Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Розрахунково-графічної роботи**

***з дисципліни:*** «Мови та парадигми програмування»

***Виконала:***

студент групи ШІ-12

Ляшеник Христина Тарасівна

# **Мета роботи:**

Посилення набутих знань та навичок у програмуванні через практичне застосування. Отримання практичного досвіду у створенні та аналізі алгоритмів для вирішення завдань.

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Practice Work - Task 1

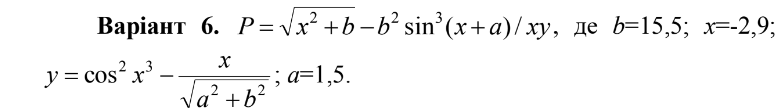
* Варіант завдання - 6
* Деталі завдання - Розробити лінійний алгоритм для розв’язання задачі.

Рисунок 1 Завдання №1

Завдання №2 VNS Practice Work – Task 2

* Варіант завдання - 25
* Деталі завдання - Розробити алгоритм, що розгалужується для розв’язання задачі

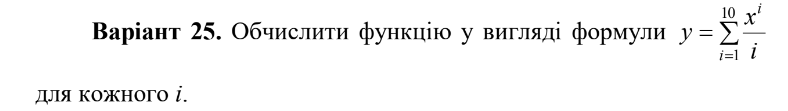


Рисунок 2 Завдання №2

Завдання №3 VNS Practice Work – Task 3\_1

* Варіант завдання - 20
* Деталі завдання

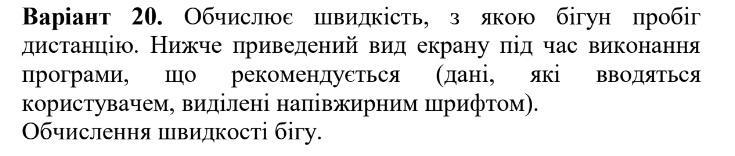
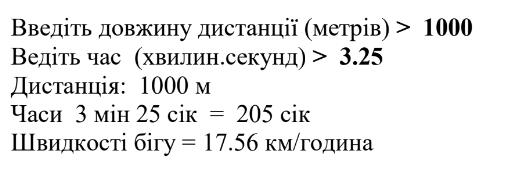
 

Рисунок 3 Завдання № 3

Завдання №4 VNS Practice Work – Task 3\_2

* Варіант завдання - 20

Деталі завдання

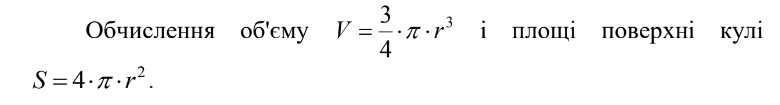


Рисунок 4 Завдання №3\_2

Завдання №5 VNS Practice Work – Task 4

* Варіант завдання - 1

Деталі завдання



Рисунок 5 завдання № 4

## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма № 1 VNS Practice Work - Task 1

