

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



## **Звіт**

**про виконання лабораторних та практичних робіт блоку № 3**

**На тему:** «Цикли. Вкладені Цикли. Завершення виконання циклів. Функції.  
Простір імен. Перевантаження функцій. Функції з змінною кількістю  
параметрів (еліпсис). Рекурсія. Вбудовані функції.»

**З дисципліни:** «Основи програмування»

**до:**

Практичних Робіт до блоку № 3

**Виконав:**

Студент групи ІІІ-11

Голейчук Іван Миколайович

Львів 2024

**Тема роботи:** Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.

**Мета роботи:** Ознайомитися з основами циклів, дослідити концепцію функцій у C++, навчитися зображати функції в блок-схемах, розібратися зі способами завершення виконання циклів.

**Теоретичні відомості:**

Тема №1: Введення в Цикли та їх Види в C++

Тема №2: Управління Виконанням Циклів

Тема №3: Вкладені Цикли

Тема №4: Основи Функцій у C++

Тема №5: Перевантаження Функцій та Простір Імен

Тема №6: Розширені Можливості Функцій

Тема №7: Вбудовані Функції в C++: Індивідуальний план опрацювання теорії

## **Індивідуальний план опрацювання теорії:**

**Тема №1: Введення в Цикли та їх Види в C++**

**Джерела інформації:**

- Лекції Олександра Пшеничного;
- Практичні заняття;
- Використання штучного інтелекту (чат gpt);
- Youtube.

**Що опрацьовано:** Ознайомився з основними видами циклів у C++: for, while та do-while. Розібрався з їхніми відмінностями та прикладами використання для ітераційних завдань.

**Тема №2: Управління Виконанням Циклів**

**Джерела інформації:**

- Лекції Олександра Пшеничного;
- Практичні заняття;

-Використання штучного інтелекту (чат gpt);

-Youtube.

**Що опрацьовано:** Вивчив способи управління виконанням циклів, зокрема оператори break, continue, і return, а також їх застосування для контролю і виходу з циклів за певних умов.

### **Тема №3: Вкладені Цикли**

#### **Джерела інформації:**

-Лекції Олександра Пшеничного;

-Практичні заняття;

-Використання штучного інтелекту (чат gpt);

-Youtube.

**Що опрацьовано:** Розглянув вкладені цикли (цикли всередині інших циклів) і навчився їх використовувати, особливо для роботи з багатовимірними масивами чи повторюваних обчислень.

### **Тема №4: Основи Функцій у C++**

#### **Джерела інформації:**

-Лекції Олександра Пшеничного;

-Практичні заняття;

-Використання штучного інтелекту (чат gpt);

-Youtube.

**Що опрацьовано:** Вивчив основи функцій: оголошення, виклик, повернення значень і параметри функцій. Зрозумів, як функції структурують код і дозволяють повторно використовувати логіку.

### **Тема №5: Перевантаження Функцій та Простір Імен**

#### **Джерела інформації:**

-Лекції Олександра Пшеничного;

-Практичні заняття;

-Використання штучного інтелекту (чат gpt);

-Youtube.

**Що опрацьовано:** Розібрався з перевантаженням функцій, тобто можливістю створювати функції з однаковою назвою, але різними параметрами. Також опрацював простір імен (namespace), щоб організувати код і уникати конфліктів імен.

## **Тема №6: Розширені Можливості Функцій**

### **Джерела інформації:**

- Лекції Олександра Пшеничного;
- Практичні заняття;
- Використання штучного інтелекту (чат gpt);
- Youtube.

**Що опрацьовано:** Розглянув розширені можливості функцій у C++, такі як параметри за замовчуванням, передача за посиланням і значенням, а також використання функцій як аргументів інших функцій.

