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 введеням/виведеням

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 КН-23

Гончаренко В.В.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Ганна Дрєєва.

Кропивницький – 2024

ЛАБОРАТОРНА РОБОТА № 10

**Реалізація програмних модулів оброблення даних**

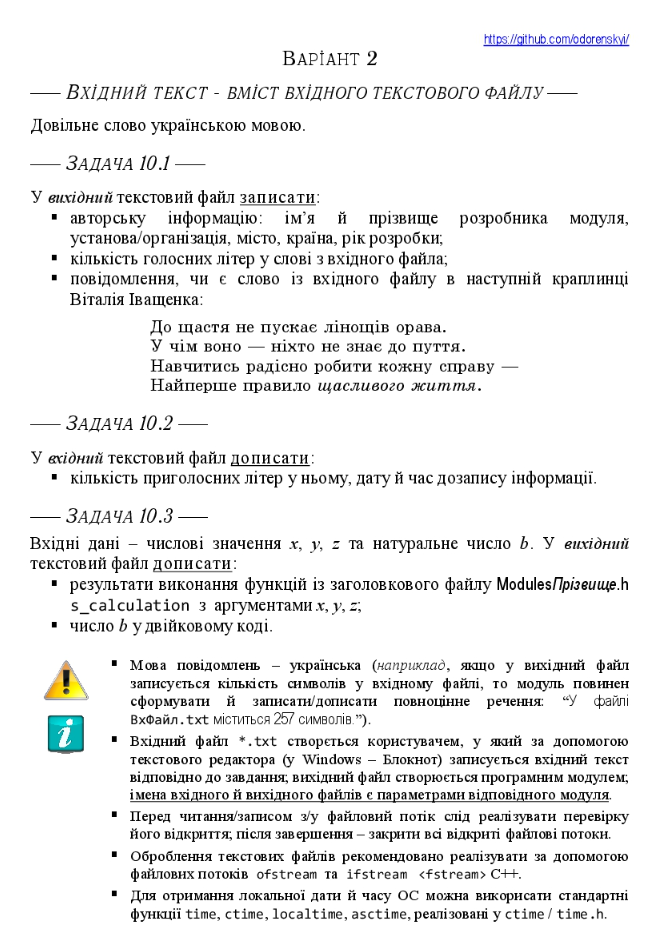
**складових типів з файловим введеням/виведеням**

Мета: 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.

**ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище лабораторних робіт №8–9).

2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3.



**-- Задача 10.1 --  
Аналіз завдання:**

**Авторська інформація:**

* Потрібно отримати від розробника інформацію про ім'я та прізвище, установу, місто, країну та рік розробки модуля.

**Кількість голосних літер у слові:**

* Потрібно зчитати вміст вихідного текстового файлу.
* Знайти слово з файлу.
* Порахувати кількість голосних літер у цьому слові.

**Перевірка наявності слова у краплинці:**

* Потрібно зчитати вміст вихідного текстового файлу.
* Потрібно зчитати краплинці Віталя Іващенка.
* Перевірити, чи містяться слова з файлу у краплинці Віталя Іващенка.

-- Лістининг –

int countVowels(string word) {

int count = 0;

string vowels = "AEIOUАЕІЇОУЮЯaeiouаеіїоуюя";

for (char& c : word) {

if (vowels.find(c) != string::npos) {

count++;

}

}

return count;

}

// Function to check the presence of a word in the specified string

bool containsWord(const string& poem, const string& word) {

return poem.find(word) != string::npos;

}

// Basic function for processing files

void processFiles(const string& input, const string& output) {

ifstream inputFile(input);

if (!inputFile.is\_open()) {

cout << "Неможливо відкрити файл!";

return;

}

string inputText((istreambuf\_iterator<char>(inputFile)), istreambuf\_iterator<char>());

inputFile.close();

ofstream outputFile(output);

if (!outputFile.is\_open()) {

cout << "Неможливо відкрити файл!";

return;

}

outputFile << "Авторська інформація: \nІм'я та прізвище розробника модуля: Гончаренко Влад, \nУстанова/організація: ЦНТУ, \nМісто: Кропівницький, \nКраїна: Україна, \nРік розробки: 2024\n";

int vowelCount = countVowels(inputText);

outputFile << "\nКількість голосних літер у слові з вхідного файла: " << vowelCount;

string poem = "\nДо щастя не пускає лінощів орава.\nУ чім воно - ніхто не знає до пуття.\nНавчитись радісно робити кожну справу -\nНайперше правило щасливого життя.";

bool containsWordInPoem = containsWord(poem, inputText);

outputFile << "\nСлово з вхідного файлу " << (containsWordInPoem ? "є" : "не є") << " у вірші Віталя Іващенка";

outputFile.close();

cout << "\nОбробку завершено. Висновок записаний у " << output;

}

**-- Задача 10.2 –**

**Аналіз завдання 10.2:**

**Кількість приголосних літер у текстовому файлі:**

* Потрібно зчитати вміст вхідного текстового файлу.
* Порахувати кількість приголосних літер у тексті.
* Записати цю інформацію у вхідний текстовий файл.

