Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 10

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ОБРОБЛЕННЯ ДАНИХ СКЛАДОВИХ ТИПІВ З ФАЙЛОВИМ ВВЕДЕННЯМ/ВИВЕДЕННЯМ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

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

ВИКОНАВ

студент академічної групи КБ-21

Іванов М.Л.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Усік П.С.

Кропивницький – 2022

**Лабораторна робота №10**

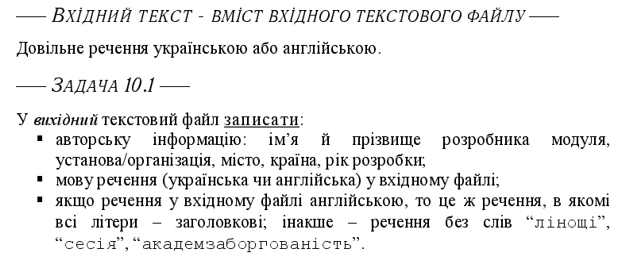
**Тема:** реалізація програмних модулів оброблення даних складових типів з файловим введенням/виведенням

**Мета роботи:** полягає у набутті ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування C++ програминх модулів створення й оброблення даних типів масив, структур, об'єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символьнох інформації.

**Завдання:**

1. Реалізувати програмні модулі розв'язування задач 10.1-10.3 як складових статичної бібліотеки libModulesПрізвище.a (проект ModulesПрізвище, створений під час виконання лабораторних робіт №8-9).
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв'язування задач 10.1-10.3.

**Варіант** **12**

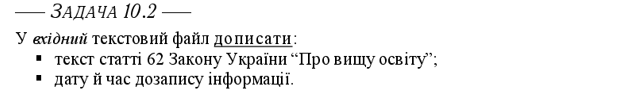


Малюнок 1 – Умова задачі 10.1

**Строга постановка задачі:**

Вхідні дані: path, a – рядки.

Вихідні дані:

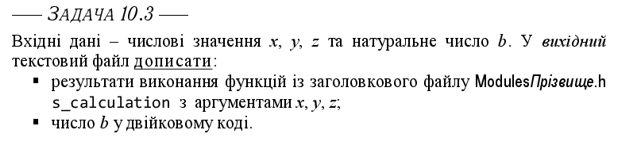


Малюнок 2 – Умова задачі 9.2

**Строга постановка задачі:**

Вхідні дані: path – рядок

Вихідні дані: –.



Малюнок 3 – Умова задачі 9.3

**Строга постановка задачі:**

Вхідні дані: path – рядок,

x, y, z – дійсні числа,

b – ціле число.

Вихідні дані:–.

**Проектування програмного модуля:**

Модуль ModulesIvanov, що вже містить функцію s\_calculation, розширюється шляхом додавання нових функцій та структур даних, призначених для реалізації повернення декількох змінних функціями.

Функція addSmth приймає два рядки, що позначають назву вхіного та вихідного файлів. Із вхідного файлу за допомогою ifstream зчитується зміст (речення). Після цього за допомогою ofsteam відкривається вихідний файл, увесь зміт якого знищуються (якщо файлу не існує – віг створюється). У нього записується інформація про розробника, речення із вхідного файлу з визначеними символами заміненими на «?», кількість літер у рядку зі змістом вхідного файлу та в залежності віл того, парна вона чи ні – відповідний вірш.

Функція PrintRule62 також приймає два рядки, що позначають назву вхіного та вихідного файлів. Із вхідного файлу зчитується зміст, після чого у кінець вихідного файлу дописуються непарні символи (кожний другий, починаючи з найпершого). Після цього дописується дата дозапису.

Функція CalcBin приймає рядок з назвою вихідного файлу, 3 дійсеих числа та ціле число. Ця функція дописує у кінець вихідного файлу результат функції s\_calculations зі змінними x, y та z в якості аргументів та число b у двійковому вигляді.

**Результати тестування модулів для задач 10.1-10.3:**

**TestDriver of ModulesIvanov.a**

**====================================**

**|Тестовий драйвер для завдання 10.1|**

**====================================**

**Визначення мови... Done**

**Привітання, Написання... Done**

**Зміна тексту... Done**

**TC-1 passed**

**Визначення мови... Done**

**Привітання, Написання... Done**

**Зміна тексту... Done**

**TC-2 passed**

**Визначення мови... Done**

**Привітання, Написання... Done**

**Зміна тексту... Done**

**TC-3 passed**

**====================================**

**|Тестовий драйвер для завдання 10.2|**

**====================================**

**Написання закону... Done**

**TC-1 passed**

**Написання закону... Done**

**TC-2 passed**

**Написання закону... Done**

**TC-3 passed**

**====================================**

**|Тестовий драйвер для завдання 10.3|**

**====================================**

**Підрахунок... Done**

**Підрахунок символів... Done**

**Написання кількості символів... Done**

**TC-1 passed**

**Підрахунок... Done**

**Підрахунок символів... Done**

**Написання кількості символів... Done**

**TC-2 passed**

**Підрахунок... Done**

**Підрахунок символів... Done**

**Написання кількості символів... Done**

**TC-3 passed**

**Висновок:** дана лабораторна робота була націлена на набуття навичок

у реалізації технології модульного програмування та оброблення даних складових типыв з файловим введенням/виведенням під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

До вихідного коду статичної бібліотеки, створеної під час виконання лабораторної роботи №8 та розширеної під час виконання лабораторної роботи №9, було додано реалізації функцій для розв’язування задач 10.1-10.3 та створено допоміжні структури для більш зручної розробки. Після компіляції проекту бібліотеки в теці \obj оновився .а файл, тобто файл статичної бібліотеки. Наступним кроком було додавання до заголовкового файлу бібліотеки прототипів функцій та нових структур, які повертаються функціями.

Під час розв'язування задач було створено декілька допоміжних функцій, які по суті є частинами розв'язків, та загальні функції, які роблять попередні налагодження та викликають інші функції. Таким чином, до заголовкового файлу було додано прототипи тільки загальних функцій-розв’язків, з чого слідує, що допоміжні функції не матимуть змоги бути викликаними з будь-якого місця окрім самих загальних функцій. Це було зроблено тільки для того, щоб через цей заголовковий файл можна було користуватись тільки функціями, потрібними для розвя’зування відповідних задач.

Усі задачі вимагали реалізації роботи з вхідними та вихідними файлами, що було зроблено за допомогою потоків ifstream та ofstream зі стандартної бібліотеки <fstream>. Саме робота з файлами була реалізована у загальних функціях статичної бібліотеки, про які йшлося вище. Кожне відкриття файлів та назначеня їх відповідним потокам супроводжувалося перевіркою на те, чи успішно відкрито файл, та у випадку функція повідомляла про помилку та повертала керування у місце, з якого була викликана. В іншому випадку виконувалися потрібні операції з файлом (зчитування/запис) та файл закривався. Закриття файлу забезпечує надійність збереження змісту файлу та вважається гарними тоном. Також загальним функціям, які призначені для безпосереднього використання у подальших ПЗ, були забезпечені інтерфейсом у вигляді параметрів типу string, які призначені для передачі у функцію назв вхідного та/або вихідного файлів.

Перша задача вимагала запису до вихідного файлу анотацій розробника, речення з вхідного файлу з певними літерами заміненими на “?” та запису відповідного вірша в залежності від кількості літер (без знаків пунктуації, цифр та ін.) у вхідному файлі. <fstream> дозволяє одночасне відкриття декількох файлових потоків, тож спочатку для запису відкривається вихідний файл (створюється потік типу ofstream). За вимогами, він має бути перезаписаним або створеним, якщо він відсутній – для цього використовується флаг ios::out (аналог “w”). Якщо відкриття відбулося успішно, в нього записується анотація, після чого у паралельному файловому потоці для читання відкриважться вхідний файл (створюється потік типу ofstream) – це відбувається за допомогою флагу ios::in (анагог “r”). Після перевікрки на відкритя з файлу построково зчитується інформація, яка зберігається у змінну типу string. Важливо відмітити, що вхідні файли створюються користувачем, тож для коректної роботи застосунку вони мають бути збережені у кодуванні ANSI. Після зчитуання файл закривається, і зміст, збережений заздалегідь, обробляється шляхом заміни символів, зазначених у завданні, на “?”. Це було реалізовано за допомогою типу set<тип> вбудованої бібліотеки <set>. Тип даних set реалізує роботу з множинами (непроіндексований набір унікальних значень), що допомагає більш зручно перевіряти, чи є символ в переліку. У циклі посимвольно перевіряються символи рядка, та якщо символ присутній у множині, від перезаписується. Після виведення зміненого рядка у вихідний файл повторно відбувається процедура зі зчитуванням вхідного файлу, далі додаткова функція приймає рядок в якості аргументу та підраховує кількість літер у рядку – це значення повертається в складену функцію, береться за модулем 2 та у файл виводиться один з двох віршів. Після цих дій файл закривається.

Друга задача полягала в тому, щоб у вихідний файл дописати непарні символи з вхідного речення та дату й час дозапису. Для того, щоб дописати інформацію в кінець файлк, його потрібно відкрити для запису з флагом ios::app. Далі вхідне речення зчитується та кожний непарний символ записується у файл. Непарністю в даному випадку вважається позиція символа у тексті з точки зору корисувача (1 символ, 3 символ...), тож на програмному рівні непарні символи рядка – ті, що маєть парний індекс. Дата та час дозапису були реалізовані за допомогою бібліотеки <ctime>, що дає змогу створити змінну типу time\_t, зберігти в неї поточно дату та час у комп'ютерному вигляді (кількість секунд, що пройшла з початку так званої «цифрової ери») за допомогою функції time() та перевести значення цієї змінної у зрозумілий людині вигляд за допомогою функції ctime().

У третій задачі потрібно було у вихідний файл дописати результат функції s\_calculation() з аргументами x, y, z, та записати число b у двійковому вигляді. Загальна функція для розв’язку третьої задачі має відповідні параметри, що дозволяють через неї передавати аргументи функції s\_calculation(). Число b було переведено у двійковий вигляд за допомогою типу даних bitset, що міститься в однойменній бібліотеці. Робота з файлом аналогічна такій, що була у задачі 2.

Далі було створено проект консольного додатка С++ під назвою TestDriver, метою створення якого є реалізація модульного тестування. Для кожної функції було розраховано еталонні вхідні дані та відповідні еталонні результати, які порівнювалися з результатами роботи функцій. Через те, що 2 та 3 задачі випагають дозапису у файл, було прийняте рішення розробити 9 файлів з еталонними результатами, відповідно кожний вихідний файл (3 тести – 3 файли) порівнювався з відповідним файлом очікуваного результату для поточного етапу роботи застосунку (умовно поділемо роботу застосунку на три етапи: перший – перезапис у файл розв'язку першої задачі, другий – додання результату роботи другої функцій, третій – додання результату роботи реалізації третьої задачі). Застосунок було спроектовано для автоматиного тестування, тож на посатку роботи тестового драйвера створюються вхідні файли, в які автоматично записуються еталонні вхідні дані.

Для першої функції перевірялася повна відповідність змісту у файі, що є результатом роботи програми, та очікуваним результатом.

Для другої функції перевірялося те саме, що й у першій функціїї, за виключенням того, що частина рядка з датою та часом перевірялися на відповіднсть формату часу (Www Mmm DD HH:MM:SS YYYY). Через те, що довжина рядка з датою та часом є костантною, вона була зазначена за допомогою макросу #define.

Для третьої функціїї увесь зміст, який знаходиться до дописаних даних, перевіряється так само, як у для другої функції, а дописані дані – так само, як для першої. Проблемою є те, що для різних вхідних даних для першої та другої функцій результат займає різну кількість пам’яті, тож для перевірки формату дати та часу довелось реалізувати пошук цього підрядка у файлі.

Через те, що у вихідному коді міститься велика кількість кириличних символів, усі файли проекту було переформатовано та змінено для роботи у кодуванні UTF-8 для коректного відображення на платформі GitHub. Проте для того, щоб полегшити роботу з обробкою та выдображенням тексту в консолы та файлах, у додаткових налаштуваннях компілятору для певного проекту було введено команду для того, щоб при білдингу коду усі потрібні файли з UTF-8 конвертувалися у CP1251.