* Блок-схема

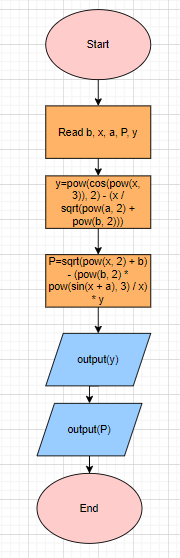


Рисунок 6 Блок-схема до програми №1

* Планований час на реалізацію-25 хвилин

Програма №2 VNS Practice Work - Task 2

* Блок-схема

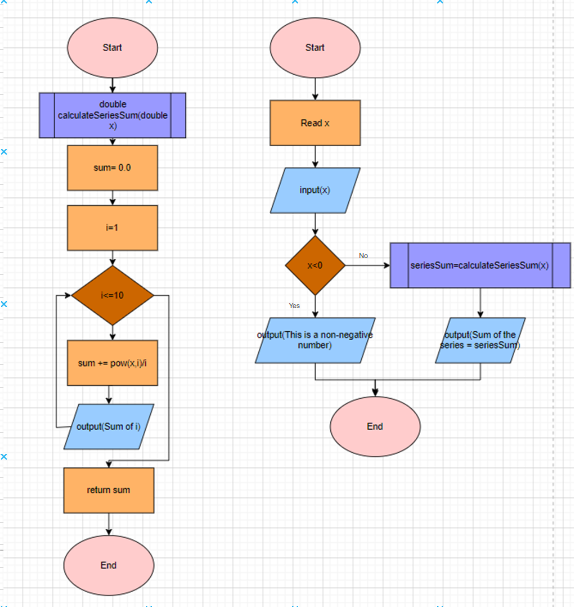


Рисунок 7 Блок-схема до програми №2

* Планований час на реалізацію-30 хвилин

Програма №3 VNS Practice Work - Task 3\_1

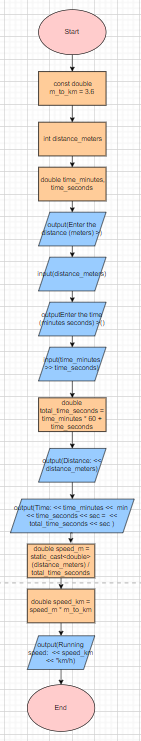
* Блок-схема
* 

Рисунок 8 Блок-схема до програми №3

* Планований час на реалізацію-30 хвилин

Програма №4 VNS Practice Work - Task 3\_2

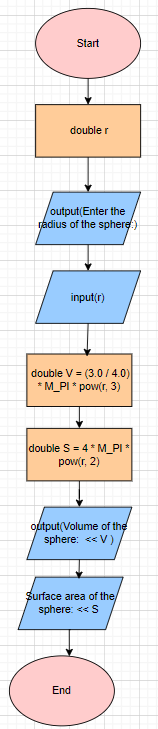
* Блок-схема
* 

Рисунок 9 Блок-схема до програми №4

* Планований час на реалізацію-15 хвилин

Програма №5 VNS Practice Work - Task 4

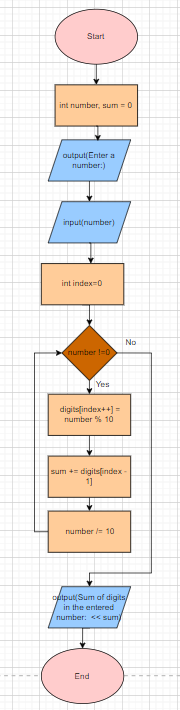
* Блок-схема
* 

Рисунок 10 Блок-схема до програми №5

* Планований час на реалізацію-30 хвилин

## **3. Код програм з посиланням на зовнішні ресурси:**

Завдання №1

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/920/files#diff-dd8ccda64c7ad8a55f3d652eb2ab8b63206034e5f3e6362f7c9ef979dbbc9c2e>

#include <iostream>

#include <cmath>

using namespace std;

int main (){

   double b = 15.5; // Оголошуємо змінну b зі значенням 15.5

    double x = -2.9; // Оголошуємо змінну x зі значенням -2.9

    double a = 1.5; // Оголошуємо змінну a зі значенням 1.5

    double P, y; // Оголошуємо змінні P та y

    // Обчислюємо значення змінної y за формулою

    y = pow(cos(pow(x, 3)), 2) - (x / sqrt(pow(a, 2) + pow(b, 2)));

    // Обчислюємо значення змінної P за формулою, використовуючи обчислене значення y

    P = sqrt(pow(x, 2) + b) - (pow(b, 2) \* pow(sin(x + a), 3) / x) \* y;

    cout<<"y= "<<y<<endl;//виводимо у

    cout<<"P= "<<P;//виводимо Р

    return 0;

}

Завдання №2

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/920/files#diff-78129dede684c0c3f7d7e7a0d4695b887e9bb0fd8908597053373fb036ae4d29>

#include <iostream>

#include <cmath>

using namespace std;// Підключення необхідних бібліотек для вводу/виводу та математичних операцій.

double calculateSeriesSum(double x) {// Функція для обчислення суми ряду відповідно до вхідного значення 'x'.

    double sum = 0.0; // Ініціалізація змінної 'sum' для збереження суми ряду.

    for (int i = 1; i <= 10; ++i) { // Цикл для обчислення суми ряду з десяти елементів від 1 до 10.

     sum += pow(x, i) / i; // Додавання до суми виразу x^i / i.

    cout<<"Sum of "<<i<<" = "<<sum<<endl;

}

    return sum; // Повернення обчисленої суми ряду.

}

int main() {

    double x;

    cout << "Enter the value of x: ";

    cin >> x; // Введення значення 'x' користувачем.

