

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет "Львівська політехніка"
Кафедра САПР



Звіт з лабораторної роботи №6
з дисципліни "Об'єктно-орієнтоване
програмування" "Стандартна бібліотека
шаблонів_Smart_pointer" Варіант - 11

Виконав:
ст. гр. КН-108
Пагута В.О.
Прийняв:

Мета роботи:

Набути умінь та навички розробки та опису програм з використанням динамічних структур, шаблонів класів та контейнерів. Навчитись застосувати розумні вказівники бібліотеки STL.

Завдання №1:

11. З текстового файлу зчитати дійсні числа у вектор. З вектора видалити всі непарні додатні числа. Утворені елементи перенести у список у зворотному порядку. Вивести на екран утворений список.

Код:

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <vector>
```

```
#include <list>
```

```
#include
```

```
<windows.h> using
```

```
namespace std;
```

```
void print(vector<float>& array);
```

```
void print(vector<float>& array)
```

```
{
```

```
if (array.empty())
```

```
{
```

```
cout << "Вектор порожній";
```

```
return;
```

```
}
```

```
vector<float>::iterator iter;
```

```
for (iter = array.begin(); iter != array.end(); iter++)  
{  
    cout << *iter << " ";  
}  
}
```

```
void print(list<float>& l);
```

```
void print(list<float>& l)  
{  
    if (l.empty())  
    {  
        cout << "Список порожній";  
        return;  
    }  
}
```

```
list<float>::iterator iter;
```

```
for (iter = l.begin(); iter != l.end(); iter++)  
{  
    cout << *iter << " ";  
}  
cout << endl;
```

```
}
```

```
int main()
```

```
{
```

```
SetConsoleCP(1251);
```

```
SetConsoleOutputCP(125
```

```
1);
```

```
string ipath =
```

```
"file_input.txt"; ifstream fin;
```

```
fin.open(ipath,
```

```
ios::in); float n;
```

```
vector<float> arr;
```

```
for (int i = 0; !fin.eof(); i++)
```

```
{
```

```
fin >> n;
```

```
arr.push_back(n
```

```
);
```

```
}
```

```
cout << "Зчитані з файлу у вектор елементи: ";
```

```
print(arr);
```

```
cout <<
```

```
endl;
```

```
fin.close();
```

```
int a;
```

```
for (int i = 0; i < arr.size(); i++)
```

```
{  
if(arr[i]>0)  
{  
a = arr[i];  
  
if(a%2 != 0)  
{  
arr.erase(arr.begin()+i);  
}  
}  
}
```

```
cout << "Видалені непарні додатні числа: ";  
print(arr);  
cout <<  
endl;
```

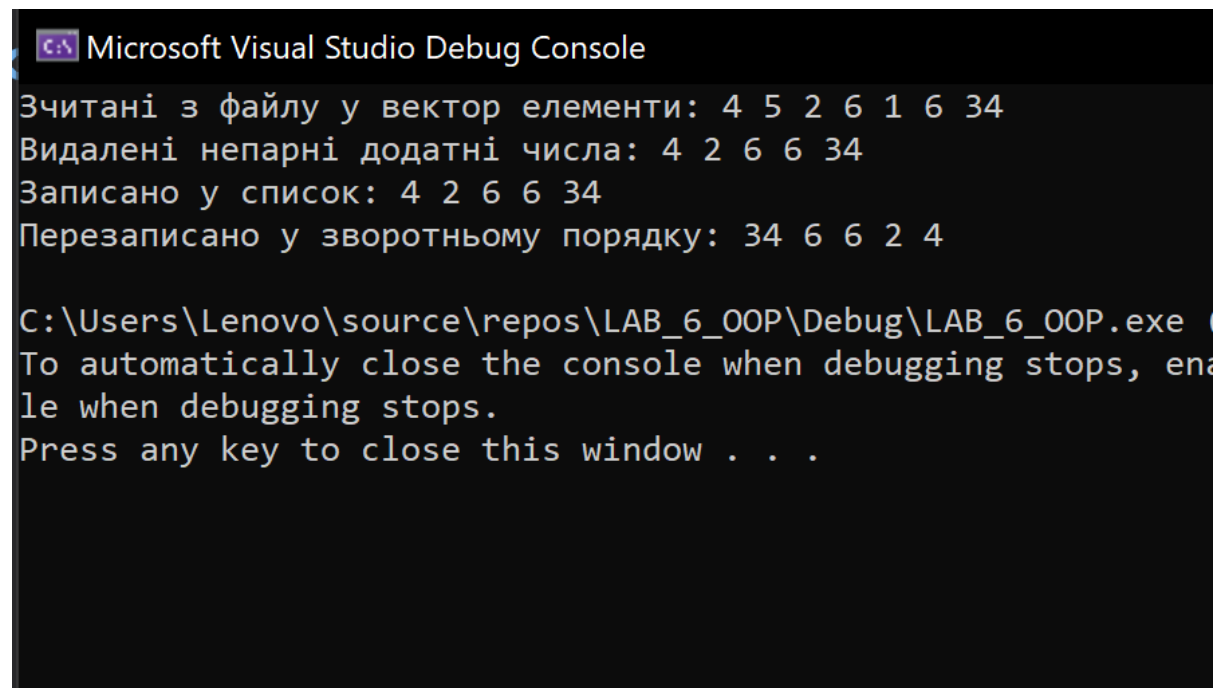
```
list<float> list;
```

