

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



Звіт

про виконання

Лабораторних та практичних робіт № 3

з дисципліни: «Мови та парадигми програмування»

з розділу: «Ерік 3. Цикли. Вкладені цикли. Функції. Перевантаження функції. Рекурсія»

Виконала:

студентка групи ШІ-14 - Гуменюк Тетяна В'ячеславівна

Тема роботи:

Вивчення різновиду циклів, створення функцій, ознайомлення з поняттями перевантаження функції та рекурсії.

Мета роботи:

Створення функцій та їх використання, застосування вкладених циклів.

Теоретичні відомості:

- 1) Теоретичні відомості з переліком важливих тем:
 - Тема №1: Цикли
 - Тема №2: Вкладені цикли
 - Тема №3: Функції
 - Тема №4: Перевантаження функцій
 - Тема №5: Рекурсія
- 2) Індивідуальний план опрацювання теорії:
 - Тема №1: Цикли
 - Джерела Інформації:
 - <https://dystosvita.org.ua/mod/page/view.php?id=876>
 - <http://cpp.dp.ua/operatory-tsyklu/>
 - Що опрацьовано:
 - Цикл з параметром, передумовою та післяумовою
 - Статус: Ознайомлена
 - Початок опрацювання теми: 1 грудня
 - Звершення опрацювання теми: 1 грудня
 - Тема №2: Вкладені цикли
 - Джерела Інформації:
 - <https://www.youtube.com/watch?v=70CkVC2vVCQ>
 - Що опрацьовано:
 - Приклади вкладених циклів
 - Статус: Ознайомлена
 - Початок опрацювання теми: 1 грудня
 - Звершення опрацювання теми: 2 грудня
 - Тема №3: Функції
 - Джерела Інформації:
 - <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
 - Що опрацьовано:
 - Поняття функції
 - Вкладені функції
 - Статус: Ознайомлена
 - Початок опрацювання теми: 2 грудня
 - Звершення опрацювання теми: 3 грудня
 - Тема №4: Перевантаження функцій
 - Джерела Інформації:
 - <https://acode.com.ua/urok-108-perevantazhennya-funktsij/>
 - Що опрацьовано:
 - Поняття перевантаження функції
 - Статус: Ознайомлена
 - Початок опрацювання теми: 2 грудня

- Звершення опрацювання теми: 2 грудня
- Тема №5: Рекурсія
 - Джерела Інформації:
 - <https://www.bestprog.net/uk/2019/01/07/recursion-examples-of-tasks-solving-advantages-and-disadvantages-of-recursion-ua-2/>
 - https://www.youtube.com/watch?v=V7q9w_s0nns
 - Що опрацьовано: ▪ Поняття рекурсії та приклади реалізації функцій на її основі
 - Статус: Ознайомлена
 - Початок опрацювання теми: 2 грудня
 - Звершення опрацювання теми: 2 грудня

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

- Завдання №1: Lab# programming: VNS Lab 2 variant 24
- Завдання №2: Lab# programming: VNS Lab 3 variant 24
- Завдання №3: Lab# programming: VNS Lab 7 variant 24

2. Дизайн та планована оцінка часу виконання завдань:

Програма № 1: Lab# programming: VNS Lab 2 variant 24

- Деталі завдання:

24) Знайти суму 9 членів ряду, у якому

$$a_n = e^{-\sqrt{n}}$$

- Планований час на реалізацію – 1 година
- Важливі деталі для врахування в імплементації

1. При визначенні суми членів ряду варто використовувати рекурентну формулу для отримання наступного члена ряду.

Програма 2: Lab# programming: VNS Lab 3 variant 24

- Деталі завдання:

24	$y = \ln\left(\frac{1}{2 + 2x + x^2}\right)$	$-2 \leq x \leq -0,1$	40	$S = -(1+x)^2 + \frac{(1+x)^4}{2} + \dots + (-1)^n \frac{(1+x)^{2n}}{n}$
----	--	-----------------------	----	--

- Планований час на реалізацію – 1 година

Програма №3: Lab# programming: VNS Lab 7 variant 24

- Деталі завдання:

24. Написати функцію (або макровизначення), що знаходить довжину сторони за координатами його точок.. Написати функцію square, що обчислює площу трикутника, заданого координатами вершин. Написати функцію squaren с змінною кількістю параметрів, що визначає площу опуклого багатокутника, заданого координатами своїх вершин.

- Планований час на реалізацію – 3 години

3. Код програм:

- Завдання №1: Lab# programming: VNS Lab 2 variant 24

```
#include <stdio.h>
#include<math.h>
int main()
{
```

```

- int n=9;
- double a=exp(-1);
- double sum=a;
- for(int i=2; i<n; ++i)
- {
-     a = a*exp(-sqrt(n+1)+sqrt(n));
-     sum += a;
- }
- printf("sum = %lf", sum);
- }

```

- Завдання №2: Lab# programming: VNS Lab 3 variant 24

```

- #include <stdio.h>
- #include <math.h>
- #include <va_list>
-
- int main()
- {
-     int n=40;
-     int k=10;
-     double e=0.0001;
-     double a=-2;
-     double b=-0.1;
-     double dx=(b-a)/k;
-     double dS=1;
-     double x=a;
-     while(x<=b)
-     {
-         double a = -pow(1+x, 2);
-         double sum_n = a;
-         //for n
-         for(int i=2; i<n; ++i)
-         {
-             a = a*(-n*(1+x)/n+1);
-             sum_n += a;
-         }
-         a = -pow(1+x, 2);
-         double sum_e = a;
-         //for e
-         int i = 0;
-         while(fabs(a)>e)
-         {
-             i++;
-             a = a*(-n*(1+x)/n+1);
-             sum_e += a;
-         }
-         double y = log(1/(2+2*x+x*x));
-         printf("X=%f , SN=%f , SE=%f ,Y=%f ", x, sum_n, sum_e, y);
-         x+=dx;
-     }
- }

```

- Завдання №3: Lab# programming: VNS Lab 7 variant 24

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

- Завдання №1: Lab# programming: VNS Lab 2 task 1 variant 24

sum = 1.785389

- Час затрачений на виконання завдання: 1 година

- Завдання №2: Lab# programming: VNS Lab 3 task 2 variant 24
- Час затрачений на виконання завдання: 1 година
- Завдання №3: Lab# programming: VNS Lab 7 task 2 variant 24
- Час затрачений на виконання завдання: 3 години