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 ТИІВ 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-24

Радченко М. Д.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А. С.

Кропивницький – 2025

Мета роботи полягає у набутті ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування С++ програмних модулів створення й оброблення даних типів масив, структура, об’єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символьної інформації. **Варіант №12**

**Завдання до лабораторної роботи**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesRadchenko.а (проект ModulesRadchenko лабораторних робіт №8–9).
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3.

# 1. Аналіз задачі 10.1

## 1.1. Вимоги та постановка задачі

* **Мета**: Створити вихідний текстовий файл із записом авторської інформації, інформації про мову вхідного речення та обробленого речення.
* **Вхідні дані**: Текстовий файл, який містить довільне речення (як українською, так і англійською).
* **Обробка речення**:
  + **Визначення мови**: Проаналізувати вміст файла для визначення, чи переважають латинські символи (англійська) чи кириличні (українська).
  + **Обробка англійського речення**: Перетворити всі літери у верхній регістр.
  + **Обробка українського речення**: Видалити слова «лінощі», «сесія», «академзаборгованість» (з урахуванням можливих варіацій регістру).
* **Запис вихідних даних**:
  + Запис авторської інформації:

"Інформація про автора: Максим Радченко, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025."

* + Запис інформації про мову вхідного речення.
  + Запис обробленого речення відповідно до визначеної логіки.
* **Особливості реалізації**:
  + Файл для запису створюється, якщо не існує, або очищається (перезаписується) перед записом.
  + Перевірка відкриття файлового потоку як для читання, так і для запису.

## 1.2. Архітектурне проектування (10.1)

* **Модуль читання файлу**:
  + Читає вхідне речення з файлу.
  + Перевірка успішного відкриття файлу.
* **Модуль аналізу тексту**:
  + Функція для аналізу символів: якщо більшість символів – латиниця, визначається англійська мова, інакше – українська.
  + Якщо англійський текст – застосовується функція конвертації до верхнього регістру.
  + Якщо український текст – застосовується функція пошуку та виключення слів "лінощі", "сесія", "академзаборгованість".
* **Модуль запису файлу**:
  + Створює або очищує вихідний файл.
  + Записує послідовно: авторські дані, інформацію про мову, оброблене речення.
  + Закриває файловий потік після запису.

# 2. Аналіз задачі 10.2

## 2.1. Вимоги та постановка задачі

* **Мета**: Доповнити (дозаписати) вхідний текстовий файл додатковою інформацією.
* **Додаткова інформація**:
  + Текст статті 62 Закону України "Про вищу освіту". Цей текст може бути збережений як константа або завантажуватись із зовнішнього джерела.
  + Дата та час моменту дозапису (отримання локальної дати та часу через стандартні функції з <ctime>/<time.h>).
* **Режим роботи з файлом**:
  + Вхідний файл відкривається в режимі допису (append).
  + Потрібна перевірка відкриття потоку для дозапису.
  + Після операцій потік слід закрити.

## 2.2. Архітектурне проектування (10.2)

* **Модуль дозапису інформації**:
  + Функція для відкриття файлу в режимі append.
  + Функція отримання поточного часу у вигляді рядка.
  + Запис тексту статті 62 та рядка з датою й часом.
* **Обробка файлу**:
  + Відкриття файлу, перевірка успішного відкриття, дозапис та закриття потоку.

# 3. Аналіз задачі 10.3

## 3.1. Вимоги та постановка задачі

* **Мета**: Доповнити вихідний текстовий файл результатами числових обчислень.
* **Вхідні дані**:
  + Три числові значення: **x**, **y**, **z**.
  + Натуральне число **b**.
* **Обчислення**:
  + Виклик функції s\_calculation(x, y, z) з заголовкового файлу ModulesZaritskyi для отримання певного результату.
  + Перетворення числа **y** до двійкового формату (через алгоритм послідовного ділення на 2 з накопиченням остач).
* **Вивід**:
  + Дописування до вихідного файлу результату виконання функції s\_calculation.
  + Дописування числа **y** у вигляді рядка, що є двійковим представленням.
* **Режим роботи з файлом**:
  + Вихідний файл відкривається в режимі дозапису (append) із відповідною перевіркою відкриття.