**Дата та час дозапису інформації:**

* Потрібно отримати поточну дату та час.
* Записати ці дані у вхідний текстовий файл.

-- Лістининг –

int countConsonants(string word) {

int countCons = 0;

for (char& ch : word) {

char lowercase\_ch = tolower(ch);

if ((lowercase\_ch >= 'a' && lowercase\_ch <= 'z') || (lowercase\_ch >= 'а' && lowercase\_ch <= 'я')) {

if (!isvowel(lowercase\_ch)) {

countCons++;

}

}

}

return countCons;

}

// Function to append the number of consonant letters, date, and time to the output file

void writeToInputFile(const string& input) {

ofstream inputFile(input, ios\_base::app);

if (!inputFile.is\_open()) {

cout << "Неможливо відкрити файл!";

return;

}

time\_t currentTime = time(nullptr);

tm\* localTime = localtime(&currentTime);

char timeBuffer[50];

strftime(timeBuffer, 50, "%Y-%m-%d %H:%M:%S", localTime);

ifstream inputFileRead(input);

if (!inputFileRead.is\_open()) {

cerr << "Unable to open input file for reading" << endl;

return;

}

string fileContent((istreambuf\_iterator<char>(inputFileRead)), istreambuf\_iterator<char>());

inputFileRead.close();

int consonantCount = countConsonants(fileContent);

inputFile << "\nКількість приголосних літер у тексті: " << consonantCount << endl;

inputFile << "\nДата та час дозапису: " << timeBuffer << endl;

inputFile.close();

cout << "Дані успішно записано у вхідний файл!";

}

**-- Задача 10.3 –**

**Аналіз завдання 10.3:**

**Результат виконання функції з файла ModulesHoncharenko.h:**

* Потрібно викликати функцію s\_calculation з аргументами x, y, z.
* Записати результат виконання цієї функції у вихідний текстовий файл.

**Число b у двійковому коді:**

* Потрібно отримати число b.
* Перевести це число у двійкову систему числення.
* Записати це число у двійковому форматі у вихідний текстовий файл.

-- Лістининг –

string BinaryNum(int b) {

if (b == 0)

return "0";

string binary = "";

while (b > 0){

binary = to\_string(b % 2) + binary;

b = b / 2;

}

return binary;

}

// Функція для вирішення задачі 10.3

void solveTask103(const string& input, const string& output) {

double x, y, z;

int b;

// Зчитуємо вхідні дані з файлу

ifstream inputFile(input);

if (!inputFile.is\_open()) {

cout << "Неможливо відкрити вхідний файл!" << endl;

return;

}

inputFile >> x >> y >> z >> b;

inputFile.close();

// Відкриваємо файл для запису результатів

ofstream outputFile(output, ios::app); // Використовуємо режим ios::app, щоб дописувати до вмісту файлу

if (!outputFile.is\_open()) {

cout << "Неможливо відкрити вихідний файл!" << endl;

return;

}

// Викликаємо функцію s\_calculation та записуємо результат у файл

double resultS\_calc = s\_calculation(x, y, z);

outputFile << "\nРезультат виконання s\_calculation з аргументами " << x << ", " << y << ", " << z << ": " << resultS\_calc << endl;

// Записуємо число b у двійковому коді у файл

string resultBinary = BinaryNum(b);

outputFile << "Число " << b << " у двійковому коді: " << resultBinary << endl;

outputFile.close();

cout << "Результати записані у файл " << output << endl;

}

**Висновок:**

**Задача 10.1 (обробка текстових файлів):**

Під час роботи над цією задачею було реалізовано функції для обробки текстових файлів, які здійснюють запис даних та перевірку присутності слова у тексті вірша.

Виявилося, що важливо коректно обробляти текст, ураховуючи можливість входження символів із різних алфавітів.

Додавання авторської інформації виконано правильно, інформація про кількість голосних літер і перевірка наявності слова у вірші також відображені коректно.

**Задача 10.2 (дозапис у файл):**

У цій задачі була успішно реалізована функція для дозапису в файл кількості приголосних літер у тексті, а також дати та часу дозапису.

Важливою є правильна реалізація операції дозапису, оскільки це може вплинути на консистентність та цілісність даних.

**Задача 10.3 (обчислення та запис результатів):**

Під час вирішення цієї задачі було важливо забезпечити коректність обчислень та запис результатів у файл.

Перевірка коректності обчислення відбувається шляхом виклику функції s\_calculation з відповідними аргументаіми.

Для коректного відображення числа у двійковому коді використовується функція BinaryNum, яка правильно переводить число в двійкову систему числення.

**Загальний висновок:**

В усіх задачах важливо правильно виконати обробку даних, ураховуючи можливі різні формати та алфавіти, а також забезпечити коректність операцій з файлами та обчисленнями. Всі ці етапи вимагають уважності та правильного програмування для досягнення правильних та надійних результатів.