```
for (int i = 0; i < arr.size(); i++)  
{  
list.push_back(arr[i]);  
}
```

```
cout << "Записано у список: ";  
print(list);
```

```
list.reverse();  
cout << "Перезаписано у зворотньому порядку: ";  
print(list);  
  
}
```

Виконання:



The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```
Зчитані з файлу у вектор елементи: 4 5 2 6 1 6 34  
Видалені непарні додатні числа: 4 2 6 6 34  
Записано у список: 4 2 6 6 34  
Перезаписано у зворотньому порядку: 34 6 6 2 4  
  
C:\Users\Lenovo\source\repos\LAB_6_OOP\Debug\LAB_6_OOP.exe  
To automatically close the console when debugging stops, enable  
the option in the console window menu.  
Press any key to close this window . . .
```

Завдання №2:

10. Змінити відмітку введення про кількість книг у функції.

11. Тест допомоги для деякої програми організований як лінійний список. Кожен компонент тексту допомоги містить термін (слово) і текст, що містить пояснення до цього терміну. Кількість рядків тексту, що відносяться до одного терміну, коливається від однієї до п'яти.

Скласти програму, що забезпечує:

- початкове формування тексту допомоги;
- виведення тексту допомоги;
- виведення тексту, що пояснює заданий термін.

Код:

```
#include <iostream>
#include <string>
#include
<windows.h>
#include <list>
#include <iterator>
using namespace
std;

class helpText
{
public:
    string term;
    string
    meaning;

    helpText(string trm,string mean)
    {
        this->term = trm;
        this->meaning = mean;
    }
    helpText()
    {
        term = "undefined";
        meaning =
        "undefined";
    }
}
```

```

~helpText()
{

}

void printTerm();
void
printMeaning();
};

void helpText::printTerm()
{
    cout << this->term << " - ";
}

void helpText::printMeaning()
{
    cout << this->meaning << "\n";
}

/*void newElement(list<helpText> lst, string trm, string mean)
{

```



```
    helpText* a = new  
    helpText(trm,mean);  
    lst.push_back(*a);  
}*/
```

```
void print(list<helpText> lst);
```

```
void print(list<helpText> lst)  
{  
    if (lst.empty())  
    {  
        cout << "Список порожній";  
        return;  
    }
```

```
    for (auto& item : lst) {  
        item.printTerm();  
        item.printMeaning(  
        ); cout << endl;  
    }
```

```
    cout << endl;  
}
```

```

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251
); int i = 1;
    int choice = 1;
    list<helpText>
    lst;

    helpText OOP;
    OOP.term = "Об'єктно-орієнтовне програмування";
    OOP.meaning = "одна з парадигм програмування, яка розглядає
програму як множину «об'єктів», що взаємодіють між собою.[4]
Оснoву ООП складають чотири основні концепції: інкапсуляція,
успадкування, поліморфізм та абстракція[5] Однією з переваг
ООП є краща модульність програмного забезпечення.";

    helpText JS;
    JS.term = "ДжаваСкрипт";
    JS.meaning = "динамічна, об'єктно-орієнтована[4] прототипна
мова програмування. Реалізація стандарту ECMAScript.
Найчастіше
використовується для створення сценаріїв вебсторінок, що надає
можливість на боці клієнта (пристрої кінцевого користувача)
взаємодіяти з користувачем, керувати браузером, асинхронно
обмінюватися даними з сервером, змінювати структуру та зовнішній
вигляд вебсторінки.";

    helpText Bandera;
    Bandera.term = "Степан Андрійович Бандера";
    Bandera.meaning = "український політичний діяч, один із чільних
ідеологів і теоретиків українського націоналістичного руху XX
століття[1], після розколу Організації українських націоналістів —
голова Проводу ОУН-Б[2].";