# Аргументи досягнення мети:

1. Поділ завдань на окремі частини (10.1, 10.2, 10.3) дозволив чітко розмежувати функціональність, що значно полегшує тестування й обслуговування програмного коду.
2. Коректне декларування функцій у заголовочному файлі (ModulesRadchenko.h) забезпечило узгодженість між модулями та уніфікований спосіб обміну даними.
3. Компоновка бібліотеки libModulesRadchenko.a свідчить про здатність перевести модулі у багаторазово використовувані компоненти, що полегшує інтеграцію в інші проєкти.
4. Застосування ifstream та ofstream забезпечило надійну обробку введення та виведення з/до файлів.
5. Впровадження перевірки відкриття файлів дозволяє своєчасно виявляти помилки введення/виведення та запобігати критичним збоям.
6. Функції, що обробляють текст (наприклад, зміна регістру або фільтрація заборонених слів), демонструють гнучке управління вхідними даними та використання стандартних засобів обробки рядків.
7. Функція, яка визначає мову тексту шляхом підрахунку латинських і кириличних символів, дозволяє програмі автоматично адаптувати подальші дії.
8. Перетворення англомовного тексту у верхній регістр підтверджує здатність системи змінювати дані відповідно до технічних вимог.
9. Реалізована функція видалення окремих слів з українських речень демонструє можливість контекстної фільтрації з урахуванням регістру.
10. Функція s\_calculation ілюструє реалізацію математичних обчислень із різними вхідними значеннями.
11. Функція convertToBinary демонструє підтримку різних числових систем, що свідчить про гнучкість розв'язку.
12. Використання таких стандартних бібліотек як , , свідчить про володіння ключовими засобами мови С++.
13. Функція логування дає змогу фіксувати вхідні параметри, результати та статус виконання тестів, що покращує контроль за якістю.
14. Тестовий драйвер, який автоматично запускає тест-кейси, підтверджує ефективність модульного підходу.
15. Формалізовані тест-кейси у вигляді константних масивів забезпечують всебічне тестування функцій за різних сценаріїв.
16. Можливість передавати параметри (імена файлів, числові значення) підвищує гнучкість виконання.
17. Опції компіляції та лінкування дозволяють інтегрувати зовнішні бібліотеки та оптимізувати код під цільову платформу.
18. Механізми створення чи очищення файлів забезпечують коректне збереження даних під час повторного запуску.
19. Отримання поточної дати й часу через функції ctime дозволяє вести журнал подій для подальшого аналізу.
20. Усі функції були реалізовані згідно з методичними рекомендаціями лабораторної роботи, що забезпечує досягнення поставлених цілей.
21. Коментарі в коді полегшують розуміння логіки реалізації та підтримку в майбутньому.
22. Попередня перевірка вхідних даних гарантує безпечне виконання операцій з файлами.
23. Збереження результатів у окремих файлах допомагає відстежувати історію виконання й полегшує налагодження.
24. Модулі проєктувались із урахуванням можливості розширення новою функціональністю та тестами.
25. Єдиний стиль написання коду та форматування підтримує чистоту структури та зменшує кількість помилок.
26. Порівняння результатів з очікуваними значеннями підвищує точність тестування й дозволяє швидко виявляти помилки.
27. Відокремлення логіки від інтерфейсу забезпечує незалежне тестування кожного модуля.
28. Збереження тестових даних у масивах сприяє повторному використанню та автоматизації перевірок.
29. Фіксація статусу проходження тестів дозволяє швидко оцінити якість окремих модулів.
30. Використання stringstream допомагає формувати структуровані й інформативні лог-повідомлення.
31. Спираючись на напрацювання з попередніх лабораторних (№8–9), вдалося забезпечити послідовність навчання та використати вже перевірені рішення.
32. Рішення можна адаптувати до практичних задач з обробки файлів, що наближує проєкт до реального застосування.
33. Обробка помилок відкриття файлів через повідомлення дозволяє швидко ідентифікувати проблему.
34. Бібліотеки , та демонструють використання сучасного інструментарію C++.
35. Рішення на основі умовних конструкцій (if-else) дозволяє враховувати різні ситуації обробки даних.
36. Застосування IDE покращує організацію проєкту, полегшує налагодження та спільну роботу над кодом.
37. Використання Git гарантує контроль версій і надійне збереження історії змін.
38. Систематичне проведення unit-тестів допомагає своєчасно виявляти помилки.
39. Виконання завдань відповідно до вимог методичних вказівок підтверджує якість реалізації.
40. Докладні коментарі роз’яснюють логіку функцій, що сприяє повторному використанню й адаптації.
41. Організація проєкту у вигляді окремих тек для коду, тестів і звітів забезпечує порядок у структурі.
42. Чітка логіка функцій дозволяє швидко перевірити правильність реалізації.
43. Параметризація обробки даних робить програму гнучкою до зміни умов виконання.
44. Підтримка обробки як текстових, так і числових даних свідчить про універсальність інструментів.
45. Архітектура системи дозволяє без зусиль додавати нові модулі.
46. Гнучка алгоритмічна логіка забезпечує адаптивну обробку вхідних даних.
47. Функції реалізовані автономно, що робить їх придатними для повторного використання в інших проектах.
48. Тестовий драйвер охоплює різноманітні типи вхідних даних, що забезпечує повноцінне тестування.
49. Автоматизація створення вхідних файлів і перевірки результатів значно зменшує обсяг ручної роботи.
50. Комплексна реалізація вимог лабораторної роботи, включно з модульністю, тестуванням, журналюванням та інтеграцією з Git, свідчить про повне досягнення поставленої мети.

