

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №11**

**з дисципліни «Об’єктно орієнтованне програмування»**

**на тему: «Послідовні контейнери list, vector»**

Виконав:

студент гр. ПЗ2011

Кудик Сергій

Прийняла:

Демидович Інна Миколаївна

Дніпро, 2022

**Тема**: «Послідовні контейнери list, vector».

**Мета**: «Навчитися використовувати вектори та списки».

1. **Завдання**

**Постановка завдання**

Написати програму, що демонструє роботу з list і vector. Для демонастрації роботи використовувати операції додавання, видалення, перегляду та алгоритми бібліотеки STL. Під час роботи з контейнерами використовувати ітератори.

**Індивідуальне завдання**

Клас та функцію обрати за власними вподобаннями для довільної предметної області.

1. **Текст програми vector**

#include <vector>

#include <list>

#include <iostream>

#include <iterator>

#include <ctime>

using namespace std;

void showVector(vector<int>& vector)

{

if (!vector.empty())

{

copy(vector.begin(), vector.end(), ostream\_iterator<int>(cout, ", "));

cout << endl;

}

}

void randomFillingVector(vector<int>& vector, int mode)

{

srand(time(0));

if (mode == 1)

{

int rsize = 5 + rand() % 10;

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

{

vector.insert(vector.end(), 1 + rand() % 10);

}

}

if (mode == 2)

{

vector.push\_back(1 + rand() % 10);

}

}

void compare(vector<int>& first, vector<int>& second)

{

if (first == second)

{

cout << "Vector == Vector2" << endl;

}

else

{

cout << "Vector != Vector2" << endl;

}

}

void vectorInfo(vector<int>& vector)

{

cout << endl << "------- Vector info ----------";

cout << endl << "Count elements in vector: " << vector.size();

cout << endl << "First elements in vector: " << vector.front();

cout << endl << "Last elements in vector: " << vector.back();

cout << endl << "------------------------------" << endl;

}

void deleteElement(vector<int>& vector)

{

srand(time(0));

int Element = 0 + rand() % vector.size();

if (!vector.empty())

{

vector.erase(vector.begin() + Element);

cout << endl << "Delete element in " << Element << " position from vector.";

}

}

void randomFillingList(list<int>& list)

{

srand(time(NULL));

for (int i = 0; i < 5; i++)

{

list.push\_back(1 + rand() % 20);

}

}

void showList(list<int>& list, int number)

{

cout << "\nList " << number << ": ";

copy(list.begin(), list.end(), ostream\_iterator<int>(cout, ", "));

cout << endl;

}

int main()

{

//Робота з вектором

vector<int> Vector;

cout << "Vector: ";

randomFillingVector(Vector, 1);

showVector(Vector);

cout << endl << "Copy vector: ";

vector<int> Vector2(Vector);

showVector(Vector2);

cout << endl << "Compare vectors: ";

compare(Vector, Vector2);

vectorInfo(Vector);

randomFillingVector(Vector, 2);

cout << endl << "Vector: ";

showVector(Vector);

compare(Vector, Vector2);

Vector2.clear();

deleteElement(Vector);

cout << endl << "Vector: ";

showVector(Vector);

Vector.clear();

cout << endl;

//Робота з лістом

list<int> List;

list<int> List2;

cout << "Lists: ";

randomFillingList(List);

showList(List, 1);

List.sort();

cout << endl << "Sort list:";

showList(List, 1);

cout << endl << "Unique list: ";

List.unique();

showList(List, 1);

cout << endl << "Second List: ";

randomFillingList(List2);

showList(List2, 2);

List2.sort();

List.merge(List2);

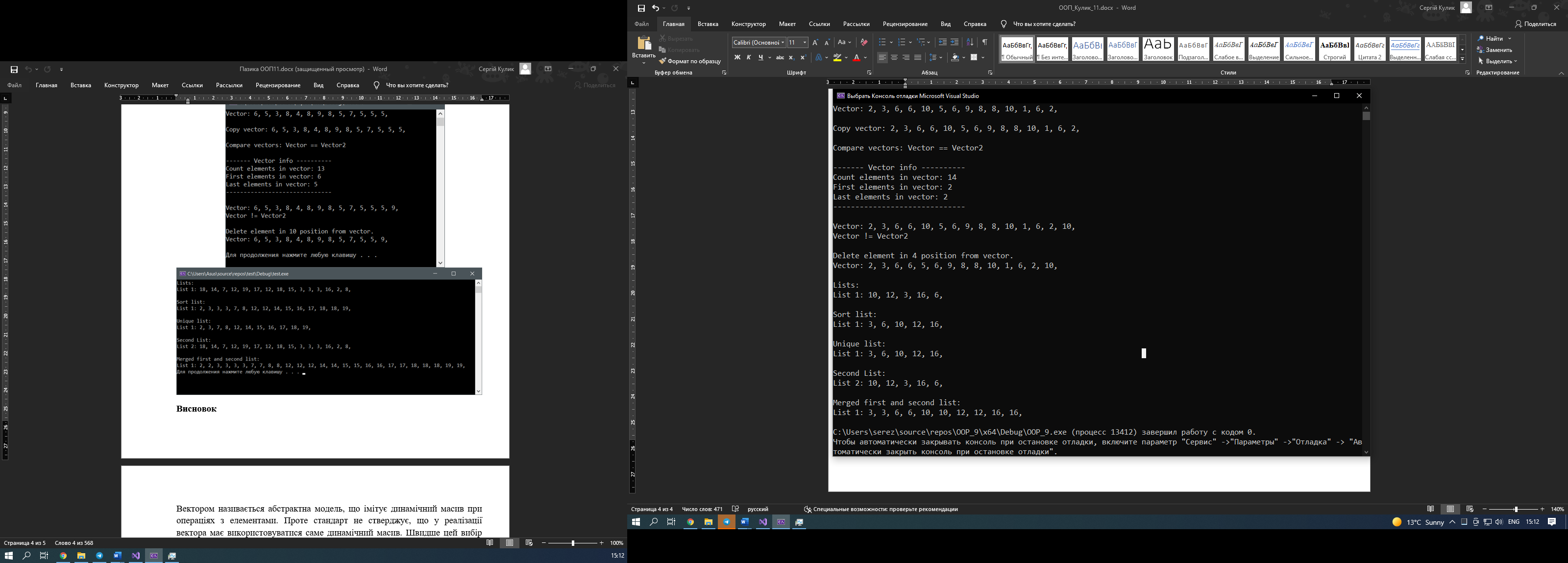
cout << endl << "Merged first and second list: ";

showList(List, 1);

return 0;

}

1. **Результати роботи програми**



1. **Висновок**

Під час виконання лабораторної роботи ознайомився з вектором і списком. Вектор являєтьтся динамічним масивом з розширеним функціоналом, а list реалізовує двухзв’язний список. Ці типи даних є шаблонними і входять до бібліотеки STL.