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”

“РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ ОБРОБЛЕННЯ ДИНАМІЧНИХ СТРУКТУР ДАНИХ ТА БІНАРНИХ ФАЙЛІВ”

ВИКОНАВ

студенти академічної групи КН-22

Кирило Бездольний

Пащенко Данило

Олексій Щиченко

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

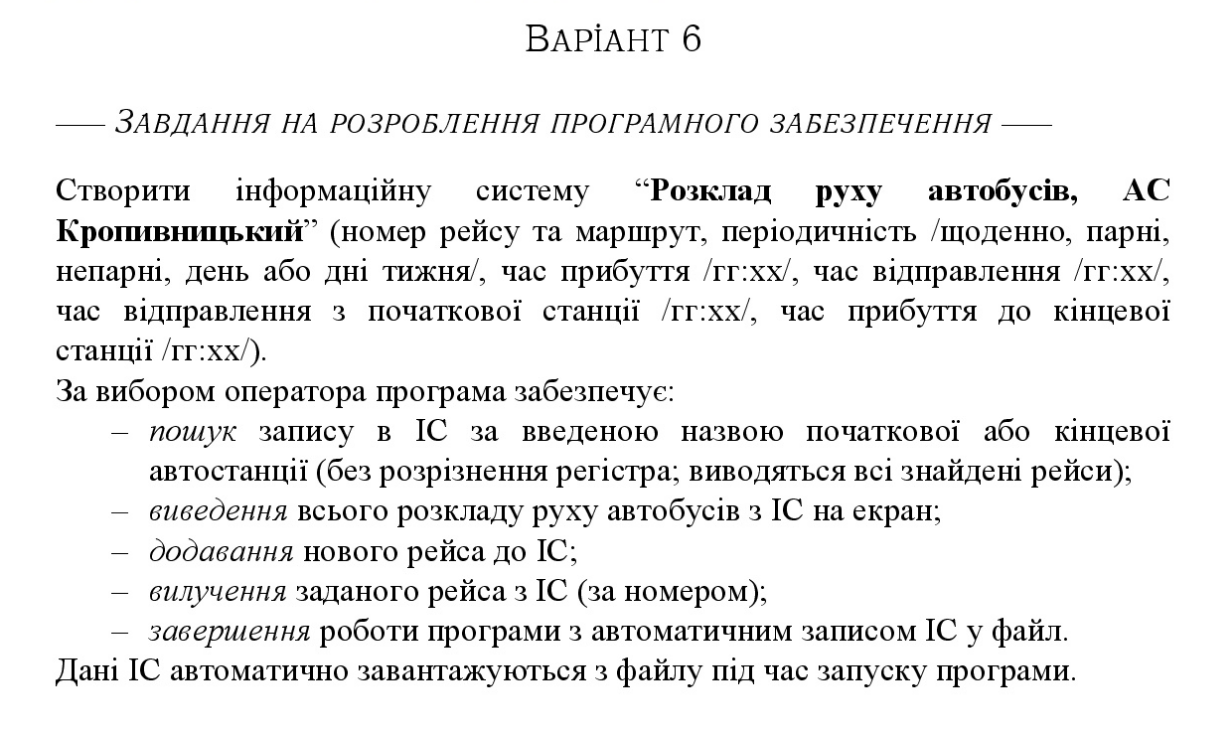
\_\_\_\_\_\_\_\_\_ Собінов О. Г.

Кропивницький – 2023

**Лабораторна робота №11**

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

**Завдання:**

****

**Склад команди IT-проєкта та обов’язки:**

1. Кирило Бездольний:
2. Знайти запис у розкладі рухів автобусів за введеною назвою початкової або кінцевої автостанції.
3. Вивести всі розклади рухів автобусів
4. Пащенко Данило:
5. Отримання розкладу рухів автобусів з файлу.
6. Збереження розкладу рухів автобусів в файл і завершення роботи програми.
7. Олексій Щиченко:
8. Додавання нового рейса до розкладу.
9. Вилучення рейса з розкладу за його номером.