**Лістинг файлу ModulesRadchenko.h**

#ifndef MODULESRADCHENKO\_H\_INCLUDED

#define MODULESRADCHENKO\_H\_INCLUDED

#include <iostream>

#include <cmath>

#include <bitset>

#include <windows.h>

#include <fstream>

#include <string>

#include <vector>

#include <algorithm>

#include <ctime>

#include <sstream>

using namespace std;

// --- Функції обробки чисел ---

float s\_calculation(int x, int y, int z);

string s\_calculation\_to\_string(int x, int y, int z);

string to\_binary(int b);

// --- Функції для задач 10.1–10.3 ---

int task\_10\_1(string path\_output, string path\_input);

int task\_10\_2(string path\_output);

int task\_10\_3(int x, int y, int z, int b, string path\_output);

// --- Допоміжні функції ---

string detect\_language(const string& sentence);

bool contains\_taboo\_words(const string& sentence);

string current\_datetime();

string read\_file(const string& path);

void append\_to\_file(const string& path, const string& content);

#endif // MODULESRADCHENKO\_H\_INCLUDED

**Лістинг статичної бібліотеки ModulesRadchenko**

#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

#include <ctime>

#include <bitset>

#include <cmath>

#include <windows.h>

using namespace std;

// Функція обчислення

float s\_calculation(int x, int y, int z) {

return pow(3 \* sin(sqrt(12 \* x + log(x - 3))), y) + (float)z / x;

}

// Перевірка чи українське речення

bool isUkrainian(const string& text) {

return text.find("і") != string::npos || text.find("ї") != string::npos || text.find("є") != string::npos;

}

// Перевірка чи англійське речення повністю у верхньому регістрі

bool isEnglishUppercase(const string& text) {

for (char c : text)

if (isalpha(c) && islower(c))

return false;

return true;

}

// Зчитування з файлу

string ReadFile(const string& filename) {

ifstream file(filename);

if (!file.is\_open()) return "ERROR: Cannot open file.";

string text, line;

while (getline(file, line))

text += line + "\n";

file.close();

return text;

}

// Запис авторських даних та обробка вхідного речення

void task\_10\_1(const string& inputPath, const string& outputPath) {

ifstream in(inputPath);

ofstream out(outputPath);

if (!in.is\_open() || !out.is\_open()) return;

// Авторські дані

out << "Радченко Максим\nЦНТУ, Кропивницький, Україна, 2025\n";

string line;

getline(in, line);

// Обробка залежно від мови

if (isUkrainian(line)) {

// Видалення "лінощі", "сесія", "академзаборгованість"

vector<string> banned = {"лінощі", "сесія", "академзаборгованість"};

for (const string& word : banned) {

size\_t pos;

while ((pos = line.find(word)) != string::npos)

line.erase(pos, word.length());

}

} else if (isEnglishUppercase(line)) {

transform(line.begin(), line.end(), line.begin(), ::toupper);

}

out << line << "\n";

in.close();

out.close();

}

// Додавання статті 62 і дати

void task\_10\_2(const string& path) {

fstream file(path, ios::app);

if (!file.is\_open()) return;

// Додаємо текст статті 62

file << "Відповідно до статті 62 Закону України \"Про вищу освіту\": особа має право здобувати вищу освіту на різних рівнях за рахунок державного або місцевого бюджету, а також за кошти фізичних (юридичних) осіб.\n";

