

Лабораторна робота №11

Тема: Дослідження контейнерних класів бібліотеки STL.

Мета: дослідити контейнерні класи vector та list бібліотеки STL, набути навичок їх використання.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. З допомогою контейнера std::vector створити вектор значень типу char, в який записати український алфавіт (малими літерами).
2. Усі голосні букви у векторі замінити прописними (за допомогою циклу).
3. Вивести вміст вектора на екран.

Код програми:

```
#include <iostream>

#include <vector>

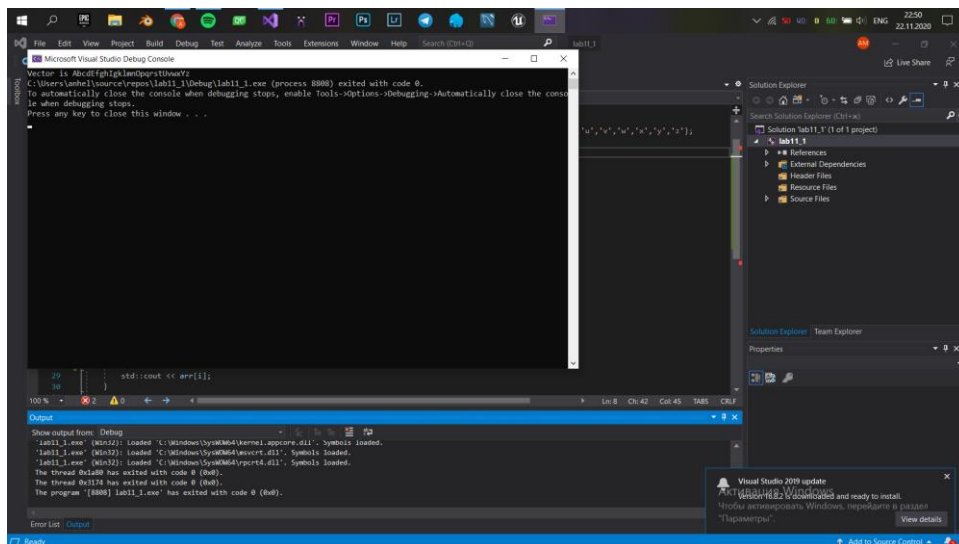
int main()
{
    std::vector<char> arr = {'a','b','c','d','e','f','g','h','i',
        'g','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
    std::cout << "Vector is ";
    for( int i = 0; i < arr.size (); i++ ) {
        switch( arr[i] ) {
            case 'a':
                arr[i] = 'A';
                break;
            case 'e':
                arr[i] = 'E';
                break;
            case 'i':
                arr[i] = 'I';
                break;
```

```

case'o':
arr[i] = 'O';
break;
case'u':
arr[i] = 'U';
break;
case 'y':
arr[i] = 'Y';
break;
}
std::cout << arr[i];
}
}

```

Результат виконання програми:



- З допомогою контейнера list створити список значень типу char, в який записати по буквах своє прізвище. Вивести вміст списку на екран.
- Відсортувати список (від А до Я). У консоль вивести повідомлення: «Список сортується!».
- Вивести вміст списку на екран.

```

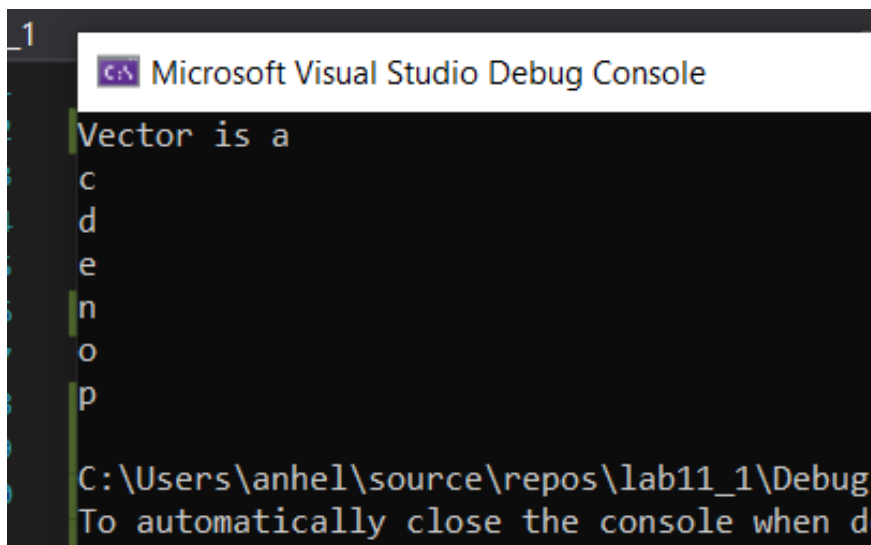
#include <iostream>

#include <list>

int main()
{
    std::list<char> arr = {'c','d','n','e','o','p','a'};
    std::cout << "Vector is ";
    arr.sort ();
    for( int n : arr ) { std::cout << static_cast<char>(n) << '\n'; }

}

