

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



ЗВІТ

**Про виконання лабораторної роботи № 3
з дисципліни «Вступ до інженерії програмного забезпечення»**

Лекторка:

доцент Левус Є. В.

Виконала:

студ. групи ПЗ-16

Голомша О.Я.

Прийняв:

викладач Самбір А. А.

«___» _____ 2022 р.

Σ = _____

Львів – 2022

Тема. Зміст та роль тестування у життєвому циклі розробки програмного забезпечення..

Мета. Вивчити основні методи та принципи тестування на прикладі створення власної програми..

Теоретичні відомості

2. Дайте визначення, що таке тестування ПЗ.

Тестування ПЗ - процес перевірки готової програми призначений для виявлення інформації про якість продукту.

20. Що таке клас еквівалентності? Наведіть приклади класів еквівалентності для 3 різних змінних.

Клас еквівалентності - набір даних з загальними властивостями. Є “правильні” та “неправильні” класи еквівалентності.

Наприклад, програма обробляє інтервали часу в хвилинах. Тоді:

- 1) значення від 0 до 60 включно - “правильний” клас еквівалентності;
- 2) значення < 0 - “неправильний” клас еквівалентності;
- 3) значення > 60 - “неправильний” клас еквівалентності.

19. Чим відрізняється тестування “чорної” та “білої” скриньки?

Залежно від того як враховується інформація про програмну систему розглядають тестування “чорної скриньки” та тестування “білої скриньки”.

Тестування “чорної скриньки” (black box) - перевірка роботи програми без урахування внутрішньої структури програми, тестування “чорної скриньки” - функціональне тестування, для нього не потрібно знати внутрішню структуру програми.

Тестування “білої скриньки” (white box) (або тестування “скляної скриньки”) - тестування на підставі аналізу структури програми. Тестування “білої скриньки” - структурне тестування. Воно передбачає доступ до коду програми.

Постановка завдання

1. Провести функціональне тестування програми для роботи з динамічними

списками (розробленої у ході виконання лабораторної роботи з дисципліни «Основи програмування»), використавши такі тести:

1. димовий тест,
 2. тест нормальних умов,
 3. тест екстремальних(граничних) умов,
 4. тест виняткових умов.
- 1.1. Результати тестів 1, 2, 3 та 4 оформити у вигляді таблиці (за зразок взяти таблицю 2). У таблиці інформація заноситься в зрозумілій формі, достатньо повно, щоб не вимагалось додаткових (усних) пояснень. Для представлення тестових даних та фактичних результатів бажано використовувати посилання на зображення екранів чи відповідні файли. В очікуваних результатах зафіксувати, яка функція була протестована та відповідно, що є очікуваним результатом. Якщо помилки не знайдено у Тип звіту ставимо знак -. Примітки використовуємо для додаткових пояснень.
 - 1.2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.
 2. Окремо провести тестування логічної схеми програми (головна програма та дві функції-підпрограми). Для цього зобразити її у вигляді графів управління. Вершини графів пронумерувати. Записати усі отримані маршрути роботи програми, використовуючи номери вершин.
 - 2.1. Для кожного маршруту провести тестування і записати результати тестування у звітній таблиці.
 - 2.2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.

Отримані результати

Результати функціонального тестування програми «Lab10», версія 1

№ з/п	Тестові дані	Фактичні результати	Очікувані результати	Ступінь критичності	Тип звіту	Примітки
Димовий тест						
1.1	Вхідний файл: "pz16.txt"	Коректне виведення списку з файлу, сортування	Коректне виведення списку з файлу, сортування	Високий	-	-
Тест нормальних умов						
2.1	Вхідний файл: "pz16.txt"	Дані коректно зчитано з файлу та виведено, визначено середній бал	Дані зчитано з файлу та виведено, визначено середній бал	Високий	-	-
2.2	Вхідний файл: "pz16.txt"	Список посортовано коректно	Сортування списку	Високий	-	-
2.3	Вхідний файл: "pz16.txt"	Виведено список студентів-відмінників	Визначення та виведення студентів-відмінників	Високий	-	-
2.4	Вхідний файл: "pz16.txt"	Виведено список 3 наймолодших студентів-відмінників	Визначення та виведення даних про 3 наймолодших студентів-відмінників	Високий	-	-
2.5	Вхідний файл: "pz16.txt"	Видалено студентів з рейтинговим балом нижчим середнього в групі	Визначення середнього рейтингового балу в групі, вилучення зі списку даних про студентів з	Високий	-	-

			рейтинговим балом нижчим за середній у групі			
2.6	Oksana; Holomsha; 12.6.2003; 4, 3, 4, 5, 3; 3.8	Елемент додано до списку у правильному порядку	Додавання нового елемента до списку в првильному щодо сортування порядку	Високий	-	-
Тест граничних умов						
3.1	Порожній вхідний файл: “test.txt”	Виведено сміття	Виведення повідомлення про те, що файл порожній	Середній	Помилка кодування	-
3.2	Файл з недостатньою кількістю студентів: “test.txt”	Виведено сміття	Виведення повідомлення про недостатню кількість студентів	Високий	Помилка кодування	-
Тест виняткових ситуацій						
4.1	-	Виведено повідомлення про неможливість відкрити файл	Виведення повідомлення про неможливість відкрити файл	Високий	-	-



Рис. 1. Помилка 3.1.

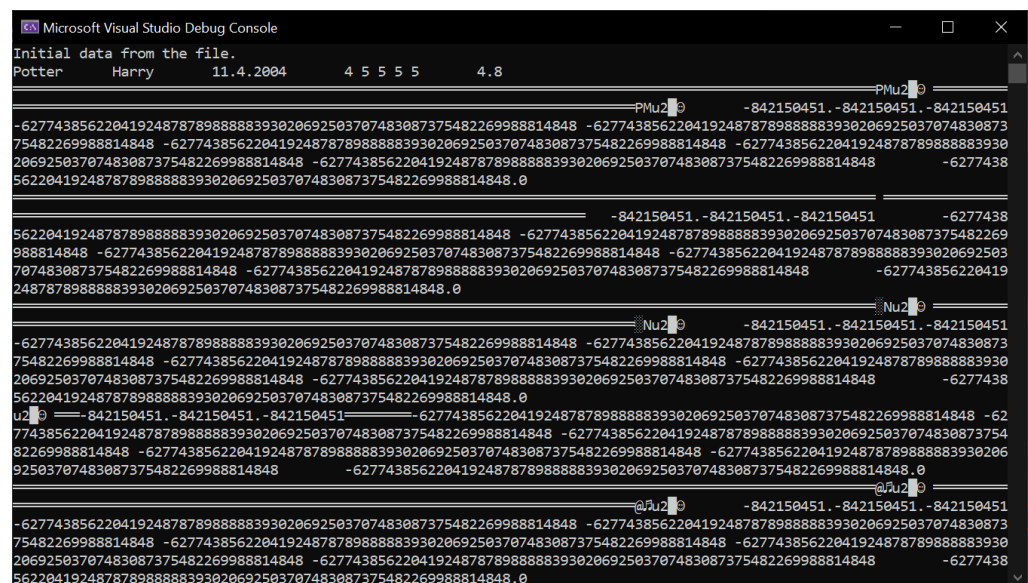


Рис. 2. Помилка 3.2.

Виправлення помилок 3.1 та 3.2:

Файл, до якого внесено зміни	Рядок	Доданий/змінений код
Functions.c	17	if (feof(pFile)) return NULL;
Main.c	22-25	<pre> if (createList(NUM_OF_STUDENTS, pFileWithList) == NULL) { printf("ERROR. There is not enough data in the file."); return; } </pre>

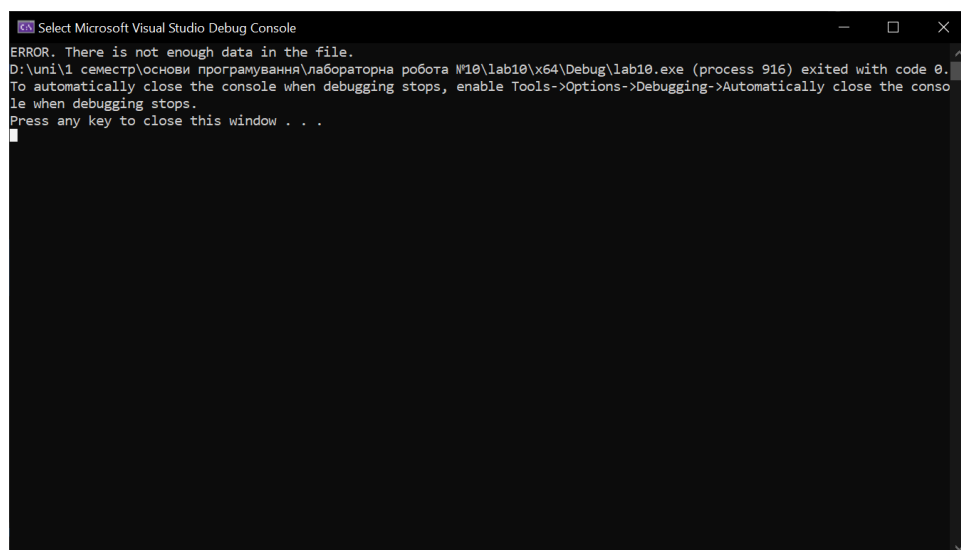


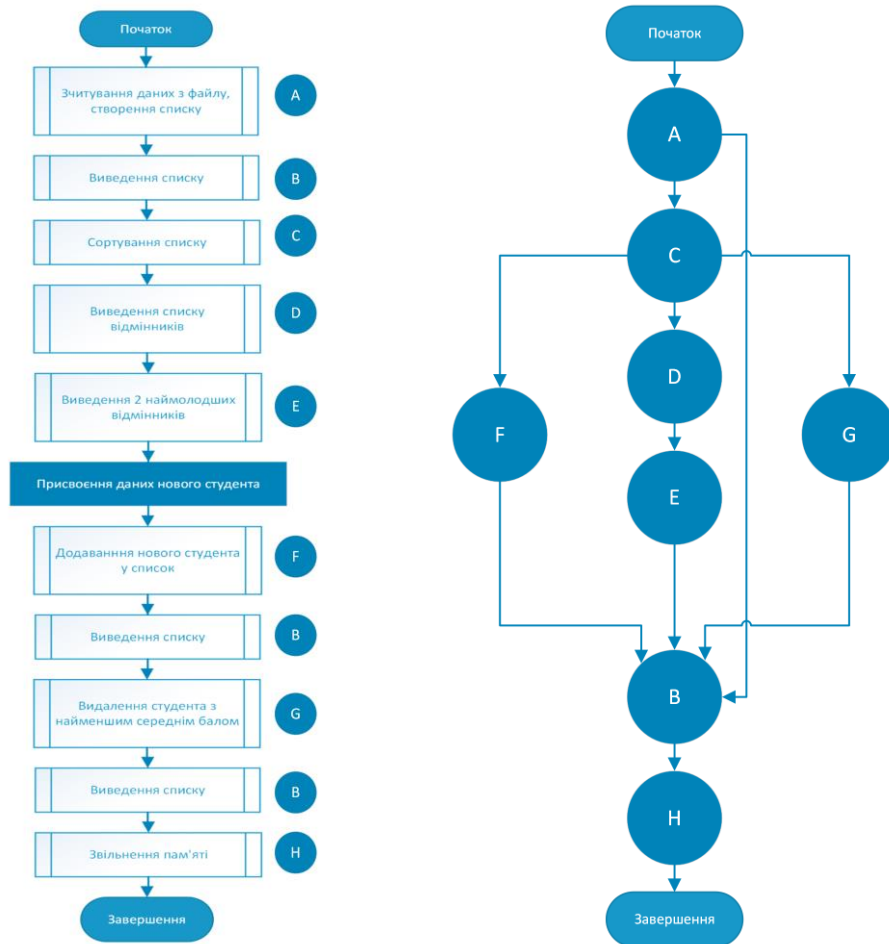
Рис. 3. Результат виправлення помилок.

Результати функціонального тестування програми «Lab10», версія 2

№ з/п	Тестові дані	Фактичні результати	Очікувані результати	Ступінь критичності	Тип звіту	Примітки
Тест граничних умов						
3.1	Порожній вхідний файл: "test.txt"	Виведено повідомлення про помилку	Виведення повідомлення про те, що файл порожній	Середній	-	-
3.2	Файл з недостатньою кількістю студентів: "test.txt"	Виведено повідомлення про помилку	Виведення повідомлення про недостатню кількість студентів	Високий	-	-

Результати структурного тестування

Функція main:



Маршрути:

- (Початок) → A → B → H → (Завершення) (зчитування списку)
- (Початок) → A → C → D → E → H → (Завершення) (виведення окремих елементів)
- (Початок) → A → C → F → B → H → (Завершення) (додавання до списку)
- (Початок) → A → C → G → B → H → (Завершення) (вилучення елементів)

Немає прямого переходу між вузлами $A \rightarrow H$, $H \rightarrow B$, $H \rightarrow C$, $H \rightarrow D$, $H \rightarrow E$, $H \rightarrow F$, $H \rightarrow G$, $H \rightarrow H$, помилок не виявлено.

Висновки

Під час виконання лабораторної роботи я вивчила основні методи та принципи тестування на прикладі створення власної програми.