Усі налаштування, перераховані у висновку, були застосовані до проектів, а не до середовища, для того, щоб проект успішно виконувався в IDE Code::Blocks на різних пристроях, незалежно від налаштувань певного середовища.

**ДОДАТОК А**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_lab10.1 |
| Назва проекта / Модуля  Name of Project / Unit | ModulesIvanov.a |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Max Ivanov |
| Виконавець  Implementer | Max Ivanov |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TC\_01 | 1. Викликати функцію addSmth( , ) з параметрами «driver.txt» та “a” | Вікно застосунку:  asdfasdf  Визначення мови... Done  Привітання, Написання... Done  Зміна тексту... Done  ==============================  |Іванов Максим |  |ЦНТУ, Кропивницький, Україна|  |2022 |  |=============================  asdfasdf  Мова: Англійська  ASDFASDF. |  |
| TC\_02 | 1. Викликати функцію addSmth( , ) з параметрами ««a»» та «output2.txt» | Вікно застосунку:  asdfasdfa  Визначення мови... Done  Привітання, Написання... Done  Зміна тексту... Done  ==============================  |Іванов Максим |  |ЦНТУ, Кропивницький, Україна|  |2022 |  |=============================  asdfasdfsa  Мова: Англійська  ASDFASDFSA |  |
| TC\_03 | 1. Викликати функцію addSmth( , ) з параметрами «input3.txt» та «output3.txt» | Вікно застосунку:  Привіт сесія  Визначення мови... Done  Привітання, Написання... Done  Зміна тексту... Done  ==============================  |Іванов Максим |  |ЦНТУ, Кропивницький, Україна|  |2022 |  |=============================  Привіт сесія  Мова: Англійська  Привіт |  |

**ДОДАТОК Б**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_lab10.2 |
| Назва проекта / Модуля  Name of Project / Unit | ModulesIvanov.a |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Max Ivanov |
| Виконавець  Implementer | Max Ivanov |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TC\_01 | 1. Викликати функцію CalcBin( ) з параметрами «driver.txt» | Вікно застосунку:  Уведіть x, y, z:  3  5  1  Уведіть b: 123456  Підрахунок... Done  S = -104.961  00000000000000011110001001000000 |  |
| TC\_02 | Викликати функцію CalcBin( , ) з параметрами ««a»» та «driver.txt» | Вікно застосунку  Уведіть x, y, z:  3  5  1  Уведіть b: 436345234  Підрахунок... Done  S = -1985.64  00000001011001101111011100001110 |  |
| TC\_03 | 1. Викликати функцію CalcBin( , ) з параметрами «driver.txt» | Вікно застосунку  Уведіть x, y, z:  4  12  3  Уведіть b: 1534634  Підрахунок... Done  S = -1568.01  00000000000101110110101010101010 |  |

**ДОДАТОК Г**

**Лістинг вихідного коду проекту ModulesIvanov:**

#include <cmath>

#include <windows.h>

#include <iostream>

#include <bitset>

#include <string>

#include <ctime>

using namespace std;

double c\_calculation(double x, double y)

{

double S;

S = pow(x, 2) - pow(y, 3) + sqrt(pow(y, x) - M\_PI);

return S;

}

float Deposite(float dep, short strok)

{

float month, a;

if(strok == 6)

{

a = (dep\*16)/100;

month = a/6;

}

if (strok == 12)

{

a = (dep\*18)/100;

month = a/12;

}

return month;

}

short Bolfort(float jan[])

{

short sz = 31;

float bolfort[sz];

short maximum = 0;

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

{

if (jan[i] < 0.3)

{

bolfort[i] = 0;

continue;

}

if ((jan[i] > 0.3) && (jan[i] < 1.5))

{

bolfort[i] = 1;

continue;

}

if ((jan[i] > 1.6) && (jan[i] < 3.4))

{

bolfort[i] = 2;

continue;

}

if ((jan[i] > 3.5) && (jan[i] < 5.4))

{

bolfort[i] = 3;

continue;

}

if ((jan[i] > 5.5) && (jan[i] < 7.9))

{

bolfort[i] = 4;

continue;

}

if ((jan[i] > 8.0) && (jan[i] < 10.7))

{

bolfort[i] = 5;

continue;