```



```

1
Microsoft Visual Studio Debug Console
Vector is a
c
d
e
n
o
p
C:\Users\anhel\source\repos\lab11_1\Debug\
To automatically close the console when de

```

7. Створити вектор об'єктів класу, створеного згідно індивідуального завдання №1 в лабораторній роботі №2. Заповнити вектор десятьма об'єктами.

8. Здійснити вивід значень об'єктів за допомогою індексу вектора.

9. Здійснити вивід значень об'єктів за допомогою ітераторів.

```

#include<iostream>

```

```

#include <vector>

```

```

template <typename T>

```

```

class Airplane

```

```

{
private:
    template<typename T>
    class Node // шаблонний клас нод однозв'язний список
    {
    public:
        Node* pNext;
        T info;
        Node(T info = T(), Node* pNext = nullptr)
        {
            this->info = info;
            this->pNext = pNext;
        }
    };
    Node<T> *m_model;
    int m_power;

public:
    Airplane();
    Airplane(T* m_model, int m_power);
    Airplane(const Airplane& obj);
    void setModel(const T* m_model, int lenght); //додає масив в кінець
    T* getModel();
    void setPower(int); //присвоює потужність
    int getPower(); //виводить потужність
    void print(); //виводить всю інформацію
    void input(); // ввід об'єкта
    void pop_front (); // видаляє голову
    void pushBack (T); // додає елемент в кінець
    T getModel (int index); // виводить одну модель
    ~Airplane();

```

```
};  
  
template<typename T>  
Airplane<T>::Airplane ():m_power(0), m_model(nullptr){}
```

```
template<typename T>  
Airplane<T>::Airplane (T *model, int power){  
    setModel(model,power);  
}
```

```
template<typename T>  
Airplane<T>::Airplane (const Airplane &obj) { m_model = obj.m_model; m_power = obj.m_power; }
```

```
template<typename T>  
void Airplane<T>::setModel (const T *model, int lenght)  
{  
    for( int i = 0; i < lenght; i++ ) {  
        pushBack (model[i]);  
    }  
}
```

```
template<typename T>  
T *Airplane<T>::getModel ()  
{  
    return *m_model.info;  
}
```

```
template<typename T>  
void Airplane<T>::setPower (int a)  
{
```

```
    m_power = a;
}
```

```
template<typename T>
int Airplane<T>::getPower ()
{
    return m_power;
}
```

```
template<typename T>
void Airplane<T>::print ()
{
    Node<T> *curr = this->m_model;
    std::cout << "\nAll elements:";
    while( curr != nullptr ) {
        std::cout << curr->info << "\t";
        curr = curr->pNext;
    }
    std::cout << std::endl;
    std::cout << "Power is" << m_power << std::endl;
}
```

```
template<typename T>
void Airplane<T>::input ()
{
    while (m_power)
        pop_front ();
    int power;
    std::cout << "Enter number of models:";
    std::cin >> power;
```

```

int counter = 1;

T curr;

while( counter <= power ) {

    std::cout << "Enter element";

    std::cin >> curr;

    pushBack(curr);

    counter++;

}
}

```

```

template<typename T>
void Airplane<T>::pop_front ()
{
    Node<T> *temp = m_model;
    m_model = m_model->pNext;
    delete temp;
    m_power--;
}