**План роботи з виконання IT-проєкта:**

1. Завантажити Git-репозиторій - https://github.com/odorenskyi/student-name;
2. В файл README.md, записати: тему, мету, завдання та варіант.
3. Створити теки Report, prj, Software, TestSuite.
4. В текі \lab11\Report створити Документ Microsoft Word «Звіт до Лабораторної роботи № 11», в ньому написати титульну сторінку, тему, мету, завдання, варіант, склад команди.
5. Розробити детальний аналіз отриманих підзадач та проектування. Для розроблення даних підзадач потрібно використати створений на meeting, заголовковий файл struct\_type\_project\_6.h.
6. Розробити свої дві підзадачі та записати їх в модуль «Modules*Прізвище*» підключивши заголовковий файл struct\_type\_project\_6.h, вміст файлу включити до звіту.
7. Завантажити файли на GitHub.
8. Коли всі учасники розроблять свої модулі та проведуть модульні тестування, при успішному тестування, завантажте Git-репозиторії інших учасників.
9. В Code::Blocks IDE створити проект консольного додатка prj\_6\_Прізвище.
10. Відповідно до стандарту ISO/IEC 12207 реалізувати програмний засіб згідно з завданням, використовуючи функції (модулі), розроблені учасниками команди ІТ-проекта.
11. Провести системне тестування консольного додатка prj\_6\_Прізвище та свого модуля. При успішному тестування переходи на наступний крок, якщо результат негативний повернутися до додатка, та виправити помилки.
12. Вихідний код (текст) проекта prj\_N\_Прізвище включити до звіту та завантажи на Git-репозиторій.
13. Проаналізувати хід виконання лабораторних завдань і самостійно одержані результати, на основі чого сформулювати обґрунтовані висновки з виконаної лабораторної роботи, викласти їх обсягом не менше двох сторінок машинного (комп’ютерного) тексту та включити до звіту.
14. Підготувати й зберегти у \Lab11\Report звіт про виконання лабораторної роботи, оформлений згідно з ДСТУ 3008:2015 “Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання”, зі змістом, визначеним цим порядком виконання лабораторної роботи.
15. Представити до захисту звіт з виконаної лабораторної роботи і проект у Git-репозиторії https://github.com/odorenskyi/student\_name.

**Аналіз виконання Task 11:**

Спочатку ми зібралися разом, і почали аналізувати завдання. Потім ми обговорили всі проектні рішення, узгодили архітектуру, загальні алгоритми та інтерфейс модулів. Визначили, що ми будемо працювати з vector, для динамічної структури даних. Потім розділи завдання на підзадачі, щоб кожному учаснику команди потрібно було створити по дві підзадачі, тобто функції.

**Постановка виконання Task 11:**

*Вхідні дані:*

1. txt – файл з розкладом рухів автобусів.

*Вихідні дані:*

1. Виведення розкладу рухів автобусів, у консольне вікно, виконання операцій над розкладом.
2. Збереження у файл новий розклад.

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

Я дотримувався такого плану роботи:

Спочатку я завантажив свій Git-репозиторій та переглянув нове завдання в папці tasks.

Потім створив Документ Word, та в нього записав мету, номер варіанту, завдання, складу команди та титульну сторінку. Також в текстовий файл README, записав тему, мету, варіант та завдання.

Потім ми зібралися командою та розпочали мітинг, в ньому ми обговорили такі теми, як:

* Аналіз задачі та вимог до неї.
* Обговорили специфікації ПЗ, концептуальні проектні рішення, сформулювати й узгодити архітектуру програмного засобу, загальні алгоритми функціонування та інтерфейси модулів тощо;
* Визначили вид динамічної структури для реалізації бази даних ПЗ, ми обрали незвичайний тип Vector.
* Створили заголовковий файл struct\_type\_project\_6.h, в ньому описали структура бази даних.
* Розподілили підзадачі, кожний учасник команди отримав по дві підзадачі.
* Склали план роботи з виконання ІТ-проекта, виходячи із розподілених між учасниками команди підзадач та вимог міжнародного стандарту ISO/IEC 12207 щодо виконання процесів реалізації програмних засобів;

Потім ми почали роботи підзадачі самостійно.

