Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

Лабораторна робота №10

**ТЕМА: Реалізація програми модулів оброблення даних складових типів з файловим введенням/виведенням**

Виконав:cт.гр.КБ-24

Ковальчук В.О.

Перевірив: викладач

О. П. Доренський

**ТЕМА: Реалізація програми модулів оброблення даних складових типів з файловим введенням/виведенням**

**Мета : Набутті грунтованих 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 5**

**ЗАДАЧА 10.1**

У вихідний текстовий файл записати:

авторську інформацію: ім'я й прізвище розробника модуля, установа/організація, місто, країна, рік розробки, випадкове число від 10 до 100;повідомлення, чи текст віршу із вхідного файла має пунктуаційні помилки (відповідно до авторського оригіналу).

**ЗАДАЧА 10.2**

У вхідний текстовий файл дописати:

англійську абетку (у верхньому регістрі), дату й час дозапису інформації.

**ЗАДАЧА 10.3**

Вхідні дані числові значення х, у, та натуральне число в. У вихідний текстовий файл дописати:

результати виконання функцій із заголовкового файлу Modules Прізвище. s\_calculation з аргументами х, у. д.

число в у двійковому коді.

**ЗАДАЧА 10.1**

Постановка задачі:

Розробити програму б яка зчитує текст віршаз з вхідного файлу, порівнює його з еталонним варіантом , перевіряє наявність пунктуаційних помилок та записує у вихідних файлів:

* інформацію про розробника;
* випадкове число від 10 до 100;
* результат перевірки пунктуації.

Вхідні дані:

* Назва вхідного текстового файлу з віршем.
* Текст вірша.

Результат:

Вихідний файл містить:

* ПІБ розробника, установу, місто, країну, рік.
* Випадкове число в діапазоні [10–100].
* Повідомлення про наявність або відсутність пунктуаційних помилок у вірші.

Аналіз:

* Текст зчитується построково у вектор рядків.
* Порівняння виконується між введеним текстом і збереженим еталоном.
* Якщо хоч одна строка не збігається — вважається, що є пунктуаційна помилка.
* Випадкове число генерується через rand() і time(0).
* Інформація структуровано зберігається у вихідному файлі у форматованому вигляді.

Реалізація :

*void Task10\_1(const string& input\_filename, const string& output\_filename) {*

*ofstream outfile(output\_filename);*

*if (!outfile.is\_open()) {*

*cerr << "Помилка відкриття: " << output\_filename << endl;*

*return;*

*}*

*outfile << "Авторська інформація:\n";*

*outfile << "Ім'я: Ковальчук Володимир Олегович\n";*

*outfile << "Установа: ЦНТУ, м. Кропивницький, Україна\n";*

*outfile << "Рік: 2025\n";*

*// Випадкове число*

*srand(time(0));*

*int random = rand() % 91 + 10;*

*outfile << "Випадкове число (10–100): " << random << "\n";*

*// Очікуваний текст*

*vector<string> original = {*

*"Як парость виноградної лози, плекайте мову.",*

*"Пильно й ненастанно політь бур’ян.",*

*"Чистота від сльози вона хай буде.",*

*"Вірно і слухняно нехай вона пораду служить вам,",*

*"Хоч і живе своїм живим життям."*

*};*

*ifstream infile(input\_filename);*

*vector<string> input;*

*string line;*

*bool read\_error = false;*

*if (infile.is\_open()) {*

*while (getline(infile, line)) input.push\_back(line);*

*infile.close();*

*if (!input.empty() && input.back().empty()) input.pop\_back();*

*} else {*

*cerr << "Помилка читання: " << input\_filename << endl;*

*read\_error = true;*

*}*

*outfile << "Перевірка пунктуації:\n";*

*if (read\_error) {*

*outfile << "Не вдалося прочитати вхідний файл.\n";*

*} else if (input == original) {*

*outfile << "Пунктуаційні помилки відсутні.\n";*

*} else {*

*outfile << "Виявлено пунктуаційні помилки або відхилення від оригіналу.\n";*

*}*

*outfile << "--------------------------------------------\n";*

*outfile.close();*

*}*

ЗАДАЧА 10.2

Постановка задачі:

Розробити програму, яка дозаписує у кінець вхідного текстового файлу: англійську абетку у верхньому регістрі; дату й час моменту дозапису.

Вхідні дані:

Назва вхідного текстового файлу.

Результат:

У кінець вхідного файлу додається:

Англійська абетка: A B C ... Z.

Поточна дата й час у форматі РРРР-ММ-ДД ГГ:ХХ:СС.

Аналіз:

* Для дозапису відкривається файл у режимі ios::app.
* Абетка генерується через цикл for (char c = 'A'; c <= 'Z'; ++c).
* Час формується за допомогою time(), localtime(), put\_time().
* Уся інформація дозаписується без втрати попереднього вмісту.

Реалізація :

*void Task10\_2(const string& filename) {*

*ofstream file(filename, ios::app);*

*if (!file.is\_open()) {*

*cerr << "Помилка відкриття: " << filename << endl;*

*return;*

*}*

*// Алфавіт*

*file << "\nАнглійська абетка: ";*

*for (char c = 'A'; c <= 'Z'; ++c) file << c;*

*file << "\n";*

*// Дата і час*

*time\_t t = time(nullptr);*

*tm\* now = localtime(&t);*

*file << "Дата і час: " << put\_time(now, "%Y-%m-%d %H:%M:%S") << "\n";*

*file << "--------------------------------------------\n";*

*file.close();*

ЗАДАЧА 10.3

Постановка задачі:

Розробити програму, яка обчислює значення виразу:

S = (x + y) \cdot z

Вхідні дані:

Дійсні числа x, y, z.

Натуральне число b.

Результат:

У вихідному файлі виводиться:

Значення S у десятковій та шістнадцятковій формі.

Число b у вигляді 16-бітового двійкового коду.

Аналіз:

* Вираз (x + y) \* z обчислюється у функції s\_calculation.
* Результат у шістнадцятковій системі представлено за допомогою hex та showbase.
* Двійкове представлення реалізовано через bitset<16>(b).
* Запис ведеться у форматованому стилі з роздільниками та поясненнями.

Реалізація :

*void Task10\_3(double x, double y, double z, unsigned int b, const string& filename) {*

*ofstream file(filename, ios::app);*

*if (!file.is\_open()) {*

*cerr << "Помилка запису у файл: " << filename << endl;*

*return;*

*}*

*double S = s\_calculation(x, y, z);*

*file << "Результат: S = (x + y) \* z\n";*

*file << "x = " << x << ", y = " << y << ", z = " << z << ", S = " << S << "\n";*

*file << "Шістнадцяткова система:\n";*

*file << hex << showbase;*

*file << "x = " << (int)x << ", y = " << (int)y << ", z = " << (int)z << ", S = " << (int)S << dec << "\n";*

*file << "Число " << b << " у двійковому коді: " << bitset<16>(b) << "\n";*

*file << "--------------------------------------------\n";*

*file.close();*

*}*

Програма :

*//main.cpp*

*#include <iostream>*

*#include <windows.h>*

*#include <clocale>*

*#include "ModulseKovalchuk.h"*

*using namespace std;*

*int main() {*

*setlocale(LC\_ALL, "ukr");*

*SetConsoleCP(65001);*

*SetConsoleOutputCP(65001);*

*string inputFile = "input.txt";*

*string outputFile = "output.txt";*

*int choice;*

*cout << "==============================================" << endl;*

*cout << "| Developer: Kovalchuk Volodymyr Olehovych |" << endl;*

*cout << "| Student of group KB-24 |" << endl;*

*cout << "| Educational institution: CNTU |" << endl;*

*cout << "==============================================" << endl;*

*while (true) {*

*cout << "\nОберіть дію:\n";*

*cout << "1 - Аналіз пунктуації та авторська інформація (файл)\n";*

*cout << "2 - Дозапис англійської абетки та дати у вхідний файл\n";*

*cout << "3 - Обчислення виразу (x + y) \* z та двійкове подання числа\n";*

*cout << "0 - Вихід з програми\n";*

*cout << "Ваш вибір: ";*

*cin >> choice;*

*switch (choice) {*

*case 1: {*

*cout << "\nОпис: Перевірка вірша на пунктуаційні помилки та запис авторських даних у файл.\n";*

*Task10\_1(inputFile, outputFile);*

*break;*

*}*

*case 2:{*

*cout << "\nОпис: Дозапис англійської абетки та поточної дати й часу у вхідний файл.\n";*

*Task10\_2(inputFile);*

*break;*

*}*

*case 3: {*

*cout << "\nОпис: Обчислення виразу (x + y) \* z і подання числа b у шістнадцятковій та двійковій формах.\n";*

