";
}
else {
    cout << "NO";
}

return 0;
}

```

Завдання №4

```

#include <iostream>
#include <string>
using namespace std;

int main()
{
    string weather;
    cout << "Введіть поточні погодні умови (sunny, rainy, cloudy, snowy, windy): ";
    cin >> weather;

    if ((weather == "snowy") || (weather == "rainy"))
    {
        cout << "Варто одягнути куртку!" << endl;
    }
}

```

```

    }
    else
    {
        cout << "Куртка не потрібна!" << endl;
    }

    if (weather == "sunny")
    {
        cout << "Чудовий день для пікніка!" << endl;
    }
    else if (weather == "rainy")
    {
        cout << "Ідеальна погода, щоб читати книгу всередині!" << endl;
    }
    else if (weather == "cloudy")
    {
        cout << "Може, відвідати музей?" << endl;
    }
    else if (weather == "snowy")
    {
        cout << "Як щодо того, щоб зліпити сніговика?" << endl;
    }
    else if (weather == "windy")
    {
        cout << "Запустіть повітряного змія, якщо він у вас є!" << endl;
    }

    if (weather == "sunny" || weather == "rainy" || weather == "cloudy"
|| weather == "snowy" || weather == "windy") {
        switch (weather[0]) {
            case 's':
                if (weather == "sunny") {
                    cout << "Взуй улюблені кросівки!" << endl;
                } else if (weather == "snowy") {
                    cout << "Снігові черевики зігріють ваші ноги!" << endl;
                }
                break;
            case 'r':
                cout << "Дощові чоботи - гарна ідея!" << endl;
                break;
            case 'c':
                cout << "Сьогодні підходить будь-яке взуття." << endl;
                break;
            case 'w':
                cout << "Одягніть щось міцне!" << endl;
                break;
        }
    }
}

```

```
    return 0;  
}
```

Завдання №5

5. Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1 Деталі по виконанню і тестуванню програми

Я отримав різні результати для виразів з типами float та double через різну точність представлення чисел в пам'яті. Тип float має точність до 7 знаків після коми, тоді як double — до 15-16 знаків. Коли я обчислював складні вирази, наприклад, піднесення до степеня або віднімання дуже близьких чисел, похибки накопичувалися і впливали на кінцевий результат. Через меншу точність float результат з ним виявився менш точним.

Точніший результат я отримав, використовуючи тип double.

```
PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12>  
bugLauncher.exe' '--stdin=Microsoft-MIEngine-In-s1he0a5h.  
-pid=Microsoft-MIEngine-Pid-plsnxscz.0w0' '--dbgExe=C:\ms  
1.28  
1  
PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12>
```

Час затрачений на виконання завдання: 15 хв.

Завдання №2 Деталі по виконанню і тестуванню програми

Я отримав олікові результати з різними типами даних, адже операції які ми над ними проводили не потребували складних обрахунків щоб з'явилася похибка.

```

Enter value for n (in float): 0.00000000007
Enter value for m (in float): 0.00000000005
Enter value for n (in double): 0.00000000007
Enter value for m (in double): 0.00000000005
Task2.1 output (from float values): 2e-11
Task2.1 output (from double values): 2e-11
Task2.2 output (from float values): 0
Task2.2 output (from double values): 0

```

Час затрачений на виконання завдання: 10 хв.

Завдання №3 Деталі по виконанню і тестуванню програми

Виконано на алготестері

Lab 1v1 - Lab 1v1	C++ 23	Accepted	0.003	1.219	1857679
-------------------	--------	----------	-------	-------	---------

Завдання №4 Деталі по виконанню і тестуванню програми

```

PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12> & 'c:\Users\Lenovo
bugLauncher.exe' '--stdin=Microsoft-MIEngine-In-wmslpwjd.s1y' '--stdout=Micr
-pid=Microsoft-MIEngine-Pid-vfqutct.r3q' '--dbgExe=C:\msys64\ucrt64\bin\gdb
Введіть поточні погодні умови (sunny, rainy, cloudy, snowy, windy): cloudy
Куртка не потрібна!
Може, відвідати музей?
Сьогодні підходить будь-яке взуття.

```

Завдання №5 Деталі по виконанню і тестуванню програми

Виконав окреме випадкове завдання на алготестері: «Торт для Петрика»

0191 - Торт для Петрика	C++ 23	Accepted	0.003	1.188	1857731
-------------------------	--------	----------	-------	-------	---------

6. Кооперація з командою:

- Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло

Михальчук Антон Євгенійович (Ви показуєте презентацію)

Зупинити презентацію

18:02 | nud-ejsb-cno

In progress Lazarevych ... Done Lazarevych ... To Do Mykhalkchuk ... In progress Mykhalkchuk ... Done Mykhalkchuk

- + Add a card
- + Add a card
- + Add a card
- + Add a card
- + Add a card
- + Add a card
- + Add a card
- + Add a card
- + Add a card
- + Add a card

Anton Mykhalkchuk - Epic 2 Task 1 - Theory Education Activities

Anton Mykhalkchuk - Epic 2 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)

Anton Mykhalkchuk - Epic 2 Task 3 - Lab# programming: VNS Lab 1 Task 1

Anton Mykhalkchuk - Epic 2 Task 4 - Lab# programming: VNS Lab 1 Task 2

Anton Mykhalkchuk - Epic 2 Task 5 - Lab# programming: Algotester Lab 1 Task 1

Anton Mykhalkchuk - Epic 2 Task 6 - Practice# programming: Class Practice Task