Коли всі вже зробили свої підзадачі, кожен учасник команди завантажив модулі інших учасників команд. Я завантажив їх в папку prj.

В Code::Blocks IDE створив проєкт консольного додатка prj\_6\_Прізвище.

Відповідно до стандарту ISO/IEC 12207 реалізував програмний засіб згідно з завданням, використовуючи функції (модулі), розроблені учасниками команди ІТ-проекта.

Завантажив всі файли на Git-репозиторій.

Отже, закінчивши роботу, я зрозумів, що командна робота – це круто. Ви ділите велику задачу, на маленькі підзадачі, які потім окремо реалізовуєте та потім в кінці роботи їх з’єднуєте і цей настрій, це відчуття прекрасний. Ви розумієте, що ви зробили пазл, який потім з’єднали в одну красиву картину. Ця робота мені сподобалась.

**Додаток А**

**Вихідний код заголовкового файлу struct\_type\_project\_6.h:**

#ifndef STRUCT\_TYPE\_PROJECT\_2\_H\_INCLUDED

#define STRUCT\_TYPE\_PROJECT\_2\_H\_INCLUDED

using namespace std;

struct BusSchedule {

short int busRouteNumber;

string busRoute;

string busPeriodicity;

string busDepartureTime;

string busArrivalTime;

string busDepartureTimeStartingStation;

string busArrivalTimeFinalStation;

};

#endif // STRUCT\_TYPE\_PROJECT\_2\_H\_INCLUDED

**Вихідний код модуля ModulesPaschenko:**

*#include <iostream>*

*#include <fstream>*

*#include <vector>*

*#include <string>*

*#include <cstdlib>*

*#include "struct\_type\_project\_6.h"*

*using namespace std;*

*vector<string> split(string str)*

*{*

*vector<string> arr;*

*string delim(";");*

*size\_t prev = 0;*

*size\_t next;*

*size\_t delta = delim.length();*

*while( ( next = str.find( delim, prev ) ) != string::npos ){*

*string tmp = str.substr( prev, next-prev );*

*arr.push\_back( str.substr( prev, next-prev ) );*

*prev = next + delta;*

*}*

*string tmp = str.substr( prev );*

*arr.push\_back( str.substr( prev ) );*

*return arr;*

*}*

*vector<BusSchedule> getDataFromFile()*

*{*

*ifstream f\_read("C:\\Users\\User\\Documents\\Pashchenko-Danylo-KN22\\lab11\\prj\\BusSchedule.txt", ios::in);*

*if(!f\_read.is\_open())*

*{*

*cout << endl << "Помилка відкриття файлу" << endl;*

*exit(EXIT\_FAILURE);*

*}*

*vector<BusSchedule> schedule;*

*string line;*

*while(getline(f\_read, line))*

*{*

*vector<string> splitLine = split(line);*

*BusSchedule bus;*

*bus.busRouteNumber = stoi(splitLine[0]);*

*bus.busRoute = splitLine[1];*

*bus.busPeriodicity = splitLine[2];*

*bus.busDepartureTime = splitLine[3];*

*bus.busArrivalTime = splitLine[4];*

*bus.busDepartureTimeStartingStation = splitLine[5];*

*bus.busArrivalTimeFinalStation = splitLine[6];*

*schedule.push\_back(bus);*

*}*

*f\_read.close();*

*return schedule;*

*}*

*void saveDataInFile(vector<BusSchedule>& schedule)*

*{*

*ofstream f\_write("..\\BusSchedule.txt", ios::out);*

*if(!f\_write.is\_open())*

*{*

*cout << endl << "Помилка відкриття файлу" << endl;*

*return;*

*}*

*for(const auto& bus : schedule)*

*{*

*f\_write << bus.busRouteNumber << ";" << bus.busRoute << ";" << bus.busPeriodicity << ";" << bus.busDepartureTime << ";" << bus.busArrivalTime << ";" << bus.busDepartureTimeStartingStation << ";" << bus.busArrivalTimeFinalStation << endl;*