## **Тема №7: Вбудовані Функції в C++**

### **Джерела інформації:**

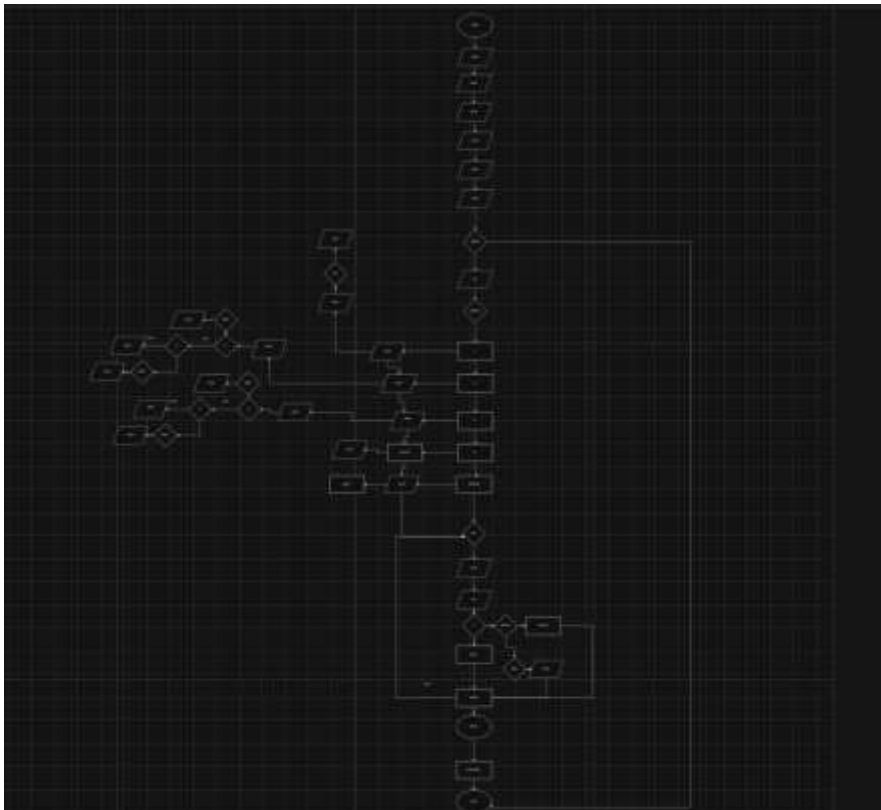
- Лекції Олександра Пшеничного;
- Практичні заняття;
- Використання штучного інтелекту (чат gpt);
- Youtube.

**Що опрацьовано:** Склав індивідуальний план для опрацювання теорії про вбудовані функції в C++, включаючи математичні функції, функції для роботи з рядками та функції введення/виведення, щоб зрозуміти їхнє практичне застосування.

## **Виконання роботи:**

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





### Task 3 - Lab# programming: VNS Lab 2

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6
7  |
8  double f(int n) {
9      double r = 1;
10     for (int i = 1; i <= n; ++i) {
11         r *= i;
12     }
13     return r;
14 }
15
16 int main() {
17     double suma = 0.0;
18
19     for (int n = 1; n <= 10; ++n) {
20         double suma = f(n) / pow(n, sqrt(n));
21     }
22
23     cout << suma << endl;
24
25     return 0;
26 }
27
28
29

```

## Task 4 - Lab# programming: VNS Lab 3

```
1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7  int main() {
8
9      double a = 0.1, b = 1.0;
10
11      int k = 10;
12
13      double D = (b - a) / k;
14
15
16      cout << "y = 2 * (cos^2(x) - 1);" << endl;
17
18
19      for (double x = a; x <= b; x += D) {
20
21          double y = 2 * (pow(cos(x), 2) - 1);
22
23          cout << "x = " << x << endl;
24          cout << "y = " << y << endl;
25      }
26
27      return 0;
28  }
```

## Task 5 - Lab# programming: VNS Lab 7



```

1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 double f(double x1, double y1, double x2, double y2) {
7     return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
8 }
9
10 bool belong(double x, double y, double x1, double y1, double x2, double y2, double x3, double y3) {
11     double abc = abs((x2 - y3) + x3 * (y3 - y1) + x1 * (y1 - y2)) / 2.0;
12     double abx = abs((x2 - y3) + x3 * (y3 - y1) + x * (y1 - y2)) / 2.0;
13     double acx = abs((x2 - y3) + x3 * (y3 - y1) + x * (y1 - y2)) / 2.0;
14     double bcx = abs((x2 - y3) + x3 * (y3 - y1) + x * (y1 - y2)) / 2.0;
15     return (abx + acx + bcx == abc);
16 }
17
18 bool p(double x, double y, double vertices[10], int numVertices) {
19     bool inside = false;
20     for (int i = 0, j = numVertices - 1; i < numVertices; j = i++) {
21         if ((vertices[i][0] > y) && (vertices[j][0] > y) && x < ((vertices[j][0] - vertices[i][0]) * (y - vertices[i][0]) / (vertices[j][0] - vertices[i][0]) + vertices[i][0])) {
22             inside = true;
23         }
24     }
25     return inside;
26 }
27
28 int main() {
29     double ab = {1, 1, 4, 1};
30     cout << "ab: " << ab << endl;
31
32     bool t = belong(2, 2, 1, 1, 4, 1, 2, 2);
33     cout << "belong(2, 2, 1, 1, 4, 1, 2, 2) << endl;
34
35     double w[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
36     bool p = p(2, 2, w, 10);
37     cout << "Does the point (2, 2) belong to a polygon? " << (p ? "Yes" : "No") << endl;
38
39     return 0;
40 }

```

## Task 6 - Practice# programming: Class Practice Task

