Міністерство освіти і науки України

Тернопільський національний технічний університет імені І. Пулюя

Звіт

про виконання комплексної лабораторної роботи

з дисципліни :

Програмування

на тему: ” Створення програм із застосуванням бібліотек

та модулів користувача”

Тернопіль 2021

**Мета роботи:** навчитися розробляти алгоритми і програмне забезпечення на мові Сі++, застосовуючи функціонально-модульний підхід.

**Завдання для виконання роботи:**

**Завдання** - створити програму опрацювання **бази даних**, застосувавши її розбиття на окремі модулі та підпрограми. Інформація у базі даних повинна бути записана у вигляді бінарного файлу структур.

Програма повинна забезпечувати такі ***основні*** та ***додаткові*** функції опрацювання бази даних:

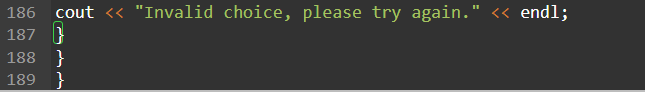
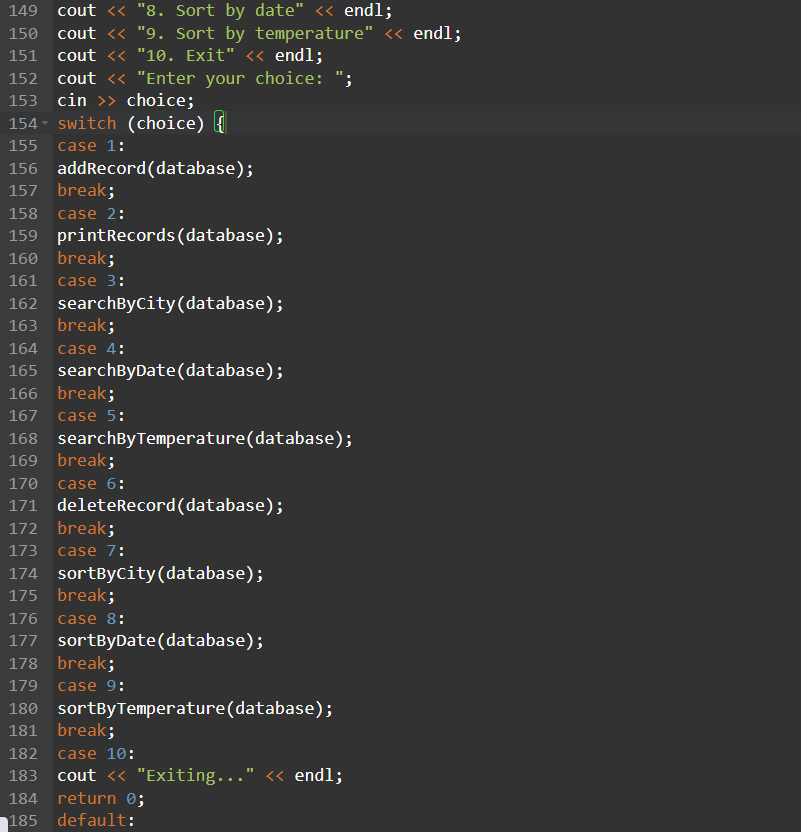
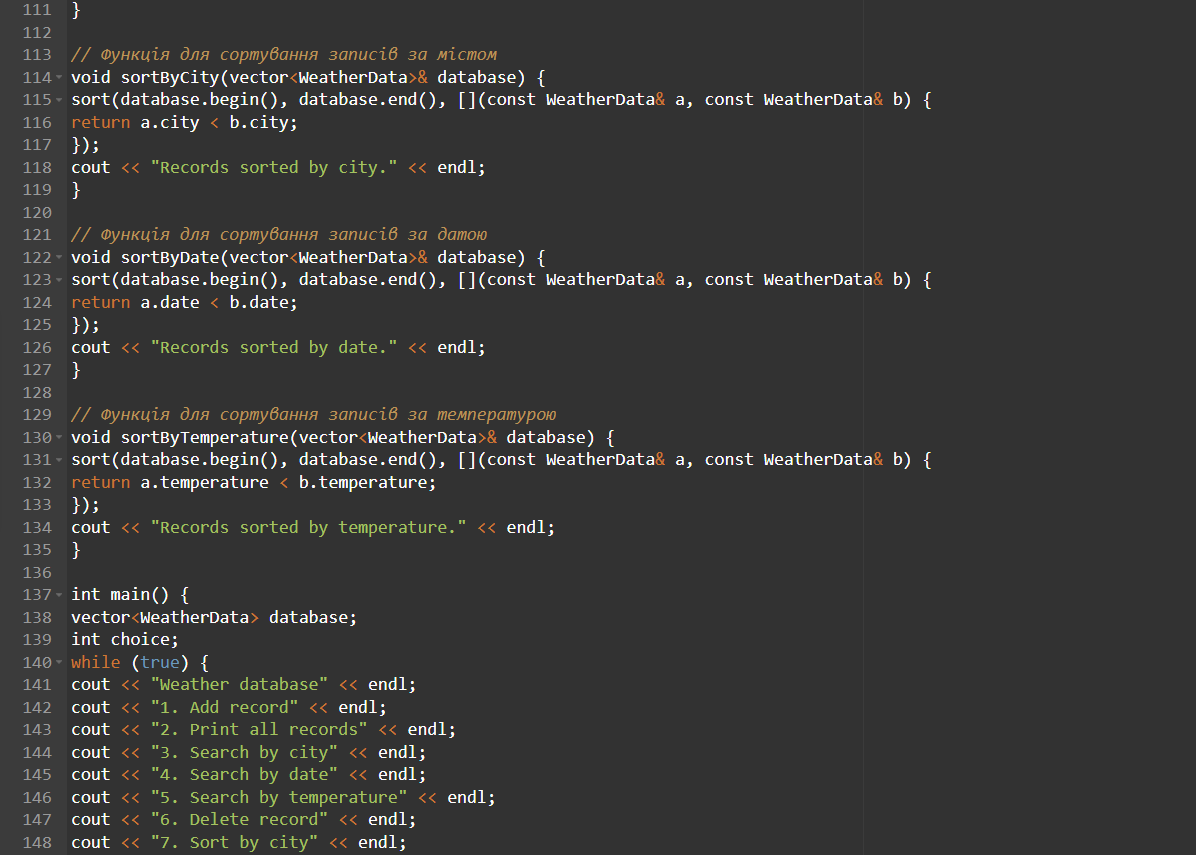
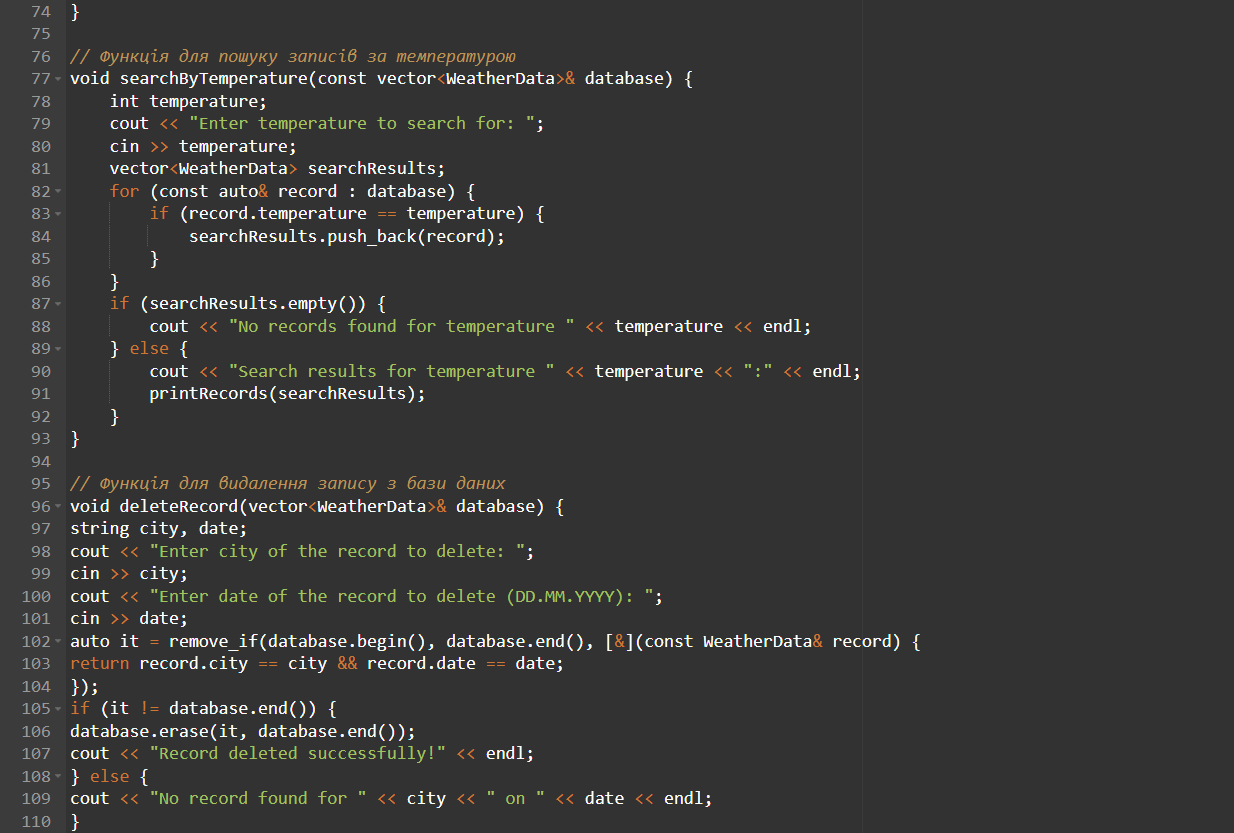
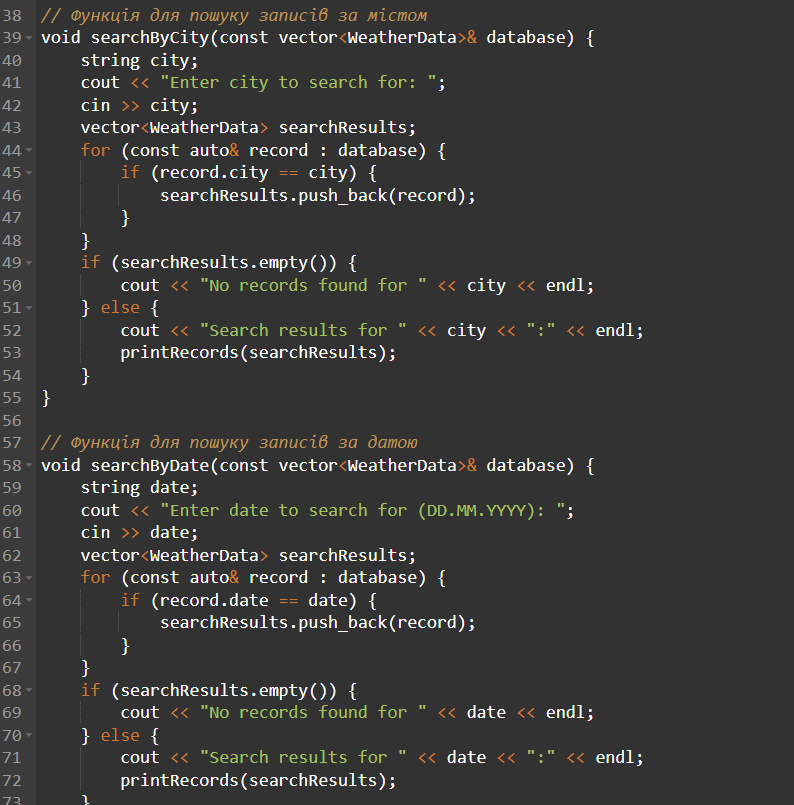
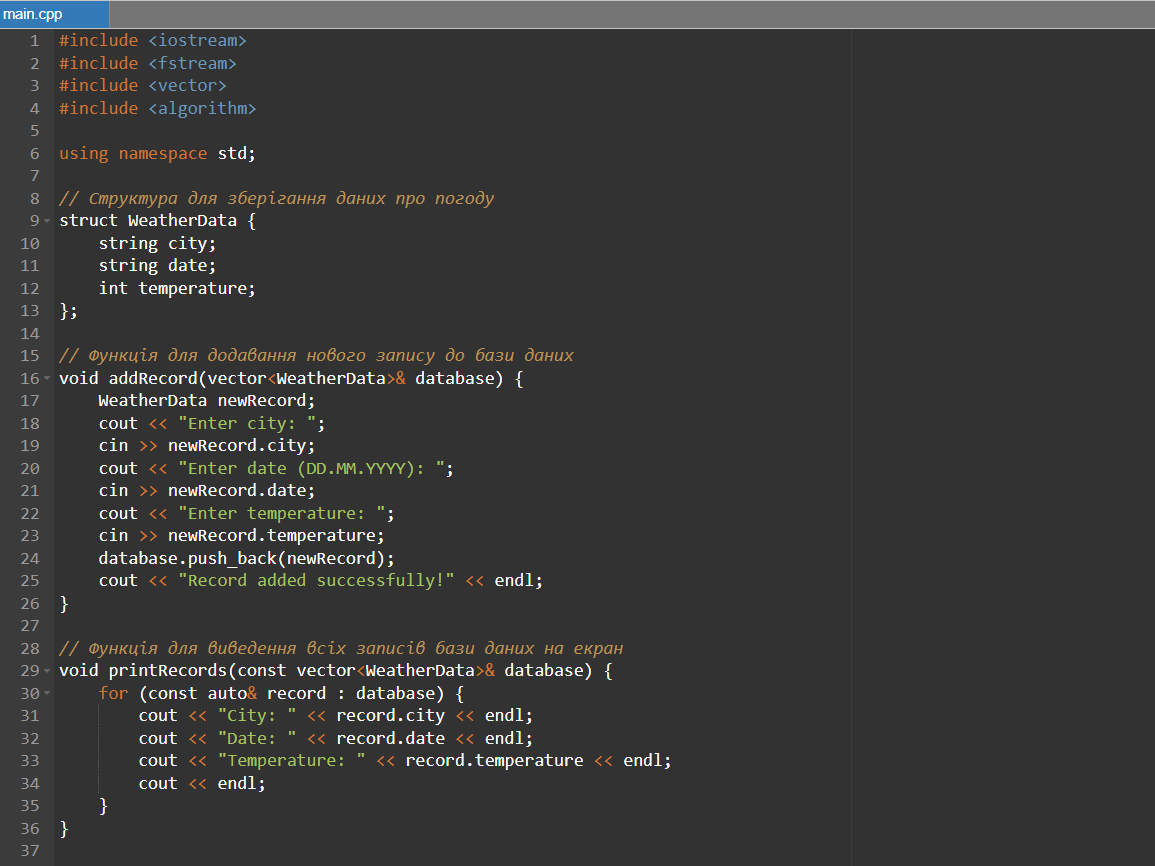
***Основні***