*}*

*f\_write.close();*

*}*

**Вихідний код заголовкового файлу ModulesPaschenko:**

#ifndef DATABASESTRUCT\_H\_INCLUDED

#define DATABASESTRUCT\_H\_INCLUDED

#include "struct\_type\_project\_6.h"

#include <vector>

using namespace std;

vector<BusSchedule> getDataFromFile();

void saveDataInFile(vector<BusSchedule>& schedule);

#endif // DATABASESTRUCT\_H\_INCLUDED

**Вихідний код модуля ModulesBezdolnyi:**

*#include <iostream>*

*#include <vector>*

*#include <cctype>*

*#include <struct\_type\_project\_6.h>*

*using namespace std;*

*vector<string> splitRoute(string str)*

*{*

*vector<string> arr;*

*string delim("-");*

*size\_t prev = 0;*

*size\_t next;*

*size\_t delta = delim.length();*

*while( ( next = str.find( delim, prev ) ) != string::npos ){*

*string tmp = str.substr( prev, next-prev );*

*arr.push\_back( str.substr( prev, next-prev ) );*

*prev = next + delta;*

*}*

*string tmp = str.substr( prev );*

*arr.push\_back( str.substr( prev ) );*

*return arr;*

*}*

*void outputRecord(int busRouteNumber, string busRoute, string busPeriodicity, string busDepartureTime, string busArrivalTime, string busDepartureTimeStartingStation, string busArrivalTimeFinalStation)*

*{*

*cout << "| Номер рейсу: " << busRouteNumber << endl*

*<< "| Маршрут: " << busRoute << endl*

*<< "| Періодичність: " << busPeriodicity << endl*

*<< "| Час відправлення: " << busDepartureTime << endl*

*<< "| Час прибуття: " << busArrivalTime << endl*

*<< "| Час відправлення з початкової станції: " << busDepartureTimeStartingStation << endl*

*<< "| Час прибуття до кінцевої станції: " << busArrivalTimeFinalStation << endl << endl;*

*}*

*void getRecordUseBusStation(vector<BusSchedule>& schedule)*

*{*

*cout << endl << ">>> Введіть назву початкової або кінцевої станції: ";*

*string sfStation;*

*cin >> sfStation;*

*string sfStationLower;*

*for (char& c : sfStation)*

*{*

*sfStationLower += tolower(c);*

*}*

*bool stationFound = false;*

*for(const auto& bus : schedule)*

*{*

*vector<string> splitText = splitRoute(bus.busRoute);*

*string splitTextLower0;*

*for (char& c : splitText[0])*

*{*

*splitTextLower0 += tolower(c);*

*}*

*string splitTextLower1;*

*for (char& c : splitText[1])*

*{*

*splitTextLower1 += tolower(c);*

*}*

*if(sfStation == splitText[0] || sfStationLower == splitTextLower0 || sfStation == splitText[1] || sfStationLower == splitTextLower1)*

*{*

*outputRecord(bus.busRouteNumber, bus.busRoute, bus.busPeriodicity, bus.busDepartureTime, bus.busArrivalTime, bus.busDepartureTimeStartingStation, bus.busArrivalTimeFinalStation);*

*stationFound = true;*

*}*

*}*

*if(stationFound == false)*

*{*

*cout << endl << "--- Помилка: невірна назва поточної або кінцевої станції ---" << endl << endl;*

*return;*

*}*

*}*

*void getAllInfo(vector<BusSchedule>& schedule)*

*{*

*for(const auto& bus : schedule)*

*{*

*cout << "| Номер рейсу: " << bus.busRouteNumber << endl*

*<< "| Маршрут: " << bus.busRoute << endl*

*<< "| Періодичність: " << bus.busPeriodicity << endl*

*<< "| Час відправлення: " << bus.busDepartureTime << endl*

*<< "| Час прибуття: " << bus.busArrivalTime << endl*

*<< "| Час відправлення з початкової станції: " << bus.busDepartureTimeStartingStation << endl*