```

```

template<typename T>
void Airplane<T>::pushBack (T element)
{
    if( m_model== nullptr ) {
        m_model = new Node<T>(element);

    }

    else {

        Node<T> *curr = this->m_model;
        while( curr->pNext != nullptr ) {

```

```

        curr = curr->pNext;
    }
    curr->pNext = new Node<T>(element);

}
m_power++;
}

```

```

template<typename T>
T Airplane<T>::getModel (int index)
{
    int counter = 0;
    Node<T> *curr = this->m_model;
    while( curr != nullptr ) {
        if( counter == index ) {
            return curr->info;
        }
        curr = curr->pNext;
        counter++;
    }
}

```

```

template<typename T>
Airplane<T>::~Airplane ()
{
    while( m_model != nullptr ) {
        pop_front();
    }
}

```



```

int main()
{
    char b[] = {'a','b','c'};
    char c[] = {'c','l'};
    std::vector<Airplane<char>> a;
    a.resize (10);
    a[8].setModel (b, 3);
    a[9].setModel (c,2);
    for( int i = 0; i < a.size (); i++ ) {
        a[i].print ();
    }

    std::cout << "Nine is ";
    a[9].print();
}

```

Результат:

```

1 Microsoft Visual Studio Debug Console
2
3 All elements:
4 Power is 0
5
6 All elements:
7 Power is 0
8
9 All elements:
10 Power is 0
11
12 All elements:
13 Power is 0
14
15 All elements:
16 Power is 0
17
18 All elements:
19 Power is 0
20
21 All elements:
22 Power is 0
23
24 All elements:
25 Power is 0
26
27 All elements:
28 Power is 0
29
30 All elements:
31 Power is 3
32
33 All elements:
34 Power is 2
35
36 All elements:
37 Power is 2
38
39
40

```

10. * З допомогою контейнера list створити список студентів групи. В контейнер поміщати об'єкти класу, в якому представлені такі атрибути: прізвище. ім'я, по батькові, вік, номер телефону, а також реалізовані сетери даних атрибутів, і перевизначена операція виводу, яка буде виводити значення атрибутів на екран.

Код програми:

```
#include <iostream>

#include <string>

#include<list>

class Student

{

private:

    std::string m_familyName;

    std::string m_name;

    std::string m_fatherName;

    int m_age;

    std::string m_phoneNum;

public:

    Student () :m_familyName ("Unknown"), m_name ("Unknown"), m_fatherName ("Unknown"),

m_age (0), m_phoneNum ("default"){ }

    Student (std::string familyName, std::string name, std::string fatherName, int age, std::string

phoneNum)

        :m_familyName (familyName), m_name (name), m_fatherName (fatherName), m_age (age),

m_phoneNum (phoneNum){ }

    void setFamilyName(std::string);

    void setName (std::string a) { m_name = a; }

    void setFatherName (std::string a) { m_fatherName = a; }

    void setAge (int i) { m_age = i; }

    void setPhoneNum (std::string a) { m_phoneNum = a; }

    friend std::ostream& operator<<(std::ostream &out, Student &st);

};
```

```

void Student::setFamilyName (std::string a)
{
    m_familyName = a;
}

int main()
{
    std::string s = "Makarchuk";
    Student a("makarchuk","Angelina", "Victor", 17,"000000000");
    Student b;
    std::list<Student> arr;
    arr.push_back (a);

    arr.push_back (b);
    for( Student n : arr) { std::cout << n << '\n'; }
}

std::ostream& operator<<(std::ostream &out, Student &st)
{
    out << "Family name :" << st.m_familyName << std::endl;
    out << "Name: " << st.m_name << std::endl;
    out << "Father name: " << st.m_fatherName << std::endl;
    out << "Age: " << st.m_age << std::endl;
    out << "Phone number:" << st.m_phoneNum << std::endl;

    return out;
}

```

Результат:

```
Microsoft Visual Studio Debug Console
11_4
25 Family name :makarchuk
26 Name: Angelina
26 Father name: Victor
27 Age: 17
28 Phone number:000000000
29
30 Family name :Unknown
31 Name: Unknown
32 Father name: Unknown
33 Age: 0
34 Phone number:default
35
36
C:\Users\anbel\source\repos\11_4\De
```

Висновок: дослідити контейнерні класи vector та list бібліотеки STL, набути навичок їх використання.