

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 C++ для керування динамічною пам'яттю та бінарними файловими потоками.

Завдання:

ВАРІАНТ 6

— ЗАВДАННЯ НА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ —

Створити інформаційну систему “**Розклад руху автобусів, АС Кропивницький**” (номер рейсу та маршрут, періодичність /щоденно, парні, непарні, день або дні тижня/, час прибуття /гг:хх/, час відправлення /гг:хх/, час відправлення з початкової станції /гг:хх/, час прибуття до кінцевої станції /гг:хх/).

За вибором оператора програма забезпечує:

- пошук запису в ІС за введеною назвою початкової або кінцевої автостанції (без розрізнення регістра; виводяться всі знайдені рейси);
- виведення всього розкладу руху автобусів з ІС на екран;
- додавання нового рейса до ІС;
- вилучення заданого рейса з ІС (за номером);
- завершення роботи програми з автоматичним записом ІС у файл.

Дані ІС автоматично завантажуються з файлу під час запуску програми.

Склад команди ІТ-проєкта та обов'язки:

- Кирило Бездольний:
 - Знайти запис у розкладі рухів автобусів за введеною назвою початкової або кінцевої автостанції.
 - Вивести всі розклади рухів автобусів
- Пашенко Данило:
 - Отримання розкладу рухів автобусів з файлу.

- Збереження розкладу рухів автобусів в файл і завершення роботи програми.
- Олексій Щиченко:
 - Додавання нового рейса до розкладу.
 - Вилучення рейса з розкладу за його номером.

План роботи з виконання ІТ-проєкта:

- Завантажити Git-репозиторій - <https://github.com/odorenskyi/student-name;>
 - В файл README.md, записати: тему, мету, завдання та варіант.
 - Створити теки Report, prj, Software, TestSuite.
 - В текі \lab11\Report створити Документ Microsoft Word «Звіт до Лабораторної роботи № 11», в ньому написати титульну сторінку, тему, мету, завдання, варіант, склад команди.
 - Розробити детальний аналіз отриманих підзадач та проектування. Для розроблення даних підзадач потрібно використати створений на meeting, заголовковий файл struct_type_project_6.h.
 - Розробити свої дві підзадачі та записати їх в модуль «ModulesПрізвище» підключивши заголовковий файл struct_type_project_6.h, вміст файлу включити до звіту.
 - Завантажити файли на GitHub.
 - Коли всі учасники розроблять свої модулі та проведуть модульні тестування, при успішному тестування, завантажте Git-репозиторії інших учасників.
 - В Code::Blocks IDE створити проєкт консольного додатка prj_6_Прізвище.

- Відповідно до стандарту ISO/IEC 12207 реалізувати програмний засіб згідно з завданням, використовуючи функції (модулі), розроблені учасниками команди IT-проекта.
- Провести системне тестування консольного додатка prj_6_Прізвище та свого модуля. При успішному тестуванні переходити на наступний крок, якщо результат негативний повернутися до додатка, та виправити помилки.
- Вихідний код (текст) проекту prj_N_Прізвище включити до звіту та завантажити на Git-репозиторій.
- 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.
- 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 оформлення”, зі змістом, визначеним цим порядком виконання лабораторної роботи.
- 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:

- Виведення розкладу рухів автобусів, у консольне вікно, виконання операцій над розкладом.
- 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 C++ 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 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: