НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



**ЗВІТ**

**Про виконання лабораторної роботи № 3**

**з дисципліни «Вступ до інженерії програмного забезпечення»**

**Лектор:**

доцент Левус Є. В.

**Виконала:**

студ. групи ПЗ-16

Матолінець Л.А.

**Прийняв:**

асистент Самбір А.А.

«\_\_\_» \_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2022

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

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

**Теоретичні відомості**

17. Чим відрізняється функціональне та нефункціональне тестування? Наведіть 3 приклади нефункціонального програмування.

За об’єктом тестування розрізняють функціональне та нефункціональне тестування. Функціональне полягає у перевірці, чи робота функцій програми відповідає очікуванням. Нефункціональне – стосується перевірки нефункціональних характеристик програми: надійність, певний рівень якості, вимоги до процесу розробки, відповідності стандартам, зручності тощо.

Приклади нефункціонального тестування: 1) ефективність – чи відповідає час запуску програми очікуванням (згідно ТЗ); 2) надійність – як поводиться програма при введенні некоректних даних у полях, призначений для вводу; 3) сумісність – чи працює програма однаково правильно на Windows OS та на Linux - дистрибутиві.

28. Яке тестування називають позитивним? Яке тестування називають успішним?

Позитивне тестування – перевірка програми з набором очікуваних даних та даних, які межують з ними, не виходячи за рамки допустимих значень. Успішне тестування – те, за допомогою якого вдалося виявити якнайбільше раніше не виявлених помилок.

39. Як проводиться статичний аналіз коду? Чим динамічний аналіз коду відрізняється від статичного?

Статичний аналіз коду – це уважна перевірка коду без запуску програми, що дозволяє виявити семантичні та інші помилки. Зазвичай проводиться командою програмістів та є дорогою в оплаті. Також може бути проведений статичними аналізаторами коду якими є звичайні програми. Динамічний, на противагу, це перевірка працездатності коду при його виконанні на реальному / віртуальному процесорі.

**Постановка завдання**

Провести функціональне тестування програми для роботи з

динамічними списками (розробленої у ході виконання лабораторної роботи №13 з дисципліни «Основи програмування»), використавши такі тести:

Димовий тест;

Тест нормальних умов;

Тест екстремальних (граничних) умов;

Тест виняткових умов;

Коротко описати, у чому полягає димовий тест для тестованої програми.

Результати тестів 1, 2, 3 та 4 оформити у вигляді таблиці (за зразок взяти таблицю 2). Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.

Окремо провести тестування логічної схеми програми. Для цього

зобразити її у вигляді графа управління. Вершини графа пронумерувати. Записати усі отримані маршрути роботи програми, використовуючи номери вершин. Для кожного маршруту провести тестування і записати результати тестування у звітній таблиці.