// Дата і час

time\_t now = time(0);

char buffer[80];

strftime(buffer, 80, "%Y-%m-%d %H:%M:%S", localtime(&now));

file << "Дата та час допису: " << buffer << "\n";

file.close();

}

// Додавання результатів обчислення та числа в двійковому коді

void task\_10\_3(const string& path, int x, int y, int z, int b) {

fstream file(path, ios::app);

if (!file.is\_open()) return;

float result = s\_calculation(x, y, z);

file << "s\_calculation(" << x << ", " << y << ", " << z << ") = " << result << "\n";

bitset<32> b\_bin(b);

file << "b в двійковому коді: " << b\_bin << "\n";

file.close();

}

int main() {

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

string inputFile = "input.txt";

string outputFile = "output.txt";

// Виконання задач

task\_10\_1(inputFile, outputFile);

task\_10\_2(outputFile);

task\_10\_3(outputFile, 4, 2, 8, 7); // приклад значень x=4, y=2, z=8, b=7

cout << "Виконання завершено. Перевірте файл: " << outputFile << endl;

return 0;

}#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

#include <ctime>

#include <bitset>

#include <cmath>

#include <windows.h>

using namespace std;

// Функція обчислення

float s\_calculation(int x, int y, int z) {

return pow(3 \* sin(sqrt(12 \* x + log(x - 3))), y) + (float)z / x;

}

// Перевірка чи українське речення

bool isUkrainian(const string& text) {

return text.find("і") != string::npos || text.find("ї") != string::npos || text.find("є") != string::npos;

}

// Перевірка чи англійське речення повністю у верхньому регістрі

bool isEnglishUppercase(const string& text) {

for (char c : text)

if (isalpha(c) && islower(c))

return false;

return true;

}

// Зчитування з файлу

string ReadFile(const string& filename) {

ifstream file(filename);

if (!file.is\_open()) return "ERROR: Cannot open file.";

string text, line;

while (getline(file, line))

text += line + "\n";

file.close();

return text;

}

// Запис авторських даних та обробка вхідного речення

void task\_10\_1(const string& inputPath, const string& outputPath) {

ifstream in(inputPath);

ofstream out(outputPath);

if (!in.is\_open() || !out.is\_open()) return;

// Авторські дані

out << "Радченко Максим\nЦНТУ, Кропивницький, Україна, 2025\n";

string line;

getline(in, line);

// Обробка залежно від мови

if (isUkrainian(line)) {

// Видалення "лінощі", "сесія", "академзаборгованість"

vector<string> banned = {"лінощі", "сесія", "академзаборгованість"};

for (const string& word : banned) {

size\_t pos;

while ((pos = line.find(word)) != string::npos)

line.erase(pos, word.length());

}

} else if (isEnglishUppercase(line)) {

transform(line.begin(), line.end(), line.begin(), ::toupper);

}

out << line << "\n";

in.close();

out.close();

}

// Додавання статті 62 і дати

void task\_10\_2(const string& path) {

fstream file(path, ios::app);

if (!file.is\_open()) return;

// Додаємо текст статті 62

file << "Відповідно до статті 62 Закону України \"Про вищу освіту\": особа має право здобувати вищу освіту на різних рівнях за рахунок державного або місцевого бюджету, а також за кошти фізичних (юридичних) осіб.\n";

// Дата і час

time\_t now = time(0);

char buffer[80];

strftime(buffer, 80, "%Y-%m-%d %H:%M:%S", localtime(&now));

file << "Дата та час допису: " << buffer << "\n";

file.close();

}

// Додавання результатів обчислення та числа в двійковому коді

void task\_10\_3(const string& path, int x, int y, int z, int b) {

fstream file(path, ios::app);

if (!file.is\_open()) return;

float result = s\_calculation(x, y, z);

file << "s\_calculation(" << x << ", " << y << ", " << z << ") = " << result << "\n";

bitset<32> b\_bin(b);

file << "b в двійковому коді: " << b\_bin << "\n";

file.close();

}

int main() {

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

string inputFile = "input.txt";

string outputFile = "output.txt";

// Виконання задач

task\_10\_1(inputFile, outputFile);

task\_10\_2(outputFile);

task\_10\_3(outputFile, 4, 2, 8, 7); // приклад значень x=4, y=2, z=8, b=7

cout << "Виконання завершено. Перевірте файл: " << outputFile << endl;

return 0;