```

```
lst.push_back(OOP);  
lst.push_back(JS);  
lst.push_back(Bander  
a);
```

```
while (choice != 0)  
{  
    cout << "Виберіть дію:" << endl;  
    cout << "1.Вибрати термін" <<  
endl;  
    cout << "2.Вивести весь текст допомоги" << endl;  
    cout << "3.Створити новий термін" << endl;  
    cin >>  
    choice;  
    switch  
(choice)  
{  
    case 1:  
        cout << "Виберіть термін:" << endl;  
        for (helpText& item :  
            lst) { cout << i << ". ";  
            item.printTerm();  
            cout <<  
endl; i++;  
        }  
        cin >>  
        choice; i = 1;  
  
        for (helpText& item : lst)  
{
```

```
        if (i == choice)
        {
            item.printTerm();
            item.printMeaning(
                );
        }
        i++;
    }
    i = 1;
    break;
case 2: print(lst); break;
```

case 3:

```
    string trm;
    string
    mean;
    cout << "Введіть назву терміну: ";
    getline(cin, trm); getline(cin, trm);
    cout << "Введіть значення
    терміну"; getline(cin, mean);
    helpText* a = new helpText(trm,
    mean); lst.push_back(*a);
    break;
}
}
}
```

Виконання:

```
C:\Users\Lenovo\source\repos\LAB_6_OOP_2\Debug\LAB_6_OOP_2.exe
Виберіть дію:
1.Вибрати термін
2.Вивести весь текст допомоги
3.Створити новий термін
3
Введіть назву терміну: Морозиво
Введіть значення терміну молочний продукт
Виберіть дію:
1.Вибрати термін
2.Вивести весь текст допомоги
3.Створити новий термін
2
Об'єктно-орієнтовне програмування - одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою.[4] Основу ООП складають чотири основні концепції: інкапсуляція, успадкування, поліморфізм та абстракція[5] Однією з переваг ООП є краща модульність програмного забезпечення.

ДжаваСкрипт - динамічна, об'єктно-орієнтована[4] прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки.

Степан Андрійович Бандера - український політичний діяч, один із чільних ідеологів і теоретиків українського націоналістичного руху XX століття[1], після розколу Організації українських націоналістів – голова Проводу ОУН-Б[2].

Морозиво - молочний продукт
```

Контрольні запитання:

1. Що таке динамічні структури? Опишіть їх.

Динамічні структури за визначенням характеризуються відсутністю фізичної суміжності елементів структури в пам'яті, непостійністю і непередбачуваністю розміру (кількість елементів) структури в процесі її обробки.

2. Як додати новий елемент у список? Приклад.

```
List<int>
```

```
lst; int l =
```

```
5;
```

```
lst.push_back(l);
```

3. Як видалити елемент з черги? Приклад.

1) `Lst.pop_back(l);`

2) `Lst.pop_front(l);`

4. Як додати елемент у стек? Приклад.

```
stack<int>
```

```
stck; Int l = 3;
```

```
Stck.push(l);
```

5. Що таке дек? Приклад.

Дек - двохсторонній список.

```
deque<char> myDeque(5);
```

6. Як включити новий елемент у бінарне дерево? Приклад.

```
node* root =
```

```
NULL; int p;
```

```
int key;
```

```
cout << "Enter amount of
```

```
keys: "; cin >> p;
```

```
while (p != 0)
```

```
{
```

```
cin >> key;
```

```
root = insert(root,
```

```
key); p--;
```

```
}
```

7. Що таке контейнер? Види та їх призначення.

Контейнери — це об'єкти, що зберігають інші елементи і реалізують механізми доступу до них. Прикладами контейнерів є вектори і списки. Кожний контейнер описується

шаблонним класом, у якому реалізуються механізми доступу і функція для обробки елементів, що містяться у контейнері. Кожний контейнер має свої ітератори для

перебору елементів і функції для їх обробки. Наприклад, клас, що описує роботу із вектором, має відповідний ітератор для прямого доступу до елементів вектору, а також функції для вставки і видалення тощо.

8. Для чого призначений вектор? Які методи він підтримує?

Вектор — це абстрактна модель, яка імітує динамічний масив.

<code>vector::clear</code>	Видаляє всі елементи вектора
<code>vector::insert</code>	Вставка елементів у вектор
<code>vector::erase</code>	Видалення вказаних елементів вектора (один або декілька)
<code>vector::push_back</code>	Вставка елемента в кінець вектора
<code>vector::pop_back</code>	Видалити останній елемент вектора
<code>vector::resize</code>	Змінює розмір вектора на задану величину
<code>vector::swap</code>	Обміняти вміст двох векторів

9. Наведіть два випадки застосування `typename`?

Для ініціалізації шаблонної функції.

10. Як використовується слово `export` стосовно шаблонів?

Ключове слово `export` може передувати оголошенню `template`. Воно дозволяє використовувати шаблон з іншого файлу, повторюючи лише його оголошення, а не усе визначення.

11,12. Що таке розумний вказівник?

Це абстрактний тип даних, який імітує вказівник з допоміжними можливостями, такими як автоматичне керування пам'яттю або перевірку виходу за межі виділеної пам'яті.

Висновок:

Я набув уміння та навички розробки та опису програм з використанням динамічних структур, шаблонів класів та контейнерів, а також навчився застосовувати розумні вказівники бібліотеки STL.