*<< "| Час прибуття до кінцевої станції: " << bus.busArrivalTimeFinalStation << endl << endl;*

*}*

*}*

**Вихідний код заголовкового файлу ModulesBezdolnyi:**

#ifndef MODULESKOZLOV\_H\_INCLUDED

#define MODULESKOZLOV\_H\_INCLUDED

#include <vector>

#include <struct\_type\_project\_6.h>

using namespace std;

void getRecordUseBusStation(vector<BusSchedule>& schedule);

void getAllInfo(vector<BusSchedule>& schedule);

#endif // MODULESKOZLOV\_H\_INCLUDED

**Вихідний код модуля ModulesShchychenko:**

*#include <iostream>*

*#include <vector>*

*#include <string>*

*#include "struct\_type\_project\_6.h"*

*using namespace std;*

*void addNewBusRoute(vector<BusSchedule>& schedule)*

*{*

*cout << endl << ">>> УВАГА: писати без пробілів (через нижнє підкреслення)" << endl;*

*cout << ">>> Введіть номер рейсу: ";*

*string busRouteNumber;*

*cin >> busRouteNumber;*

*cout << ">>> Введіть маршрут: ";*

*string busRoute;*

*cin >> busRoute;*

*cout << ">>> Введіть періодичність (наприклад: щоденно, день або декілька днів): ";*

*string busPeriodicity;*

*cin >> busPeriodicity;*

*cout << ">>> Введіть час відправлення (введіть 0, якщо не задано): ";*

*string busDepartureTime;*

*cin >> busDepartureTime;*

*cout << ">>> Введіть час прибуття (введіть 0, якщо не задано): ";*

*string busArrivalTime;*

*cin >> busArrivalTime;*

*cout << ">>> Введіть час відправлення з початкової станції (введіть 0, якщо не задано): ";*

*string busDepartureTimeStartingStation;*

*cin >> busDepartureTimeStartingStation;*

*cout << ">>> Введіть час прибуття до кінцевої станції (введіть 0, якщо не задано): ";*

*string busArrivalTimeFinalStation;*

*cin >> busArrivalTimeFinalStation;*

*cout << endl;*

*BusSchedule bus;*

*bus.busRouteNumber = stoi(busRouteNumber);*

*bus.busRoute = busRoute;*

*bus.busPeriodicity = busPeriodicity;*

*bus.busDepartureTime = busDepartureTime;*

*bus.busArrivalTime = busArrivalTime;*

*bus.busDepartureTimeStartingStation = busDepartureTimeStartingStation;*

*bus.busArrivalTimeFinalStation = busArrivalTimeFinalStation;*

*schedule.push\_back(bus);*

*}*

*void deleteBusRoute(vector<BusSchedule>& schedule)*

*{*

*cout << endl << ">>> Введіть номер рейсу: ";*

*short int routeNumber;*

*cin >> routeNumber;*

*if(!routeNumber)*

*{*

*cout << endl << "--- Помилка: Невірний номер рейсу ---" << endl;*

*return;*

*}*

*for(int i = 0; i < schedule.size(); i++)*

*{*

*if(routeNumber == schedule[i].busRouteNumber)*

*{*

*schedule.erase(schedule.begin() + i);*

*cout << endl << "--- Готово ---" << endl << endl;*

*return;*

*}*

*}*

*cout << endl << "--- Помилка: Рейсу з таким номером не існує ---" << endl << endl;*

*}*

**Вихідний код заголовкового файлу ModulesShchychenko:**

#ifndef MODULESZADOROZHNY\_H\_INCLUDED

#define MODULESZADOROZHNY\_H\_INCLUDED

#include <vector>

#include "struct\_type\_project\_6.h"

using namespace std;

void addNewBusRoute(vector<BusSchedule>& schedule);

void deleteBusRoute(vector<BusSchedule>& schedule);

#endif // MODULESZADOROZHNY\_H\_INCLUDED

**Додаток Б**

**Системне тестування prj\_6\_Прізвище файл TestSuite №11:**