}

**Лістинг TestDrive**

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

#include <ctime>

#include <vector>

#include <algorithm>

#include <cctype>

#include "ModulesRadchenko.h"

using namespace std;

// Структура тест-кейсу для задач 10.1

struct TestCase10\_1 {

string inputFileName;

string outputFileName;

string inputContent;

string expectedOutput; // очікується точна відповідність тексту

};

// Структура тест-кейсу для задач 10.2 (порівнюємо наявність статті у тексті)

struct TestCase10\_2 {

string fileName;

string initialContent;

string expectedSubstring; // хоча б частина тексту, що має бути дописана

};

// Структура тест-кейсу для задач 10.3

struct TestCase10\_3 {

string outputFileName;

double x;

double y;

double z;

int b;

string expectedSubstring; // наприклад, результат виконання s\_calculation та двійкове представлення

};

// Функція зчитування вмісту файлу у рядок

string readFileContent(const string &fileName) {

ifstream file(fileName);

if (!file)

return "";

ostringstream ss;

ss << file.rdbuf();

return ss.str();

}

// Функція запису логів тестування

void logTestResult(ofstream &logFile, const string &testName, const string &inputArgs,

const string &obtainedResult, bool passed) {

logFile << "Test: " << testName << "\n";

logFile << "Вхідні дані: " << inputArgs << "\n";

logFile << "Отриманий результат:\n" << obtainedResult << "\n";

logFile << "Статус тесту: " << (passed ? "passed" : "failed") << "\n";

logFile << "---------------------------------------------\n";

}

int main() {

// Відкриваємо лог-файл для протоколювання тестування

ofstream logFile("test\_log.txt", ios::out);

if (!logFile) {

cerr << "Не вдалося відкрити файл test\_log.txt для запису логів.\n";

return 1;

}

// ========================

// ТЕСТИ ЗАДАЧІ 10.1

// ========================

vector<TestCase10\_1> tests10\_1 = {

{

"input\_en.txt",

"output.txt",

"Hello world!",

"Розробник: Максим Радченко, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025.\n"

"Мова речення: англійська\n"

"Оброблене речення: HELLO WORLD!\n"

},

{

"input\_ua.txt",

"output.txt",

"Привіт студенте!",

"Розробник: Максим Радченко, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025.\n"

"Мова речення: українська\n"

"Оброблене речення: Привіт студенте!\n"

}

// Додаткові приклади можна додати за потребою.

};

for (size\_t i = 0; i < tests10\_1.size(); i++) {

// Preliminary Steps: створення вхідного файлу з даними

{

ofstream inFile(tests10\_1[i].inputFileName);

inFile << tests10\_1[i].inputContent;

inFile.close();

}

// Очистка або створення вихідного файлу

{

ofstream outFile(tests10\_1[i].outputFileName);

outFile.close();

}

// Дії: виклик функції processTask10\_1

processTask10\_1(tests10\_1[i].inputFileName, tests10\_1[i].outputFileName);

// Отримання результату

string obtained = readFileContent(tests10\_1[i].outputFileName);

bool passed = (obtained == tests10\_1[i].expectedOutput);

// Протоколювання

stringstream inputArgs;

inputArgs << "inputFileName = " << tests10\_1[i].inputFileName

<< ", outputFileName = " << tests10\_1[i].outputFileName;

logTestResult(logFile, "Task10\_1 Test " + to\_string(i+1), inputArgs.str(), obtained, passed);

}

// ========================

// ТЕСТИ ЗАДАЧІ 10.2

// ========================

vector<TestCase10\_2> tests10\_2 = {

{

"input\_ua.txt",

"Привіт студенте!\n",

R"(Стаття 62. Права осіб, які навчаються у закладах вищої освіти

1. Особи, які навчаються у закладах вищої освіти, мають право на:

1) вибір форми навчання під час вступу до закладу вищої освіти;

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

3) трудову діяльність у позанавчальний час;

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;

5) безоплатне користування бібліотеками, інформаційними фондами, навчальною, науковою та спортивною базами закладу вищої освіти;

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);

7) користування виробничою, культурно-освітньою, побутовою, оздоровчою базами закладу вищої освіти у порядку, передбаченому статутом закладу вищої освіти;

8) забезпечення гуртожитком та цілодобовим доступом до нього на строк навчання у порядку, встановленому законодавством;

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

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

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

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

13) участь у громадських об’єднаннях;

