Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 11

з навчальної дисципліни

“Базові методології та технології програмування”

Завдання на розроблення програмного забезпечення

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студенти академічної групи КН-22

Кохан І.О.

Кіхоть М.В.

Ковальов М.В.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Собінов О.Г.

Кропивницький – 2023

**Лабораторна робота №11**

**Мета роботи:** полягає у набутті ґрунтовних вмінь і практичних навичок командної роботи реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам’яттю та бінарними файловими потоками.

**Завдання:** створити електронний довідник кодів операторів мобільного зв'язку України(назва оператора мобільного зв'язку, код)

За вибором користувача програмне забезпечення забезпечує :  
 1)*Пошук* запису в е-довіднику за кодом або назвою оператора;

2)*Виведення* всього е-довідника на екран або запис у текстовий файл;

3)*Додавання* нового запису в е- довідник;

4)*Вилучення* заданого запису з е-довідника;

5)*Завершення* роботи програми зі збереженням даних у файл.

Дані довідника автоматично завантажуються з файлу під час запуску програми.

**Варіант 7**

**Лістинг 1.1**#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

using namespace std;

class Operator {

public:

Operator(string name, string code) : name\_(name), code\_(code) {}

string name() const { return name\_; }

string code() const { return code\_; }

private:

string name\_;

string code\_;

};

class Phonebook {

public:

void loadFromFile(const string& filename) {

ifstream infile(filename);

if (!infile) {

cerr << "Failed to open file: " << filename << endl;

return;

}

string name, code;

while (infile >> name >> code) {

operators\_.push\_back(Operator(name, code));

}

}

void saveToFile(const string& filename) {

ofstream outfile(filename);

if (!outfile) {

cerr << "Failed to open file: " << filename << endl;

return;

}

for (const auto& op : operators\_) {

outfile<< op.name() << " " << op.code() << endl;

}

}

void searchByCode(const string& code) const {

auto it = find\_if(operators\_.begin(), operators\_.end(), [&](const Operator& op) {

return op.code() == code;

});

if (it != operators\_.end()) {

cout << "Operator: " << it->name() << ", Code: " << it->code() << endl;

} else {

cout << "No operator found with code: " << code << endl;

}

}

void searchByName(const string& name) const {

auto it = find\_if(operators\_.begin(), operators\_.end(), [&](const Operator& op) {

return op.name() == name;

});

if (it != operators\_.end()) {

cout << "Operator: " << it->name() << ", Code: " << it->code() << endl;

} else {

cout << "No operator found with name: " << name << endl;

}

}

void addOperator(const string& name, const string& code) {

operators\_.push\_back(Operator(name, code));

}

void removeOperator(const string& name, const string& code) {

operators\_.erase(remove\_if(operators\_.begin(), operators\_.end(), [&](const Operator& op) {

return op.name() == name && op.code() ==code;

}), operators\_.end());

}

void displayAllOperators() const {

for (const auto& op : operators\_) {

cout << "Operator: " << op.name() << ", Code: " << op.code() << endl;

}

}

private:

vector<Operator> operators\_;

};

int main() {

string filename = "phonebook.txt";

Phonebook phonebook;

phonebook.loadFromFile(filename);

while (true) {

cout << "Select an option:" << endl;

cout << "1. Search by code" << endl;

cout << "2. Search by name" << endl;

cout << "3. Display all operators" << endl;

cout << "4. Add new operator" << endl;

cout << "5. Remove operator" << endl;

cout << "6. Save and exit" << endl;

int choice;

cin >> choice;

switch (choice) {

case 1: {

string code;

cout << "Enter code: ";

cin >> code;

phonebook.searchByCode(code);

break;

}

case 2: {

string name;

cout << "Enter name: ";

cin >> name;

phonebook.searchByName(name);

break;

}

case 3: {

phonebook.displayAllOperators();

break;

}

case 4: {

string name,code;

cout << "Enter name: ";

cin >> name;

cout << "Enter code: ";

cin >> code;

phonebook.addOperator(name, code);

cout << "Operator added" << endl;

break;

}

case 5: {

string name, code;

cout << "Enter name: ";

cin >> name;

cout << "Enter code: ";

cin >> code;

phonebook.removeOperator(name, code);

cout << "Operator removed" << endl;

break;

}

case 6: {

phonebook.saveToFile(filename);

cout << "Exiting program..." << endl;

return 0;

}

default:

cout << "Invalid choice. Please try again." << endl;

}

}

return 0;

}

Даний лістинг програми є консольною програмою для роботи з телефонним довідником.

Користувач може виконувати наступні операції:

1. Пошук оператора за кодом;
2. Пошук оператора за ім'ям;
3. Відображення всіх операторів;
4. Додавання нового оператора;
5. Видалення оператора;
6. Збереження телефонного довідника та вихід з програми.

Програма використовує класи Operator та Phonebook для зберігання даних про операторів та роботи з телефонним довідником відповідно.

Основна функція main() містить цикл, який пропонує користувачу вибрати опцію та виконує відповідну функцію з *Phonebook*.

**Висновок**

Під час виконання лабораторної роботи №11, ми працювали як один колектив.

Збиралися на конференції, щоб вирішити проблеми які в нас виникали. Також

Допомагали один одному якщо в нас виникали якісь питання. З допомогою цієї лабораторної роботи ми трохи погрузилися до нового світу “командної роботи в програмуванні”. Це один невеликий крок до майбутнього в якому ми будемо постійно працювати в колективі та командно вирішувати проблеми.