* Створення файла бази даних із довільним іменем, що вводиться з клавіатури
* Наповнення бази даних інформацією з клавіатури
* Видрук інформації про елемент бази даних на дисплей
* Пошук інформації за вказаним критерієм ( не менше 3 різних критеріїв пошуку)
* Доповнення інформації у базу даних

***Додаткові***

* Сортування інформації у базі даних (або створення відповідних індексних файлів для сортування за різними показниками)
* Зміна існуючої інформації про певний елемент бази даних
* Видалення вибраного елемента бази даних
* Пошук інформації за складними критеріями
* Обчислення певних узагальнених (сумарних) характеристик, що потребують розрахунку

**Програма до лабараторної роботи:**

****

**Код до програми:**

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

using namespace std;

// Структура для зберігання даних про погоду

struct WeatherData {

string city;

string date;

int temperature;

};

// Функція для додавання нового запису до бази даних

void addRecord(vector<WeatherData>& database) {

WeatherData newRecord;

cout << "Enter city: ";

cin >> newRecord.city;

cout << "Enter date (DD.MM.YYYY): ";

cin >> newRecord.date;

cout << "Enter temperature: ";

cin >> newRecord.temperature;

database.push\_back(newRecord);

cout << "Record added successfully!" << endl;

}

// Функція для виведення всіх записів бази даних на екран

void printRecords(const vector<WeatherData>& database) {

for (const auto& record : database) {

cout << "City: " << record.city << endl;

cout << "Date: " << record.date << endl;

cout << "Temperature: " << record.temperature << endl;

cout << endl;

}

}

// Функція для пошуку записів за містом

void searchByCity(const vector<WeatherData>& database) {

string city;

cout << "Enter city to search for: ";

cin >> city;

vector<WeatherData> searchResults;

for (const auto& record : database) {

if (record.city == city) {

searchResults.push\_back(record);