**Отримані результати**

**Результати функціонального тестування програми**

**BookList.exe, версія 1.0**

1. Димове тестування – це тип тестування ПЗ, який включає набір тестів, спрямованих на перевірку роботи найбільш важливих, критичних функцій в системі.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  з/п | Тестові дані | Фактичні результати | Очікувані результати | Ступінь критичності | Тип звіту | Примітки |
| Димовий тест | | | | | | |
| 1 | Запуск програми | Користувачу одразу дається можливість ввести дані про нову книгу | Програма запускається успішно та повідомляє користувача про початок роботи програми, і після цього користувач може ввести дані про нову книгу | Високий | Пропозиція | Реалізувати сповіщення у командному рядку про початок роботи програми та після цього дати користувачу ввести дані про книгу |
| 2 | Відкриття файлу “new.txt” | Файл успішно відкривається після запуску програми | Файл успішно відкривається після запуску програми | Низький | Пропозиція | Реалізувати повідомлення про відкриття, файлу |
| Тест нормальних умов | | | | | | |
| 1 | Додавання елементу в список після запуску програми | Елемент додається в кінець списку та відбувається його вивід | Програма додає елемент у кінець списку та виводить невідфільтрований список | Середній | Пропозиція | Реалізувати повідомлення про успішне додавання елементу в список |
| 2 | Вивід середньої кількості сторінок всіх книг. Вивід назв книг, які вміщують більше сторінок, ніж середня кількість | Відбувається вивід середньої кількості сторінок та книг, кількість сторінок яких є більшою | Програма успішно  виводить середню кількість сторінок та книг, кількість сторінок яких є більшою | -- | -- | -- |
| 3 | Вивід початкового та відфільтрованого списку | Дії відбуваються згідно очікувань | Програма успішно виводить початковий та відфільтрований список | -- | -- | -- |
| 4 | Видалення елементів, кількість сторінок яких менша за 90 та рік видання якого менший за 1980 | Дії відбуваються згідно з очікуваннями | Програма успішно виводить список, видаливши елементи, кількість сторінок яких менша за 90 та рік видання якого менший за 1980 | Середній | Пропозиція | У повідомленні про видалення елемнтів слід вказати критерії, за якими елементи були видалені |
| 5 | Запис всього списку в файл | Дії відбуваються згідно з очікуваннями | Програма успішно записує список у файл | -- | -- | -- |
| Тест екстремальних умов | | | | | | |
| 1 | У полі імені та прізвища автора, назви книги вписуються 40 символів, у другому випадку – 1 символ | Програма зупиняє компіляцію при 40 символах, не повідомляючи про те, що відбулося не так | Програма правильно компілює дані, які входять в проміжок від 1 до 40. В іншому випадку – повідомляє про помилку | Високий | Помилка кодування | -- |
| 2 | У полі введення чисельних даних (рік публікування, ціна та кількість сторінок) вводяться нулі | Програма приймає ці дані та продовжує формувати список | Програма повідомляє про некоректні дані та пропонує ввести їх знову | Високий | Помилка кодування | -- |
| Тест виняткових ситуацій | | | | | | |
| 1 | Спроба додати у список елемент з прізвищем/іменем, яке має в складі цифри | Програма зчитує й опрацьовує введені дані | Програма повідомляє про некоректні дані та пропонує ввести їх знову | Високий | Помилка кодування | -- |
| 2 | Спроба ввести пропуски у всі текстові поля | Програма зчитує пусті рядки й продовує компіляцію | Програма повідомляє про некоректні дані та пропонує ввести їх знову | Високий | Помилка кодування |  |
| 3 | Спроба ввести у текстові поля рядки, кількість символів у яких більша за 40 | Програма зупиняє компіляцію, не повідомляючи про помилку | Програма повідомляє про некоректні дані та пропонує ввести їх знову | Середній | Пропозиція | Створити вивід, який пояснює користувачу його помилку |
| 4 | У поля ціна/рік/кількість сторінок вводяться від’ємні числа | Програма зчитує й опрацьовує введені дані | Програма повідомляє про некоректні дані та пропонує ввести їх знову | Високий | Помилка кодування | -- |

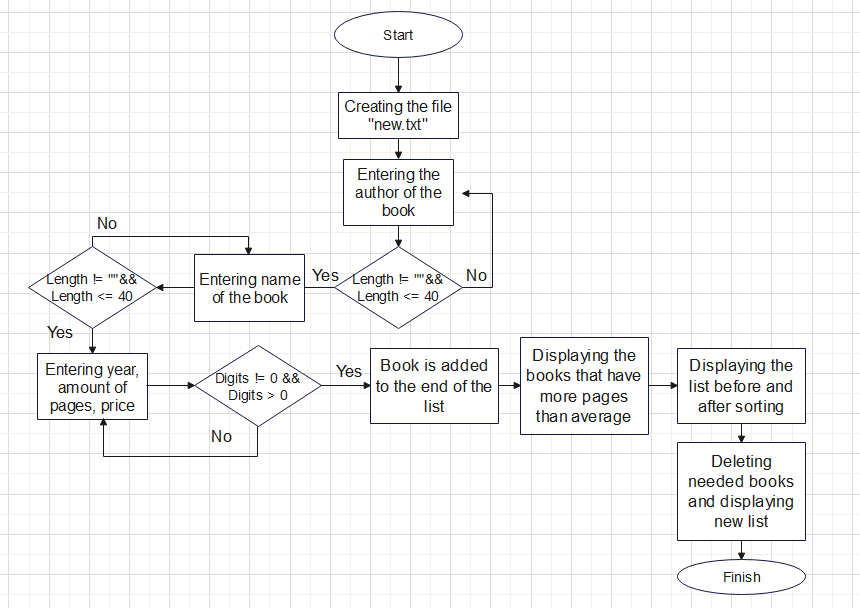
**Результати функціонального тестування програми**

