***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***

**Завдання 1.** З допомогою контейнера std::vector створити вектор значень типу char, в який записати український алфавіт (малими літерами).

**Завдання 2.** Усі голосні букви у векторі замінити прописними (за допомогою циклу).

**Завдання 3.** Вивести вміст вектора на екран.

**Завдання 4.** З допомогою контейнера list створити список значень типу char, в який записати по буквах своє прізвище. Вивести вміст списку на екран.

**Завдання 5.** Відсортувати список (від А до Я). У консоль вивести повідомлення: «Список сортується!».

**Завдання 6.** Вивести вміст списку на екран.

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

**Завдання 8.** Здійснити вивід значень об’єктів за допомогою індексу вектора.

**Завдання 9.** Здійснити вивід значень об’єктів за допомогою ітераторів.

**Завдання 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 <QCoreApplication>

#include <iostream>

#include <string>

using namespace std;

class Furniture

{

char \*Room = new char;

int Weight;

public:

Furniture(char\* room, int weight)

{

SetRoom(room);

Weight = weight;

}

Furniture():Furniture("Unknown", 0) {}

Furniture(const Furniture& furn)

{

//cout << "copy" << endl;

Weight = furn.Weight;

SetRoom(furn.Room);

}

~Furniture() {/\*delete Room;\*/ }

void SetRoom(const char\* room)

{

delete Room;

Room = new char[40];

for(int i = 0; i < 40; i++)

Room[i] = room[i];

}

char\* GetRoom(){return Room;}

void SetWeight(int val) { Weight = val; }

int GetWeight() { return Weight; }

void Print() { cout << "Mass: " << Weight << ", room: " << Room << endl;

}

void Input() {

cout << "Write mass: ";

cin >> Weight;

cout << "Room: ";

cin >> Room;

}

};

class Student

{

string name,surname, fatherName, phoneNumber;

int age;

public:

Student(string name, string surname, string fatherName, string phoneNumber, int age)

{

this->name = name;

this->surname = surname;

this->fatherName = fatherName;

this->phoneNumber = phoneNumber;

this->age = age;

}

string GetName(){return name;}

string GetSurname(){return surname;}

string GetFatherName(){return fatherName;}

string GetPhoneNumber(){return phoneNumber;}

int GetAge(){return age;}

void Print()

{

cout << "Student - " << name << " " << surname << " " << fatherName << endl << "Phone number - " << phoneNumber << endl <<

"Age - " << age << endl;

}

};

int main(int argc, char \*argv[])

{

QCoreApplication a(argc, argv);

//--------------1---------------------

setlocale(0, "RUS");

vector<wchar\_t> alphabet; //= {L'а', L'б', L'в', L'г', L'u\1169', L'д'};

for(int i = 0; i < 33; i++)

{

alphabet.push\_back((wchar\_t)((int)(L'а')+i));

wcout << alphabet[i] << " ";

}

cout << endl;

//--------------2---------------------

vector<char> v;

for(auto i: "abcdefgigklnoprstuvwxyz")

v.push\_back(i);

for(unsigned int i = 0 ;i < v.size(); i++)

if(QString("aioyeu").contains(v[i]))

v[i] = (char)(v[i]-32);

for(auto i : v)

cout << i << " " ;

cout << endl;

//--------------3---------------------

vector<int> vect = {1,2,4,5,6,7,8};

for(unsigned int i = 0; i < vect.size(); i++)

cout << vect.at(i) << " ";

cout << endl;

//--------------4---------------------

list<char> surname;

QString n = "Pelishko";

for(auto i:n.toStdString())

surname.push\_back(i);

for(auto i:n.toStdString())

cout << i;

cout << endl;

//--------------5---------------------

vector<char> alphabet1;

for(int i = 25; i >= 0; i--)

alphabet1.push\_back((char)(i+97));

sort(alphabet1.begin(), alphabet1.end());

cout << "Array is sorting" << endl;

//--------------6---------------------

for(unsigned int i = 0; i < alphabet1.size(); i++)

cout << alphabet1[i] << " ";

cout << endl;

//--------------7---------------------

vector<Furniture> vect1;

for(int i = 0; i < 10; i++) vect1.push\_back(Furniture());

//--------------8---------------------

for(unsigned int i = 0; i < vect1.size(); i++)

vect1[i].Print();

cout << endl;

//--------------9---------------------

vector<Furniture>::iterator iter;

for(iter = vect1.begin(); iter != vect1.end(); iter++)

iter->Print();

cout << endl;

//--------------10--------------------

list<Student> listStud;

list<Student>::iterator iter1;

for(int i = 0; i < 2; i++)

listStud.push\_back(Student("q", "w", "e", "a", 1234));

for(iter1 = listStud.begin(); iter1 != listStud.end(); iter1++)

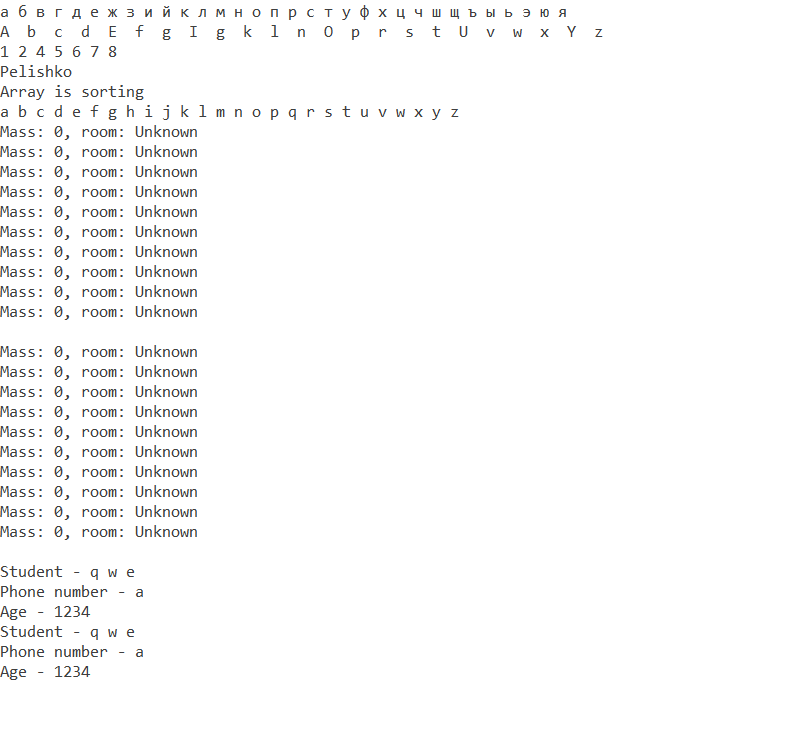
iter1->Print();

cout << endl;

return a.exec();

}

Вивід програми:



**Висновок:** Дослідив контейнерні класи vector та list бібліотеки STL, набув навичок їх використання.