}

if ((jan[i] > 10.8) && (jan[i] < 13.8))

{

bolfort[i] = 6;

continue;

}

if ((jan[i] > 13.9) && (jan[i] < 17.1))

{

bolfort[i] = 7;

continue;

}

if ((jan[i] > 17.2) && (jan[i] < 20.7))

{

bolfort[i] = 8;

continue;

}

if ((jan[i] > 20.8) && (jan[i] < 24.4))

{

bolfort[i] = 9;

continue;

}

if ((jan[i] > 24.5) && (jan[i] < 28.4))

{

bolfort[i] = 10;

continue;

}

if ((jan[i] > 28.5) && (jan[i] < 32.6))

{

bolfort[i] = 11;

continue;

}

if (jan[i] >= 32.7)

{

bolfort[i] = 12;

continue;

}

}

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

{

if(maximum < bolfort[i])

maximum = bolfort[i];

}

return maximum;

}

int BitNumber(int num)

{

bitset<32> b\_number(num);

if(b\_number[13])

{

return b\_number.count();

}

return 32 - b\_number.count();

}

string greetings()

{

return "==============================\n"

"|Іванов Максим |\n"

"|ЦНТУ, Кропивницький, Україна|\n"

"|2022 |\n"

"|=============================\n\n";

}

string addSmth(string path, string text)

{

bool flag = true;

string beta = text;

fstream file(path, ios::out);

file.close();

file.open(path, ios::in);

if(file.is\_open())

{

cout << "Визначення мови... ";

for(int i = 0; i < text.length(); i++)

{

if((text[i] >= 'A' && text[i] <= 'Z') || (text[i] >= 'a' && text[i] <= 'z'))

flag = true;

if((text[i] >= 'А' && text[i] <= 'Я') || (text[i] >= 'а' && text[i] <= 'я'))

flag = false;

}

cout << "Done" << endl;

}

else

{

cout << "Визначення мови... error" << endl;

}

file.close();

file.open(path, ios::app);

if(file.is\_open())

{

cout << "Привітання, Написання... ";

file << greetings();

file << text << endl;

file << "Мова: " << ((flag == true) ? "Англійська\n" : "Українська\n");

cout << "Done" << endl;

}

else

cout << "Привітання, Написання error" << endl;

file.close();

file.open(path, ios::in);

if(file.is\_open())

{

cout << "Зміна тексту... ";

if(flag == true)

{

for(int i =0; i < text.length(); i++)

{

if(text[i] >= 'a' && text[i] <= 'z')

beta[i] = text[i] - 32;

else

beta[i] = text[i];

}

cout << "Done" << endl;

return beta;

}

else

{

for(int i =0; i < text.length(); i++)

{

if((text[i] == 'л') && (text[i+1] == 'і') && (text[i+2] == 'н') && (text[i+3] == 'о') && (text[i+4] == 'щ') && (text[i+5] == 'і'))

{

beta[i] = ' ';

beta[i+1] = ' ';

beta[i+2] = ' ';

beta[i+3] = ' ';

beta[i+4] = ' ';

beta[i+5] = ' ';

}

if((text[i] == 'с') && (text[i+1] == 'е') && (text[i+2] == 'с') && (text[i+3] == 'і') && (text[i+4] == 'я'))

{

beta[i] = ' ';

beta[i+1] = ' ';

beta[i+2] = ' ';

beta[i+3] = ' ';

beta[i+4] = ' ';

}

if((text[i] == 'а') && (text[i+1] == 'к') && (text[i+2] == 'а') && (text[i+3] == 'д') && (text[i+4] == 'е') && (text[i+5] == 'м')

&& (text[i+6] == 'з') && (text[i+7] == 'а') && (text[i+8] == 'б') && (text[i+9] == 'о') && (text[i+10] == 'р') && (text[i+11] == 'г')

&& (text[i+12] == 'о') && (text[i+13] == 'в') && (text[i+14] == 'а') && (text[i+15] == 'н') && (text[i+16] == 'і') && (text[i+17] == 'с')

&& (text[i+18] == 'т') && (text[i+19] == 'ь'))

