Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № (замінити і вказати номери лабораторних з ВНС)**

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

***з розділу***: «Лінійні та розгалужені алгоритми. Умовні оператори. Константи, змінні»

***Виконав:***

студент групи ШІ-12

Бортник Тарас Петрович

# **Тема роботи:**

# **Вивчення лінійних та розгалужених алгоритмів, умовних операторів, робота зі змінними та константами, операції введення-виведення у мові програмування C++.**

# **Мета роботи:**

# **Ознайомлення з основами програмування на C++ та використання умов та операцій введення-виведення для розробки простих програм.**

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Лінійні алгоритми
* Тема №2: Умовні оператори
* Тема №3: Розгалужені алгоритми

1. Індивідуальний план опрацювання теорії:

* Тема №1: Лінійні алгоритми
  + Джерела Інформації
    - Відео.
    - <https://www.youtube.com/watch?v=RIU2y1EkS1U>
    - Стаття
    - http://cpp.dp.ua/ponyattya-algoritmu/
  + Що опрацьовано:
    - Вивчення та впровадження послідовних алгоритмів для вирішення завдань в програмуванні.
    - Освоєння основ лінійних алгоритмів: послідовних кроків у програмуванні для ефективного вирішення завдань.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.10.2023
  + Звершення опрацювання теми: 14.10.2023
* Тема №2: Умовні оператори.
  + Джерела Інформації:.
    - Відео.
    - <https://www.youtube.com/watch?v=r7Gg_B_FXDs>
    - Стаття.
    - <https://acode.com.ua/urok-67-operatory-umovnogo-rozgaluzhennya-if-else/>
    - <https://www.bestprog.net/uk/2017/08/02/conditional-jump-operator-if-2_ua/>
  + Що опрацьовано:
    - Вивчення умовних операторів розвинуло мою здатність створювати гнучкі програми, реагуючи на різні умови, що полегшило написання ефективного та адаптивного коду.
    - Вивчення умовних операторів розширило мої можливості у написанні програм, які інтелігентно реагують на різні умови, забезпечуючи гнучкість та точність у виконанні завдань.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.10.2023
  + Звершення опрацювання теми: 14.10.2023
* Тема №3: Розгалужені алгоритми
  + Джерела Інформації:
    - Стаття
    - <https://disted.edu.vn.ua/courses/learn/10348>
  + Що опрацьовано:
    - Опрацювання розгалужених алгоритмів навчило створювати гнучкі програми, які адаптують свою дію до різних умов, роблячи код більш ефективним та адаптивним.
    - Коментар 2
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.10.2023
  + Звершення опрацювання теми: 15.10.2023

# **Виконання роботи:**

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

Завдання №2 VNS lab 1 task 2

* Варіант завдання : 22
* Деталі завдання
* Цей код виконує обчислення виразу, використовуючи змінні типу float. Основна ідея полягає в тому, що спочатку обчислюється чисельник і знаменник, а потім результат обчислення виразу записується в змінну result і виводиться на екран.

Зміна типу даних на double дозволяє використовувати більше бітів для зберігання значень і може забезпечити більшу точність при обчисленнях.

* Важливі деталі для врахування в імплементації програми
* Важливо враховувати точність числових типів, особливо у випадку типу `float`, щоб уникнути втрат точності при роботі з великими або дуже малими значеннями. Користуйтеся функціями з бібліотеки `<cmath>` для виконання математичних операцій. Перевіряйте вирази уважно, щоб уникнути помилок та підвищити продуктивність програми.

Завдання №1 VNS lab 1 task 2

* Варіант завдання: 22
* Деталі завдання:
* Цей код використовує бібліотеки <iostream> та <cmath>. Він приймає введені користувачем числа m і n та виводить результати трьох виразів: добуток (n+1) \* (m+1), порівняння m і n після постфіксного інкременту m, та порівняння n і m після постфіксного інкременту n. Результати порівнянь виводяться у вигляді логічних значень (true або false). Програма використовує інкременти, що може впливати на значення змінних.
* Важливі деталі для врахування в імплементації програми
* Важливі деталі включають використання бібліотек <iostream> та <cmath>, введення користувача для m і n, префіксний інкремент та порівняння результатів у форматі логічних значень.

Завдання №3 Algotester lab1

* Варіант завдання: 1
* Деталі завдання
* Цей код отримує введення щодо початкового часу (hbegin, mbegin) та трьох інших часових відрізків (h1, m1, h2, m2, h3, m3). Використовується цикл for та умовні оператори для визначення можливості відняти часові відрізки від початкового часу. Результат, "YES" або "NO", виводиться в залежності від умов.
* Важливі деталі для врахування в імплементації програми
* Цей код отримує введення щодо часу початку та трьох інших часових відрізків, використовуючи цикл та умовні оператори для визначення можливості відняти відрізки від початкового часу. Результат "YES" виводиться, якщо це можливо, інакше виводиться "NO"

Завдання №4 Class practice

* Варіант завдання 1
* Деталі завдання :
* Цей код створює простий порадник щодо погоди. Користувач вводить поточні погодні умови, і програма виводить рекомендації щодо одягу та активностей, використовуючи оператори умови та перeключeння.
* Важливі деталі для врахування в імплементації програми:
* Програма враховує тип погоди, щоб надати рекомендації з одягу та активностей. Використовуються оператори умови (if-else) для визначення, чи потрібно носити куртку, та switch-case для різних типів погоди, щоб вказати рекомендоване взуття. Код також перевіряє правильність введення погоди.

Завдання №5 Self-practice

Варіант завдання:name of task (Депутатські гроші)

* Деталі завдання :
* Цей код призначений для обчислення мінімальної кількості банкнот та монет, які складають задану суму грошей (введену користувачем). Використовуються вартості різних банкнот та монет, і програма визначає, скільки кожного типу потрібно для утворення введеної суми. Результат виводиться у вигляді кількості використаних банкнот та монет.
* Важливі деталі для врахування в імплементації програми:
* Програма приймає від користувача суму грошей та визначає мінімальну кількість банкнот та монет для утворення цієї суми. Використовується масив вартостей грошових одиниць, цикли для перевірки кожного типу грошей та цикл while для обчислення кількості. Результат виводиться у вигляді кількості використаних банкнот та монет.

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

Програма №\_\_ Заголовок задачі

* Блок-схема
* Планований час на реалізацію
* Важливі деталі для врахування в імплементації

Програма №\_\_ Заголовок задачі

* Блок-схема
* Планований час на реалізацію
* Важливі деталі для врахування в імплементації

Програма №\_\_ Заголовок задачі

* Блок-схема
* Планований час на реалізацію
* Важливі деталі для врахування в імплементації

## **3. Конфігурація середовища до виконання завдань:**

Завдання №\_\_ Деталі по конфігурації середовища + скріншоти з підписами до скріншотів.

Підпис та № до блоку з скріншотами до конфігурації

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

Завдання №

Підпис та № до блоку з кодом програми

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

Завдання №\_\_ Деталі по виконанню і тестуванню програми

Підпис та № до блоку з виконанням та тестуванням програми

Час затрачений на виконання завдання

# **Висновки:**

Деталі по результатам виконання робіт та висновки згідно тем та завдань