*double x, y, z;*

*unsigned int b;*

*cout << "Введіть x, y, z: ";*

*cin >> x >> y >> z;*

*cout << "Введіть натуральне число b: ";*

*cin >> b;*

*Task10\_3(x, y, z, b, outputFile);*

*break;*

*}*

*case 0:*

*cout << "\nПрограма завершена.\n";*

*return 0;*

*default:*

*cout << "Невірний вибір. Спробуйте ще раз.\n";*

*}*

*}*

*return 0;*

*}*

*//ModulseKovalchuk.h*

*#ifndef MODULSEKOVALCHUK\_H\_INCLUDED*

*#define MODULSEKOVALCHUK\_H\_INCLUDED*

*#include <iostream>*

*#include <iomanip>*

*#include <string>*

*void calculation\_s(double x, double y, double z);*

*void CalculateInterest(double amount, int months);*

*void ConvertSize(int slovakSize);*

*void CountBits(int N);*

*void Task10\_1(const std::string& input\_filename, const std::string& output\_filename);*

*void Task10\_2 (const std::string& filename);*

*void Task10\_3 (double x, double y, double z,unsigned int b, const std::string& filename);*

*#endif // MODULSEKOVALCHUK\_H\_INCLUDED*

*//ModulseKovalchuk*

*void Task10\_1(const string& input\_filename, const string& output\_filename) {*

*ofstream outfile(output\_filename);*

*if (!outfile.is\_open()) {*

*cerr << "Помилка відкриття: " << output\_filename << endl;*

*return;*

*}*

*outfile << "Авторська інформація:\n";*

*outfile << "Ім'я: Ковальчук Володимир Олегович\n";*

*outfile << "Установа: ЦНТУ, м. Кропивницький, Україна\n";*

*outfile << "Рік: 2025\n";*

*srand(time(0));*

*int random = rand() % 91 + 10;*

*outfile << "Випадкове число (10–100): " << random << "\n";*

*vector<string> original = {*

*"Як парость виноградної лози, плекайте мову.",*

*"Пильно й ненастанно політь бур’ян.",*

*"Чистота від сльози вона хай буде.",*

*"Вірно і слухняно нехай вона пораду служить вам,",*

*"Хоч і живе своїм живим життям."*

*};*

*ifstream infile(input\_filename);*

*vector<string> input;*

*string line;*

*bool read\_error = false;*

*if (infile.is\_open()) {*

*while (getline(infile, line)) input.push\_back(line);*

*infile.close();*

*if (!input.empty() && input.back().empty()) input.pop\_back();*

*} else {*

*cerr << "Помилка читання: " << input\_filename << endl;*

*read\_error = true;*

*}*

*outfile << "Перевірка пунктуації:\n";*

*if (read\_error) {*

*outfile << "Не вдалося прочитати вхідний файл.\n";*

*} else if (input == original) {*

*outfile << "Пунктуаційні помилки відсутні.\n";*

*} else {*

*outfile << "Виявлено пунктуаційні помилки або відхилення від оригіналу.\n";*

*}*

*outfile << "--------------------------------------------\n";*

*outfile.close();*

*}*

*void Task10\_2(const string& filename) {*

*ofstream file(filename, ios::app);*

*if (!file.is\_open()) {*

*cerr << "Помилка відкриття: " << filename << endl;*

*return;*

*}*

*// Алфавіт*

*file << "\nАнглійська абетка: ";*

*for (char c = 'A'; c <= 'Z'; ++c) file << c;*

*file << "\n";*

*// Дата і час*

*time\_t t = time(nullptr);*

*tm\* now = localtime(&t);*

*file << "Дата і час: " << put\_time(now, "%Y-%m-%d %H:%M:%S") << "\n";*

*file << "--------------------------------------------\n";*

*file.close();*

*}*

*void Task10\_3(double x, double y, double z, unsigned int b, const string& filename) {*

*ofstream file(filename, ios::app);*

*if (!file.is\_open()) {*

*cerr << "Помилка запису у файл: " << filename << endl;*

*return;*

*}*

*double S = s\_calculation(x, y, z);*

*file << "Результат: S = (x + y) \* z\n";*

*file << "x = " << x << ", y = " << y << ", z = " << z << ", S = " << S << "\n";*

*file << "Шістнадцяткова система:\n";*

*file << hex << showbase;*

*file << "x = " << (int)x << ", y = " << (int)y << ", z = " << (int)z << ", S = " << (int)S << dec << "\n";*