14) участь у діяльності органів громадського самоврядування закладу вищої освіти, інститутів, факультетів, відділень, вченої ради закладу вищої освіти, органів студентського самоврядування;

15) вибір навчальних дисциплін у межах, передбачених відповідною освітньою програмою та 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;

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

17) академічну мобільність, у тому числі міжнародну;

18) отримання соціальної допомоги у випадках, встановлених законодавством;

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

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;

21) участь у формуванні індивідуального навчального плану;

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

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

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

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

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

27) оскарження дій органів управління закладу вищої освіти та їх посадових осіб, педагогічних і науково-педагогічних працівників;

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

2. Особи, які навчаються у закладах вищої освіти за денною формою навчання за рахунок коштів державного або місцевих бюджетів, мають право на отримання академічних та соціальних стипендій у встановленому законодавством порядку.

3. Особи, які навчаються у закладах вищої освіти за денною формою навчання, можуть отримувати інші стипендії, призначені фізичними (юридичними) особами.

4. Соціальні стипендії призначаються студентам (курсантам) закладу вищої освіти в порядку, встановленому Кабінетом Міністрів України.

Студенти (курсанти) закладу вищої освіти з числа дітей-сиріт та дітей, позбавлених батьківського піклування, а також студенти (курсанти) закладу вищої освіти, які в період навчання у віці від 18 до 23 років залишилися без батьків, мають гарантоване право на отримання соціальної стипендії, у тому числі у разі отримання академічної стипендії.

Академічні стипендії призначаються особам, які досягли значних успіхів у навчанні та/або науковій діяльності згідно з критеріями, встановленими Кабінетом Міністрів України. Частка студентів (курсантів), які мають право на отримання академічних стипендій, встановлюється вченою радою закладу вищої освіти у межах визначеного Кабінетом Міністрів України загального відсотка студентів (курсантів), які мають право на отримання академічних стипендій, та стипендіального фонду.

Студентам (курсантам) закладів вищої освіти, які мають право на отримання соціальної стипендії і набувають право на отримання академічної стипендії, надається один вид стипендії за їхнім вибором.

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

6. Для студентів (курсантів), які навчаються за гостродефіцитними спеціальностями (спеціалізаціями) (у галузях знань освіта, математичні, природничі, технічні науки), встановлюється підвищений розмір академічної стипендії. Перелік таких спеціальностей (спеціалізацій) та розмір підвищення визначаються Кабінетом Міністрів України.

{Частину сьому статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

{Частину восьму статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

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

10. Студенти, курсанти закладів вищої освіти мають право на отримання студентського квитка, зразок якого затверджується центральним органом виконавчої влади у сфері освіти і науки.)"

},

{

"input\_en.txt",

"HELLO WORLD!\n",

R"(Стаття 62. Права осіб, які навчаються у закладах вищої освіти

1. Особи, які навчаються у закладах вищої освіти, мають право на:

1) вибір форми навчання під час вступу до закладу вищої освіти;

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

3) трудову діяльність у позанавчальний час;

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;

5) безоплатне користування бібліотеками, інформаційними фондами, навчальною, науковою та спортивною базами закладу вищої освіти;

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);

7) користування виробничою, культурно-освітньою, побутовою, оздоровчою базами закладу вищої освіти у порядку, передбаченому статутом закладу вищої освіти;

8) забезпечення гуртожитком та цілодобовим доступом до нього на строк навчання у порядку, встановленому законодавством;

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

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

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

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

13) участь у громадських об’єднаннях;

14) участь у діяльності органів громадського самоврядування закладу вищої освіти, інститутів, факультетів, відділень, вченої ради закладу вищої освіти, органів студентського самоврядування;

15) вибір навчальних дисциплін у межах, передбачених відповідною освітньою програмою та 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;

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

17) академічну мобільність, у тому числі міжнародну;

18) отримання соціальної допомоги у випадках, встановлених законодавством;

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

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;

21) участь у формуванні індивідуального навчального плану;

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

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

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

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

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

27) оскарження дій органів управління закладу вищої освіти та їх посадових осіб, педагогічних і науково-педагогічних працівників;

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

2. Особи, які навчаються у закладах вищої освіти за денною формою навчання за рахунок коштів державного або місцевих бюджетів, мають право на отримання академічних та соціальних стипендій у встановленому законодавством порядку.

3. Особи, які навчаються у закладах вищої освіти за денною формою навчання, можуть отримувати інші стипендії, призначені фізичними (юридичними) особами.