{

beta[i] = ' ';

beta[i+1] = ' ';beta[i+2] = ' ';beta[i+3] = ' ';beta[i+4] = ' ';beta[i+5] = ' ';beta[i+5] = ' ';beta[i+5] = ' ';

beta[i+6] = ' ';beta[i+7] = ' ';beta[i+8] = ' ';beta[i+9] = ' ';beta[i+10] = ' ';beta[i+11] = ' ';beta[i+12] = ' ';

beta[i+13] = ' ';beta[i+14] = ' ';beta[i+15] = ' ';beta[i+16] = ' ';beta[i+17] = ' ';beta[i+18] = ' ';beta[i+19] = ' ';

}

}

cout << "Done" << endl;

return beta;

}

}

else

cout << "Зміна тексту error" << endl;

file.close();

file.open(path, ios::app);

if(file.is\_open())

{

cout << "Написання зміненого тексту... ";

file << beta << endl;

cout << "Done" << endl;

}

else

cout << "Написання зміненого тексту error" << endl;

file.close();

long cnt = 0;

string t;

file.open(path, ios::in);

if(file.is\_open())

{

cout << "Підрахунок символів... ";

while(!file.eof())

{

getline(file, t);

cnt += t.length();

}

cout << "Done" << endl;

}

else

cout << "Підрахунок символів error" << endl;

file.close();

file.open(path, ios::app);

if(file.is\_open())

{

cout << "Написання кількості символів... ";

file << "Кількість символів: " << cnt << endl;

cout << "Done" << endl;

}

else

cout << "Написання кількості символів error" << endl;

file.close();

}

long PrintRule62(string path)

{

string rule;

fstream file(path, ios::out);

if(file.is\_open())

{

cout << "Написання закону... ";

rule = "1. Особи, які навчаються у вищих навчальних закладах, мають право на:\n"

"1) вибір форми навчання під час вступу до вищого навчального закладу;\n"

"2) безпечні і нешкідливі умови навчання, праці та побуту;\n"

"3) трудову діяльність у позанавчальний час;\n"

"4) додаткову оплачувану відпустку у зв’язку з навчанням за основним місцем роботи, скорочений робочий час та інші пільги, передбачені законодавством для осіб, які поєднують роботу з навчанням;\n"

"5) безоплатне користування бібліотеками, інформаційними фондами, навчальною, науковою та спортивною базами вищого навчального закладу;\n"

"6) безоплатне забезпечення інформацією для навчання у доступних форматах з використанням технологій, що враховують обмеження життєдіяльності, зумовлені станом здоров’я (для осіб з особливими освітніми потребами);\n"

"7) користування виробничою, культурно-освітньою, побутовою, оздоровчою базами вищого навчального закладу у порядку, передбаченому статутом вищого навчального закладу;\n"

"8) забезпечення гуртожитком на строк навчання у порядку, встановленому законодавством;\n"

"9) участь у науково-дослідних, дослідно-конструкторських роботах, конференціях, симпозіумах, виставках, конкурсах, представлення своїх робіт для публікації;\n"

"10) участь у заходах з освітньої, наукової, науково-дослідної, спортивної, мистецької, громадської діяльності, що проводяться в Україні та за кордоном, у встановленому законодавством порядку;\n"

"11) участь в обговоренні та вирішенні питань удосконалення навчального процесу, науково-дослідної роботи, призначення стипендій, організації дозвілля, побуту, оздоровлення;\n"

"12) внесення пропозицій щодо умов і розміру плати за навчання;\n"

"13) участь у громадських об’єднаннях;\n"

"14) участь у діяльності органів громадського самоврядування вищого навчального закладу, інститутів, факультетів, відділень, вченої ради вищого навчального закладу, органів студентського самоврядування;\n"

"15) вибір навчальних дисциплін у межах, передбачених відповідною освітньою програмою та робочим навчальним планом, в обсязі, що становить не менш як 25 відсотків загальної кількості кредитів ЄКТС, передбачених для даного рівня вищої освіти. При цьому здобувачі певного рівня вищої освіти мають право вибирати навчальні дисципліни, що пропонуються для інших рівнів вищої освіти, за погодженням з керівником відповідного факультету чи підрозділу;\n"

"16) навчання одночасно за декількома освітніми програмами, а також у декількох вищих навчальних закладах, за умови отримання тільки однієї вищої освіти за кожним ступенем за кошти державного (місцевого) бюджету;\n"