     // Перевірка на введення від'ємного числа

    if (x < 0) {

        cout << "This is a non-negative number." << endl;

    } else {

        double seriesSum = calculateSeriesSum(x);

        cout << "Sum of the series = " << seriesSum << endl;// Виведення обчисленої суми ряду.

    }

    return 0;

}

Завдання №3

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/920/files#diff-266806684f98edf0c705a5812b54578593ae540022fa3b71353ebd3deba6e66f>

#include <iostream>

#include <iomanip> // Для використання функції setprecision()

using namespace std;

int main() {

    const double m\_to\_km = 3.6; // Коефіцієнт для конвертації метрів за секунду в кілометри за годину

    int distance\_meters;

    double time\_minutes, time\_seconds;

    cout << "Enter the distance (meters) > ";

    cin >> distance\_meters;

    cout << "Enter the time (minutes seconds) > ";

    cin >> time\_minutes >> time\_seconds;

    double total\_time\_seconds = time\_minutes \* 60 + time\_seconds;  // Переведення часу в секунди

    cout << "Distance: " << distance\_meters << " m" << endl;

    cout << "Time: " << time\_minutes << " min " << time\_seconds << " sec = " << total\_time\_seconds << " sec" << endl;

    double speed\_m = static\_cast<double>(distance\_meters) / total\_time\_seconds; // Обчислення швидкості в метрах за секунду

    double speed\_km = speed\_m \* m\_to\_km; // Конвертація швидкості з метрів за секунду в кілометри за годину

    cout << fixed << setprecision(2); // Встановлення точності для виводу дробових чисел

    cout << "Running speed: " << speed\_km << " km/h" << endl;

    return 0;

}

Завдання №4

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/920/files#diff-06ed219456c61168ee05074957219584a01f9178a8196f19caa0d425e0325857>

#include <iostream>

#include <cmath>

using namespace std;

int main() {

    double r; // Оголошення змінної для радіусу кулі

    cout << "Enter the radius of the sphere: ";

    cin >> r;

    // Обчислення об'єму кулі за формулою V = (4/3) \* π \* r^3

    double V = (3.0 / 4.0) \* M\_PI \* pow(r, 3);

    // Обчислення площі поверхні кулі за формулою S = 4 \* π \* r^2

    double S = 4 \* M\_PI \* pow(r, 2);

    // Вивід обчисленого об'єму кулі на екран

    cout << "Volume of the sphere: " << V << endl;

    // Вивід обчисленої площі поверхні кулі на екран

    cout << "Surface area of the sphere: " << S << endl;

    return 0;

}

Завдання №5

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/920/files#diff-0c0ccb75a668d8ed560b0e6cedcc8f9d43a08f9079e1ac47a0f8e9a9cde3bf7c>

#include <iostream>

using namespace std;

int main() {

    int number, sum = 0;

    int digits[10]; // Масив для зберігання цифр введеного числа

    cout << "Enter a number: ";

    cin >> number;

    int index = 0;

    while (number != 0) {

        digits[index++] = number % 10; // Зберігаємо останню цифру числа в масиві

        sum += digits[index - 1]; // Додаємо останню цифру до суми

        number /= 10; // Видаляємо останню цифру числа

    }

    cout << "Sum of digits in the entered number: " << sum << endl;

    return 0;

}

## **4. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1



Рисунок 11 Результат програми №1

Час затрачений на виконання завдання – 20 хвилин

Завдання №2

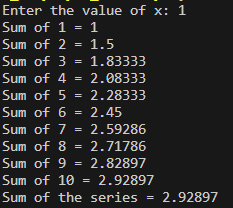


Рисунок 12 Результат програми №2

Час затрачений на виконання завдання – 30 хвилин

Завдання №3

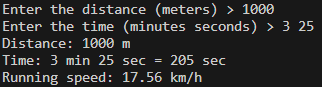


Рисунок 13 Результат програми №3

Час затрачений на виконання завдання – 30 хвилин

Завдання №4



Рисунок 14 Результат програми №4

Час затрачений на виконання завдання – 30 хвилин

Завдання №1



Рисунок 15 Результат програми №5

Час затрачений на виконання завдання – 30 хвилин

# **Висновки:**

Я підсумувала набуті знання вивчені цього навчального курсу виконанням розрахункової роботи.