4. Соціальні стипендії призначаються студентам (курсантам) закладу вищої освіти в порядку, встановленому Кабінетом Міністрів України.

Студенти (курсанти) закладу вищої освіти з числа дітей-сиріт та дітей, позбавлених батьківського піклування, а також студенти (курсанти) закладу вищої освіти, які в період навчання у віці від 18 до 23 років залишилися без батьків, мають гарантоване право на отримання соціальної стипендії, у тому числі у разі отримання академічної стипендії.

Академічні стипендії призначаються особам, які досягли значних успіхів у навчанні та/або науковій діяльності згідно з критеріями, встановленими Кабінетом Міністрів України. Частка студентів (курсантів), які мають право на отримання академічних стипендій, встановлюється вченою радою закладу вищої освіти у межах визначеного Кабінетом Міністрів України загального відсотка студентів (курсантів), які мають право на отримання академічних стипендій, та стипендіального фонду.

Студентам (курсантам) закладів вищої освіти, які мають право на отримання соціальної стипендії і набувають право на отримання академічної стипендії, надається один вид стипендії за їхнім вибором.

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

6. Для студентів (курсантів), які навчаються за гостродефіцитними спеціальностями (спеціалізаціями) (у галузях знань освіта, математичні, природничі, технічні науки), встановлюється підвищений розмір академічної стипендії. Перелік таких спеціальностей (спеціалізацій) та розмір підвищення визначаються Кабінетом Міністрів України.

{Частину сьому статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

{Частину восьму статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

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

10. Студенти, курсанти закладів вищої освіти мають право на отримання студентського квитка, зразок якого затверджується центральним органом виконавчої влади у сфері освіти і науки.)"

}

};

for (size\_t i = 0; i < tests10\_2.size(); i++) {

// Preliminary Steps: створення файлу з початковим вмістом

{

ofstream file(tests10\_2[i].fileName);

file << tests10\_2[i].initialContent;

file.close();

}

// Дія: виклик processTask10\_2

processTask10\_2(tests10\_2[i].fileName);

// Зчитування вмісту файлу

string obtained = readFileContent(tests10\_2[i].fileName);

// Перевіримо чи містить отриманий текст очікуваний підрядок

bool passed = (obtained.find(tests10\_2[i].expectedSubstring) != string::npos);

stringstream inputArgs;

inputArgs << "fileName = " << tests10\_2[i].fileName;

logTestResult(logFile, "Task10\_2 Test " + to\_string(i+1), inputArgs.str(), obtained, passed);

}

// ========================

// ТЕСТИ ЗАДАЧІ 10.3

// ========================

vector<TestCase10\_3> tests10\_3 = {

{

"output.txt", 5, 1, 2, 5,

"Результат виконання s\_calculation(5, 1, 2) = 1.20824"

// Очікуємо рядок з "Число 10 у двійковій системі числення: 1010"

},

{

"output.txt", 10, 0, -4, 1,

"Результат виконання s\_calculation(10, 0, -4) = -1.99745"

// Очікується, що "Число 0 у двійковій системі числення: 0"

}

};

for (size\_t i = 0; i < tests10\_3.size(); i++) {

// Preliminary Steps: створення або очищення вихідного файлу (якщо потрібно, з попередніми даними)

{

ofstream outFile(tests10\_3[i].outputFileName);

// Для тестування задач 10.3 можна записати попередню інформацію, якщо вона потрібна

outFile << "Попередній текст (як результат задач 10.1/10.2)\n";

outFile.close();

}

// Дія: виклик processTask10\_3

processTask10\_3(tests10\_3[i].outputFileName, tests10\_3[i].x, tests10\_3[i].y, tests10\_3[i].z, tests10\_3[i].b);

// Зчитування файлу

string obtained = readFileContent(tests10\_3[i].outputFileName);

bool passed = (obtained.find(tests10\_3[i].expectedSubstring) != string::npos);

stringstream inputArgs;

inputArgs << "outputFileName = " << tests10\_3[i].outputFileName

<< ", x = " << tests10\_3[i].x << ", y = " << tests10\_3[i].y

<< ", z = " << tests10\_3[i].z << ", b = " << tests10\_3[i].b;

logTestResult(logFile, "Task10\_3 Test " + to\_string(i+1), inputArgs.str(), obtained, passed);

}

cout << "Тестування завершено. Результати записано у файл test\_log.txt.\n";

logFile.close();

return 0;

}