"17) академічну мобільність, у тому числі міжнародну;\n"

"18) отримання соціальної допомоги у випадках, встановлених законодавством;\n"

"19) зарахування до страхового стажу відповідно до Закону України \"Про загальнообов’язкове державне пенсійне страхування\" періодів навчання на денній формі навчання у вищих навчальних закладах, аспірантурі, докторантурі, клінічній ординатурі, інтернатурі, резидентурі, за умови добровільної сплати страхових внесків;\n"

"20) академічну відпустку або перерву в навчанні із збереженням окремих прав здобувача вищої освіти, а також на поновлення навчання у порядку, встановленому центральним органом виконавчої влади у сфері освіти і науки;\n"

"21) участь у формуванні індивідуального навчального плану;\n"

"22) моральне та/або матеріальне заохочення за успіхи у навчанні, науково-дослідній і громадській роботі, за мистецькі та спортивні досягнення тощо;\n"

"23) захист від будь-яких форм експлуатації, фізичного та психічного насильства;\n"

"24) безоплатне проходження практики на підприємствах, в установах, закладах та організаціях, а також на оплату праці під час виконання виробничих функцій згідно із законодавством;\n"

"25) канікулярну відпустку тривалістю не менш як вісім календарних тижнів на навчальний рік;\n"

"26) отримання цільових пільгових державних кредитів для здобуття вищої освіти у порядку, визначеному Кабінетом Міністрів України;\n"

"27) оскарження дій органів управління вищого навчального закладу та їх посадових осіб, педагогічних і науково-педагогічних працівників;\n"

"28) спеціальний навчально-реабілітаційний супровід та вільний доступ до інфраструктури вищого навчального закладу відповідно до медико-соціальних показань за наявності обмежень життєдіяльності, зумовлених станом здоров’я.\n"

"2. Особи, які навчаються у вищих навчальних закладах за денною формою навчання за рахунок коштів державного або місцевих бюджетів, мають право на отримання стипендій у встановленому законодавством порядку.\n"

"3. Особи, які навчаються у вищих навчальних закладах за денною формою навчання, можуть отримувати інші стипендії, призначені фізичними (юридичними) особами.\n"

"4. Розмір стипендіального фонду вищого навчального закладу повинен забезпечувати виплату академічних стипендій не менш як двом третинам і не більш як 75 відсоткам студентів (курсантів невійськових вищих навчальних закладів) денної форми навчання, які навчаються за кошти державного бюджету, без урахування осіб, які отримують соціальні стипендії.\n"

"5. Розмір академічної та соціальної стипендій, порядок їх призначення і виплати встановлюються Кабінетом Міністрів України.\n"

"6. Розмір мінімальної академічної та соціальної стипендій для здобувачів освітніх ступенів вищої освіти не може бути меншим, ніж розмір прожиткового мінімуму з розрахунку на одну особу на місяць. Розмір мінімальної академічної та соціальної стипендій для здобувачів ступеня молодшого бакалавра не може бути меншим, ніж дві третини розміру прожиткового мінімуму з розрахунку на одну особу на місяць.\n"

"7. Академічні стипендії виплачуються не менш як двом третинам студентів (курсантів) кожного курсу (крім першого семестру першого року навчання) за кожною спеціальністю, без урахування осіб, які отримують соціальні стипендії. Академічну стипендію отримують студенти (курсанти), які згідно з рейтингом, складеним у вищому навчальному закладі за результатами семестрового контролю, здобули найкращі результати навчання. До проведення першого семестрового контролю академічна стипендія у мінімальному розмірі призначається всім студентам (курсантам), зарахованим до вищого навчального закладу на перший курс навчання.\n"

"8. Особам, які досягли значних успіхів у навчанні та/або науковій діяльності, за рішенням вченої ради вищого навчального закладу можуть призначатися персональні стипендії.\n"

"9. Здобувачі вищої освіти, які навчаються у вищих навчальних закладах за денною формою навчання, мають право на пільговий проїзд у транспорті у порядку, встановленому Кабінетом Міністрів України.\n"