**BookList.exe, версія 2.0**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  з/п | Тестові дані | Фактичні результати | Очікувані результати | Ступінь критичності | Тип звіту | Примітки |
| Димовий тест | | | | | | |
| 1 | Запуск програми | Користувача сповіщають про початок роботи програми та надається можливість ввести дані його книги | Програма запускається успішно та повідомляє користувача про початок роботи програми, і після цього користувач може ввести дані про нову книгу | -- | -- | -- |
| 2 | Відкриття файлу “new.txt” | Файл успішно відкривається після запуску програми, все це супроводжується відповідним повідомленням | Файл успішно відкривається після запуску програми | -- | -- | -- |
| Тест нормальних умов | | | | | | |
| 1 | Додавання елементу в список після запуску програми | Елемент додається в кінець списку та відбувається його вивід, супроводжується повідомленням про успішне додавання елементу | Програма додає елемент у кінець списку, сповіщає про успішне додавання та виводить невідфільтрований список | -- | -- | -- |
| 2 | Вивід середньої кількості сторінок всіх книг. Вивід назв книг, які вміщують більше сторінок, ніж середня кількість | Відбувається вивід середньої кількості сторінок та книг, кількість сторінок яких є більшою | Програма успішно  виводить середню кількість сторінок та книг, кількість сторінок яких є більшою | -- | -- | -- |
| 3 | Вивід початкового та відфільтрованого списку | Дії відбуваються згідно очікувань | Програма успішно виводить початковий та відфільтрований список | -- | -- | -- |
| 4 | Видалення елементів, кількість сторінок яких менша за 90 та рік видання якого менший за 1980 | Дії відбуваються згідно з очікуваннями | Програма успішно виводить список, видаливши елементи, кількість сторінок яких менша за 90 та рік видання якого менший за 1980, програма повідомляє, згідно з якими критеріями були видалені книги | -- | -- | -- |
| 5 | Запис всього списку в файл | Дії відбуваються згідно з очікуваннями | Програма успішно записує список у файл | -- | -- | -- |
| Тест екстремальних умов | | | | | | |
| 1 | У полі імені та прізвища автора, назви книги вписуються 40 символів, у другому випадку – 1 символ | Дії відбуваються згідно з очікуваннями | Програма повідомляє користувача про помилку та пропонує ввести дані знову | -- | -- | -- |
| 2 | У полі введення чисельних даний (рік публікування, ціна та кількість сторінок) вводяться нулі | Дії відбуваються згідно з очікуваннями | Програма повідомляє про некоректні дані та пропонує ввести їх знову | -- | -- | -- |
| Тест виняткових умов | | | | | | |
| 1 | Спроба додати у список елемент з прізвищем/іменем, яке має в складі цифри | Програма поводиться згідно з очікуваним результатом | Програма повідомляє про некоректні дані та пропонує ввести їх знову | -- | -- | -- |
| 2 | Спроба ввести пропуски у всі текстові поля | Програма поводиться згідно з очікуваним результатом | Програма повідомляє про некоректні дані та пропонує ввести їх знову | -- | -- |  |
| 3 | Спроба ввести у текстові поля рядки, кількість символів у яких більша за 40 | Програма поводиться згідно з очікуваним результатом | Програма повідомляє про некоректні дані та пропонує ввести їх знову | -- | -- | -- |
| 4 | У поля ціна/рік/кількість сторінок вводяться від’ємні числа | Програма поводиться згідно з очікуваним результатом | Програма повідомляє про некоректні дані та пропонує ввести їх знову | -- | -- | -- |

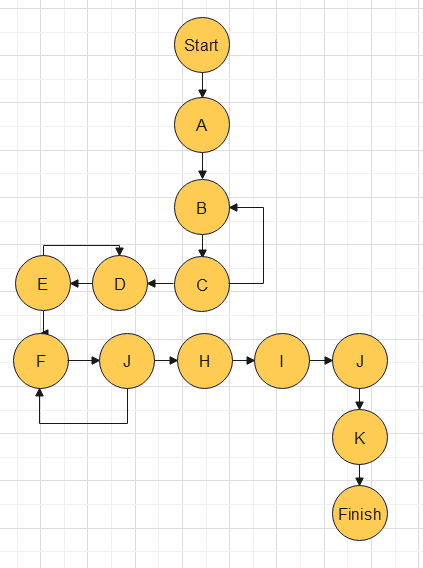
**Тестування логічної схеми програми**

1. **Структурне тестування**

****

*Рис. 1 Блок – схема роботи програми*

1. **Граф управління**

****

*Рис. 2 Граф управління програми*

|  |  |  |
| --- | --- | --- |
| № | Маршрут | Результат |
| 1 | [Start]-А-B-C-D-E-F-G-H-I-J-K-L-[Finish] | Додавання до кінця списку нової книги та вивід посортованого списку |
| 2 | [Start]-А-B-C-B-C-D-E-F-G-H-I-J-F-J-K-L-[Finish] | Додавання до кінця списку нової книги та вивід посортованого списку |

**Висновок**

На цій лабораторній роботі я освоїла основи тестування, провела димовий тест, тест нормальних, екстремальних та виняткових умов. За результатами цього тестування я виправила помилки в коді. Також за отриманим кодом я склала граф управління програми та можливі road maps користувача під час використання програми.