```

1 //include libraries
2 #include <iostream>
3 #include <vector>
4 #include <string>
5
6 using namespace std;
7
8 //display menu
9 void d() {
10
11     cout << "Use library menu now" << endl;
12
13     cout << "Show all books" << endl;
14
15     cout << "Take a book" << endl;
16
17     cout << "Return the book" << endl;
18
19     cout << "Exit" << endl;
20
21     cout << "Select an option: " << endl;
22 }
23
24 int main() {
25
26     vector<string> b = { "book 1", "book 2", "book 3", "book 4" }; //books
27     vector<bool> a = { true, true, true, true }; //available
28
29     int q; //choice
30     bool w = false; //exit
31
32     while (true) {
33         d();
34         cin >> q;
35
36         switch (q) {
37
38             case 1: {
39
40                 cout << "List of all books:" << endl;
41                 for (int i = 0; i < b.size(); i++) {
42                     cout << i + 1 << " < " << b[i] << " (available) : " << (a[i]) << endl;
43                 }
44                 break;
45             } case 2: {
46
47                 int b1; //book's number
48
49                 cout << "Enter the number of the book you want to take: " << endl;
50                 cin >> b1;
51
52                 if (b1 > 0 && b1 <= b.size()) {
53                     if (a[b1 - 1]) {
54                         a[b1 - 1] = false;
55                         cout << "You took the book: " << b[b1 - 1] << endl;
56                     } else {
57                         cout << "This book is already taken!" << endl;
58                     }
59                 } else {
60                     cout << "Invalid book number!" << endl;
61                 }
62                 break;
63             } case 3: {
64
65                 int r1; //number of book
66
67                 cout << "Select the book you want to return : " << endl;
68                 cin >> r1;
69
70                 if (r1 > 0 && r1 <= b.size()) {
71                     if (!a[r1 - 1]) {
72                         a[r1 - 1] = true;
73                         cout << "You returned the book: " << b[r1 - 1] << endl;
74                     } else {
75                         cout << "This book is now available!" << endl;
76                     }
77                 } else {
78                     cout << "Invalid book number!" << endl;
79                 }
80                 break;
81             } case 4: {
82
83                 w = true;
84
85                 cout << "Thank you for using the library!" << endl;
86                 break;
87             } default: {
88
89                 cout << "Wrong choice. Try again." << endl;
90                 goto start;
91             }
92         }
93
94         char c; //continue
95
96         do {
97
98             cout << "Do you want to perform another operation? (y/n): " << endl;
99             cin >> c;
100
101             if (c == 'y' || c == 'Y') {
102                 goto start;
103             } else if (c == 'n' || c == 'N') {
104                 w = true;
105             } else cout << "Wrong choice. ";
106         } while (c != 'y' && c != 'Y' && c != 'n' && c != 'N');
107
108         start:
109
110         continue;
111     }
112
113     return 0;
114 }

```

## Task 7 - Practice# programming: Self Practice Task

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      ios::sync_with_stdio(false);
7      cin.tie(nullptr);
8
9      int n, k;
10     cin >> n >> k;
11     vector<int> p(n);
12
13     for (int i = 0; i < n; i++) {
14         cin >> p[i];
15     }
16
17     int s = 0;
18
19
20     for (int i = 0; i < n - 1; i++) {
21         if (p[i] + 1 != p[i + 1]) {
22             s++;
23         }
24     }
25     s++;
26
27
28     if (s <= k) {
29         cout << "Yes\n";
30     }
31     else {
32         cout << "No\n";
33     }
34
35     return 0;
36 }
37

```

**Робота у команді:**



Це останній епік в цій команді, і я дуже вдячний хлопцям за затрачений час, допомогу та роботу разом.

**Висновок:** Опрацювавши ці теми, я отримав ґрунтовне розуміння циклів і функцій у C++. Це дозволило мені освоїти різні типи циклів та способи управління їх виконанням, що є необхідним для побудови ефективних ітераційних структур. Знання вкладених циклів розширило мої можливості роботи з багатовимірними структурами, а функції допомогли краще структурувати код, зробивши його більш організованим і зручним для повторного використання.

Додатково я зрозумів, як працює перевантаження функцій і простір імен, що дозволяє уникати конфліктів між іменами та організовувати код. Завдяки вивченню розширених можливостей функцій я можу налаштовувати їхню поведінку за потреби. Індивідуальний план опрацювання вбудованих функцій допоможе мені використовувати їх у різних завданнях та підвищити ефективність свого коду.

Загалом, ці теми сформували основу для розуміння важливих концепцій у програмуванні, які будуть корисними для подальшого розвитку в алгоритмах та структурі даних.