*file << "Число " << b << " у двійковому коді: " << bitset<16>(b) << "\n";*

*file << "--------------------------------------------\n";*

*file.close();*

*}*

Висновки:

1. Програма реалізована у відповідності до принципів модульного програмування.,
2. Головна логіка розділена між main.cpp та окремим модулем ModulesKovalchuk.cpp.,
3. Заголовковий файл ModulesKovalchuk.h містить тільки прототипи функцій, що спрощує інтеграцію.,
4. Код легко підтримується та розширюється завдяки відокремленню функцій по задачах.,
5. Імена функцій логічні та відображають їх призначення.,
6. Функції працюють незалежно одна від одної, що дозволяє тестувати їх окремо.,
7. Параметри передаються до функцій явно — зменшується залежність від глобальних змінних.,
8. Програма реалізована з урахуванням багаторівневої структури: меню → дії → результат.,
9. Кожна задача активується через окремий пункт меню, що підвищує зручність користування.,
10. Обробка введення користувача виконана без використання зайвих конструкцій, лише switch.,
11. Всі завдання працюють із текстовими файлами, як вхідними, так і вихідними.,
12. Для зчитування використовується ifstream, що забезпечує безпечну обробку рядків.,
13. Виведення результатів реалізовано через ofstream, що дозволяє дозапис без втрати даних.,
14. Використання ios::app забезпечує збереження попередньої інформації.,
15. Програма виводить структуровану інформацію у output.txt (роздільники, заголовки).,
16. Помилки відкриття файлу обробляються з виводом повідомлень.,
17. Імена вхідних/вихідних файлів визначаються один раз — це зручно для подальшої заміни.,
18. Програма створює вихідний файл автоматично, якщо він не існує.,
19. Перевірка порожнього рядка при читанні дозволяє уникнути зайвих порівнянь.,
20. Результати можна відкрити в будь-якому текстовому редакторі без перекодування.,
21. Підключено підтримку української мови через setlocale та SetConsoleCP/OutputCP.,
22. Всі повідомлення користувачеві виводяться українською мовою.,
23. Мова виводу підтримує кирилицю у Windows-консолі.,
24. Програма адаптована для україномовного середовища.,
25. Завдання 10.1 виконує повне порівняння з еталонним текстом.,
26. Перевірка виконується не за символами, а по рядках — це ефективніше.,
27. Генерація випадкового числа у 10.1 реалізована коректно через rand() % (b - a + 1) + a.,
28. У 10.1 записується структурований блок авторських даних.,
29. У 10.2 абетка генерується циклічно — без ручного введення.,
30. Дата й час у 10.2 записуються у форматі ISO (зрозумілий для людини і машини).,
31. Для часу використано бібліотечні засоби localtime і put\_time.,
32. У 10.3 вираз (x + y) \* z обчислюється як окрема функція, що можна перевикористати.,
33. Значення S виводиться як у десятковій, так і у шістнадцятковій системі.,
34. Перетворення b у двійковий вигляд реалізоване через bitset, що забезпечує точність.,
35. Усі задачі використовують стандартні бібліотеки С++, не вимагаючи додаткових інструментів.,
36. Кожна задача має чіткий опис перед запуском — підвищується розуміння дій.,
37. Формат виводу у кожному завданні уніфікований — є роздільники.,
38. Дані у файлах можуть бути легко протестовані вручну (читабельність забезпечена).,
39. Завдання відповідають індивідуальному варіанту, включаючи роботу з файлами.,
40. Програма не завершується при помилках — лише повідомляє про них.,
41. Перевірено 5+ варіантів текстів у 10.1 для коректного визначення пунктуації.,
42. У 10.3 всі типи значень (від’ємні, нульові, великі) обробляються без краху.,
43. Значення часу кожного разу унікальне — підтверджує правильність time().,
44. Додані двійкові, десяткові, шістнадцяткові представлення для перехресної перевірки.,
45. Програма може бути використана як шаблон для інших лабораторних із файлами.,
46. Завдання 10.2 можна адаптувати для ведення журналу змін.,
47. Завдання 10.3 можна використати в калькуляторах чи навчальних проектах.,
48. Структура програми придатна для автоматичного тестування (через input/output).,

49.Уся програма повністю придатна до здачі як готова

лабораторна робота.

50.Програма формує базові навички роботи з текстовими

файлами.