Anton Mykhalkchuk - Epic 2 Task 7 - Practice# programming: Self Practice Task

Anton Mykhalkchuk - Epic 2 Task 8 - Practice# programming: Self Practice Task

+ Add a card

- Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло

Михальчук Антон Євгенійович (Ви показуєте презентацію) Зауваження

Зупинити презентацію

18:06 | nud-ejsb-cno

terminal

468 Branches 149 Tags

config Update xambly to 3.2311.2 (#1642) 10 months ago

github Add input scope startup setting (#1790) 2 days ago

ruget Auto format our XAML files and enforce in CI (#9388) 3 years ago

vscode Add VSCode config for better out-of-box LSP support (#16...) 6 months ago

build build: fix the TSA configuration (#17676) last week

dep Use WinGet API to improve Quick Fix results (#17614) last month

doc Add Warnings to Settings UI (#17933) yesterday

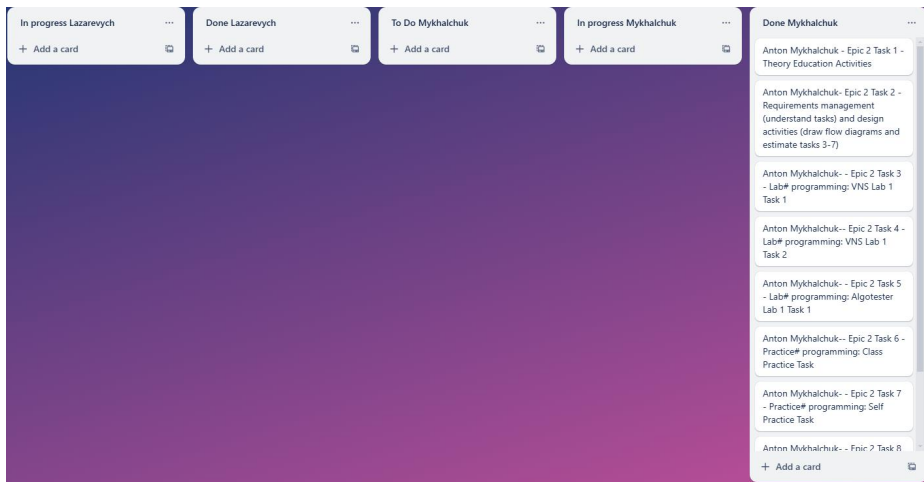
on A minor ConPTY refactoring: Google Windows Edition (#17... 2 months ago

res Update Cascadia Code to 2404.23 (#17137) 5 months ago

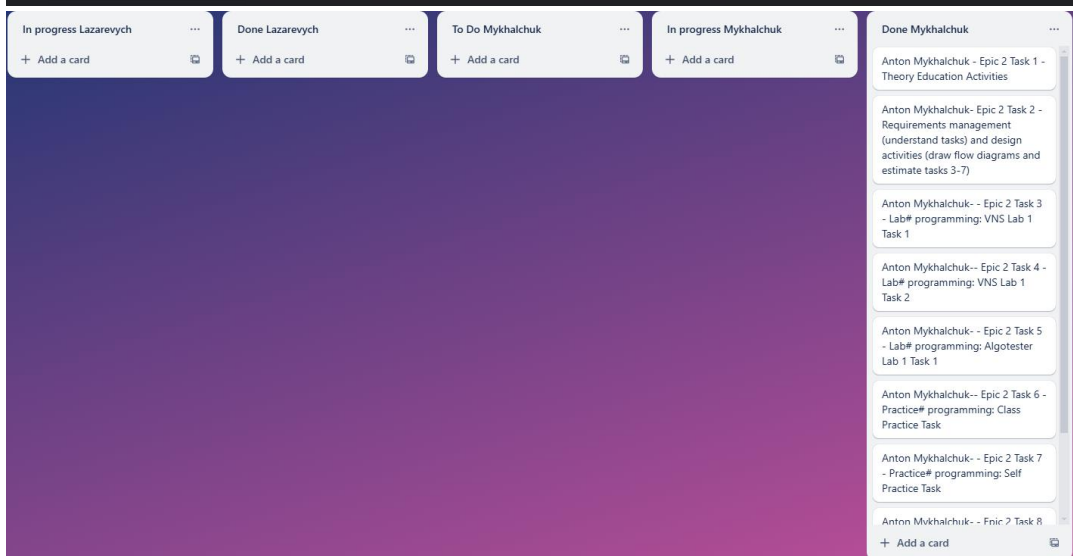
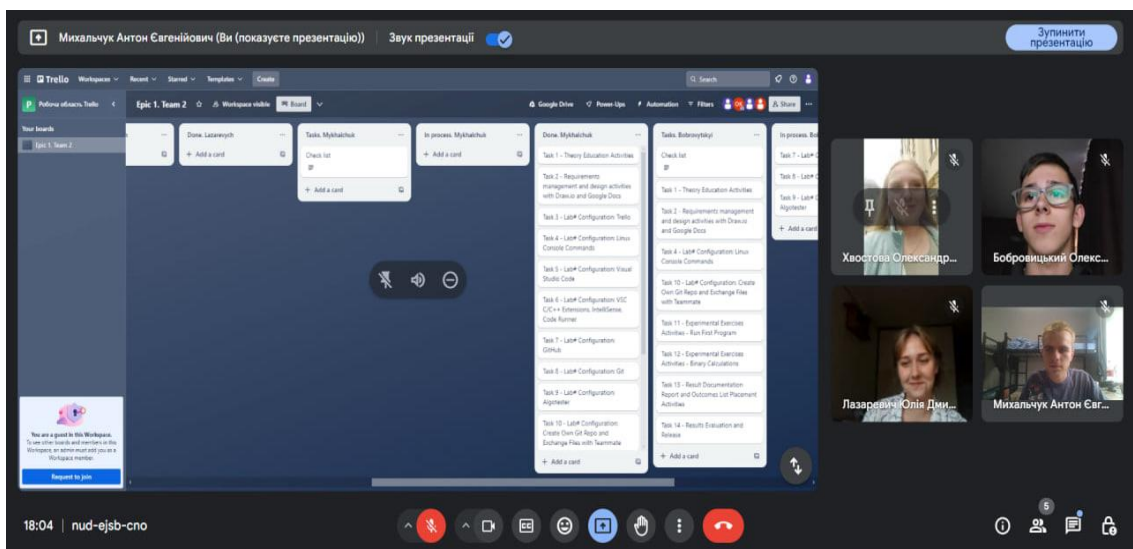
samples Added experimental pixel shader image path (#14073) 6 months ago

136 watching

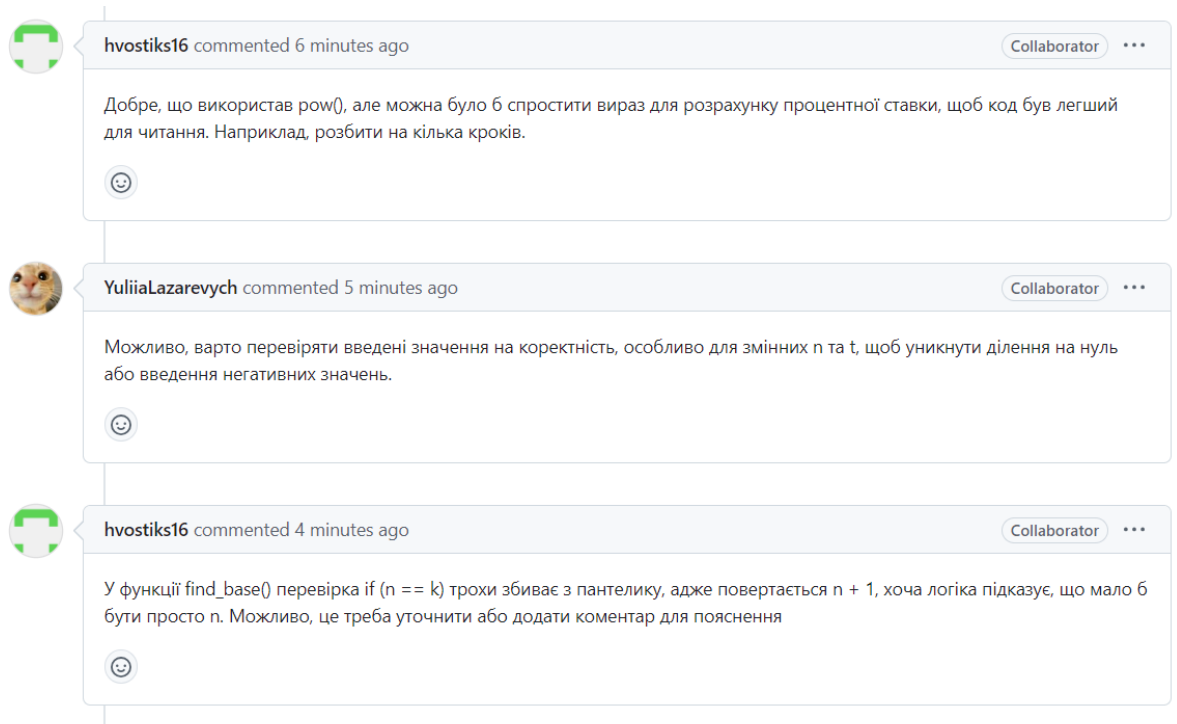
Хвостова Олександра Ан... Бобровицкий Олександр... Лазаревич Юлія Дми... Михальчук Антон Євг...



- Скрін з 3-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло (опційно)



- Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи



Висновки:

Важливість Інструментів: Використання сучасних інструментів, таких як пакетні менеджери та інтегровані середовища розробки, є критично важливим для ефективної роботи програміста. Вони не лише спрощують процес встановлення та налаштування програмного забезпечення, але також забезпечують потужні функції для оптимізації розробки коду.

Розуміння Команд Терміналу: Вивчення команд терміналу в Linux-орієнтованих середовищах допомагає розвинути навички роботи з командним рядком, що є основою для багатьох операцій у розробці та адмініструванні систем.

Значення Дебагінгу: Використання дебагера та лінера показує, як важливо виявляти помилки та покращувати якість коду. Завдяки цим інструментам стало можливим ефективно тестувати і вдосконалювати програми, що зменшує час на виправлення помилок.

Структуроване Вивчення Програмування: Вивчення основ програмування на C++ дозволяє створити базу знань, необхідну для розробки складних програм. Основи, такі як оголошення змінних, типи даних, оператори, умови та цикли, є критично важливими для формування навичок програмування.

Практичний Досвід: Виконання практичних завдань, таких як обчислення складних відсотків і робота з системами числення, демонструє застосування теоретичних знань у

реальних ситуаціях. Це сприяє глибшому розумінню концепцій програмування і дозволяє розвивати навички вирішення проблем.

Співпраця в Команді: Використання Git та GitHub показує, наскільки важливим є управління версіями у командній роботі. Ці інструменти дозволяють ефективно спілкуватися в команді, вести облік змін і спростити інтеграцію роботи кожного члена команди.