}

}

if (searchResults.empty()) {

cout << "No records found for " << city << endl;

} else {

cout << "Search results for " << city << ":" << endl;

printRecords(searchResults);

}

}

// Функція для пошуку записів за датою

void searchByDate(const vector<WeatherData>& database) {

string date;

cout << "Enter date to search for (DD.MM.YYYY): ";

cin >> date;

vector<WeatherData> searchResults;

for (const auto& record : database) {

if (record.date == date) {

searchResults.push\_back(record);

}

}

if (searchResults.empty()) {

cout << "No records found for " << date << endl;

} else {

cout << "Search results for " << date << ":" << endl;

printRecords(searchResults);

}

}

// Функція для пошуку записів за температурою

void searchByTemperature(const vector<WeatherData>& database) {

int temperature;

cout << "Enter temperature to search for: ";

cin >> temperature;

vector<WeatherData> searchResults;

for (const auto& record : database) {

if (record.temperature == temperature) {

searchResults.push\_back(record);

}

}

if (searchResults.empty()) {

cout << "No records found for temperature " << temperature << endl;

} else {

cout << "Search results for temperature " << temperature << ":" << endl;

printRecords(searchResults);

}

}

// Функція для видалення запису з бази даних

void deleteRecord(vector<WeatherData>& database) {

string city, date;

cout << "Enter city of the record to delete: ";

cin >> city;

cout << "Enter date of the record to delete (DD.MM.YYYY): ";

cin >> date;

auto it = remove\_if(database.begin(), database.end(), [&](const WeatherData& record) {

return record.city == city && record.date == date;

});

if (it != database.end()) {

database.erase(it, database.end());

cout << "Record deleted successfully!" << endl;

} else {

cout << "No record found for " << city << " on " << date << endl;

}

}

// Функція для сортування записів за містом

void sortByCity(vector<WeatherData>& database) {

sort(database.begin(), database.end(), [](const WeatherData& a, const WeatherData& b) {

return a.city < b.city;

});

cout << "Records sorted by city." << endl;

}

// Функція для сортування записів за датою

void sortByDate(vector<WeatherData>& database) {

sort(database.begin(), database.end(), [](const WeatherData& a, const WeatherData& b) {

return a.date < b.date;

});

cout << "Records sorted by date." << endl;

}

// Функція для сортування записів за температурою

void sortByTemperature(vector<WeatherData>& database) {

sort(database.begin(), database.end(), [](const WeatherData& a, const WeatherData& b) {

return a.temperature < b.temperature;

});

cout << "Records sorted by temperature." << endl;

}

int main() {

vector<WeatherData> database;

int choice;

while (true) {

cout << "Weather database" << endl;

cout << "1. Add record" << endl;

cout << "2. Print all records" << endl;

cout << "3. Search by city" << endl;

cout << "4. Search by date" << endl;

cout << "5. Search by temperature" << endl;

cout << "6. Delete record" << endl;

cout << "7. Sort by city" << endl;

cout << "8. Sort by date" << endl;

cout << "9. Sort by temperature" << endl;

cout << "10. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

addRecord(database);

break;

case 2:

printRecords(database);

break;

case 3:

searchByCity(database);

break;

case 4:

searchByDate(database);

break;

case 5:

searchByTemperature(database);

break;

case 6:

deleteRecord(database);

break;

case 7:

sortByCity(database);

break;

case 8:

sortByDate(database);

break;

case 9:

sortByTemperature(database);

break;

case 10:

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

return 0;

default:

cout << "Invalid choice, please try again." << endl;

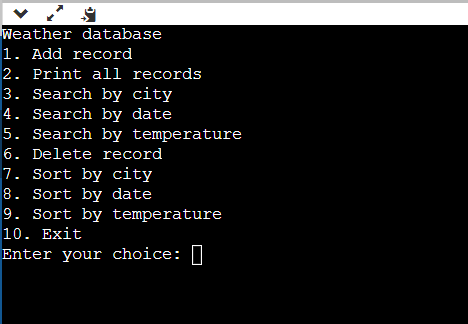
}

}

}

**Результат:**

Обравши будь-який код, можна перейти в те, чи інше меню.



**Висновок:** навчився розробляти алгоритми і програмне забезпечення на мові Сі++, застосовуючи функціонально-модульний підхід