"10. Студенти, курсанти вищих навчальних закладів мають право на отримання студентського квитка, зразок якого затверджується центральним органом виконавчої влади у сфері освіти і науки.\n";

file << rule << endl;

time\_t t;

t = time(NULL);

file << ctime(&t);

cout << "Done" << endl;

}

else

cout << "Написання закону error" << endl;

file.close();

long cnt = 0;

string t;

file.open(path, ios::in);

if(file.is\_open())

{

cout << "Підрахунок символів... ";

while(!file.eof())

{

getline(file, t);

cnt += t.length();

}

cout << "Done" << endl;

}

else

cout << "Підрахунок символів error" << endl;

file.close();

file.open(path, ios::app);

if(file.is\_open())

{

cout << "Написання кількості символів... ";

file << "Кількість символів: " << cnt << endl;

cout << "Done" << endl;

}

else

cout << "Написання кількості символів error" << endl;

file.close();

}

long CalcBin(string path, double x, double y, double z, int b)

{

fstream file(path, ios::out);

if(file.is\_open())

{

cerr << "Підрахунок... ";

bitset<32> b\_number{b};

file << "S = " << c\_calculation(x, y) << endl;

file << b\_number << endl;

cerr << "Done" << endl;

}

else

cout << "Підрахунок error" << endl;

file.close();

long cnt = 0;

string t;

file.open(path, ios::in);

if(file.is\_open())

{

cerr << "Підрахунок символів... ";

while(!file.eof())

{

getline(file, t);

cnt += t.length();

}

cerr << "Done" << endl;

}

else

cout << "Підрахунок символів error" << endl;

file.close();

file.open(path, ios::app);

if(file.is\_open())

{

cerr << "Написання кількості символів... ";

file << "Кількість символів: " << cnt << endl;

cerr << "Done" << endl;

}

else

cout << "Написання кількості символів error" << endl;

file.close();

return cnt;

}

void HowManySymbols(string path)

{

long cnt = 0;

string t;

fstream file(path, ios::in);

if(file.is\_open())

{

cout << "Підрахунок символів... ";

while(!file.eof())

{

getline(file, t);

cnt += t.length();

}

cout << "Done" << endl;

}

else

cout << "Підрахунок символів error" << endl;

file.close();

file.open(path, ios::app);

if(file.is\_open())

{

cout << "Написання кількості символів... ";

file << "Кількість символів: " << cnt << endl;

cout << "Done" << endl;

}

else

cout << "Написання кількості символів error" << endl;

file.close();

}

**ДОДАТОК Д**

**Лістинг вихідного коду проекту TestDriver:**

#include <iostream>

#include <string>

#include "ModulesIvanov.h"

using namespace std;

int main()

{

system("chcp 1251 & cls");

string input1[3] = {"Ukraine", "Привіт сесія", "Привіт, сусіде"};

string expected1[3] = {"UKRAINE", "Привіт ", "Привіт, сусіде"};

cout << "====================================" << endl

<< "|Тестовий драйвер для завдання 10.1|" << endl

<< "====================================" << endl << endl;

for(int i = 0; i < 3; i++)

{

if(addSmth("driver.txt", input1[i]) == expected1[i])

{

cout << "TC-" << i+1 << " passed\n" << endl;

}

else

cout << "failed" << endl;

}

cout << "====================================" << endl

<< "|Тестовий драйвер для завдання 10.2|" << endl

<< "====================================" << endl << endl;

string input2[3];

for(int i = 0; i < 3; i++)

{

time\_t s;

s = time(NULL);

if(PrintRule62("driver.txt") == (ctime(&s)))

{

cout << "TC-" << i+1 << " passed" << endl;

}

else

cout << "TC-" << i+1 << " failed" << endl;

}

cout << "====================================" << endl

<< "|Тестовий драйвер для завдання 10.3|" << endl

<< "====================================" << endl << endl;

int input3[3] {123456, 654565321, 228337};

long expected3[3] {44, 44, 44};

for(int i = 0; i < 3; i++)

{

if(CalcBin("driver.txt", 3,5,1, input3[i]) == expected3[i])

{

cout << "TC-" << i+1 << " passed" << endl;

}

else

cout << "TC-" << i+1 << " failed" << endl;

}

return 0;

}