МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет “Львівська політехніка”

Інститут **ІКНІ**

Кафедра **ПЗ**

**ЗВІТ**

До лабораторної роботи № 3

На тему: “*Зміст та роль тестування у життєвому циклі програмного забезпечення”*

З дисципліни : “*Вступ до інженерії програмного забезпечення* ”

**Лекторка:**

ст. викл. каф. ПЗ

Левус Є.В.

**Виконала:**

студ. групи ПЗ-16

Гук А.М.

**Прийняв**:

асист. каф. ПЗ

Cамбір А.А.

“ ” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Львів-2022

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

**Мета.** Вивчити та закріпити основні методи тестування на основі власної програми.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

***7.*** *Назвіть нульові приклади для різних типів даних.*

Для числового вводу – це звичайні нульові значення, для послідовності символів – це ланцюжок пропусків, для вказівників – нульове значення адреси.

**22.** *Що таке регресійне тестування? Як санітарне тестування пов’язано з регресійним?*

Санітарне тестування або перевірка узгодженості – вузьконаправлене тестування, достатнє для доведення того, що конкретна функція працює згідно заявлених у специфікації вимог. Таке тестування є підмножиною регресійного тестування і використовується для визначення працездатності певної частини програми після змін, вироблених у ній або навколишньому середовищі. Регресійне тестування орієнтоване на повторне вибіркове тестування системи або її 7 компонентів після внесення в них змін на тих самих тестах, що і до модифікації.

**32*.*** *Яке тестування називають формальним, а яке неформальним?*

Згідно з тим, чи відбувається повний опис (документування) тестування, розрізняють види тестування – формальне чи неформальне тестування.

**Постановка завдання**

1. Провести функціональне тестування програми для роботи з динамічними списками, використавши такі тести:
2. димовий тест,
3. тест нормальних умов,
4. тест екстремальних(граничних) умов,
5. тест виняткових умов.
   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 вимагалося 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.
   2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.
6. 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.
   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**

**ListOfStudents, версія 1.0**

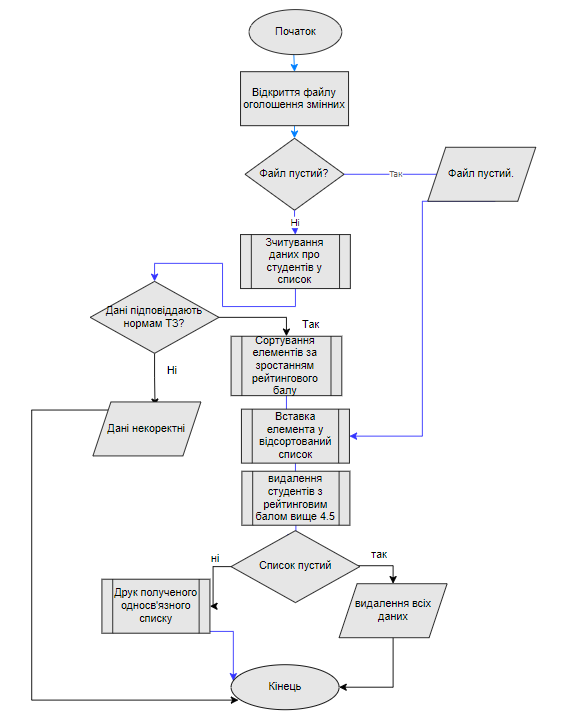
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| №  з/п | Тестові дані | | Фактичні результати | Очікувані результати | Ступінь критичності | Тип звіту | Примітки |
| **Димовий тест** | | | | | | |  |
| 1 | Запуск програми . | | Вивід зчитаного списку. | Успішне зчитування списку та виведення виведення на консоль | Високий | - | - |
| **Тест нормальних умов** | | | | | | |  |
| 1 | Додавання нової книги до посортова-ного списку | Вивід зчитаного списку. Потім списку з новим елементом | | Вивід зчитаного списку,потім списку з новим елементом | Високий | - | Список сортується за зростанням рейтингово-го балу |
| 2 | Видалення зі списку студентів з рейтинговим балом вищим за 4.5 | Вивід на консоль відкоректовано-го списку | | Вивід на консоль відкоректованого списку | Середній | - | Рейтинго-вий бал- це сума оцінок за екзамени розділені на кількість екзаменів та помножений на спеціальний коефіцієнт |
| **Тест граничних умов** | | | | | | |  |
| 1 | Спроба зчитати дані з порожнього файлу | | Не виведено жодної інформації про стан файлу | Вивід повідомлення про те що файл є порожнім. | Високий | Помилка кодування |  |
| 2 | Додавання студента з лише оцінками “1” | | Успішне додавання студента до списку | Успішне додавання студента до списку | Низький | - |  |
| 3 | Додавання студента у порожній список | | Успішне додавання студента до списку | Успішне додавання студента до списку | Середній | - |  |
| 4 | Видалення всіх студентів | | Не виведено жодної інформації про стан списку | Вивід повідомлення про видалення всіх студетів | Середній | Помилка кодування |  |
| **Тест виняткових умов** | | | | | | |  |
| 1 | Спроба зчитати дані з файлу, не існує. | | Список не зміниться | Список не зміниться | Середній | - |  |
| 2 | Ведення некоректних  даних | | Підставлення цих даних в список та подальше опрацьову-  вання некоректних даних | Вивід повідомлення про некоректність введених даних. | Низький | Помилка кодування |  |

**Результати функціонального тестування**

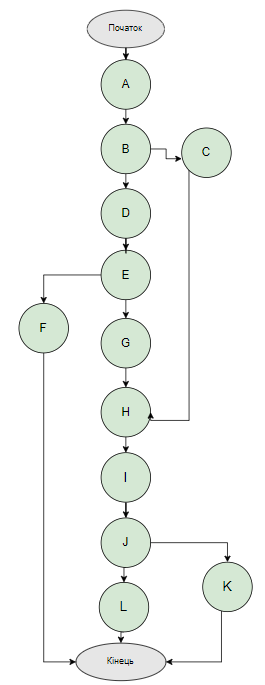
**ListOfStudents, версія 2.0**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  з/п | Тестові дані | Фактичні результати | Очікувані результати | Ступінь критичності | Тип звіту | Примітки |
| **Димовий тест** | | | | | |  |
| 1 | Запуск програми . | Вивід зчитаного списку. | Успішне зчитування списку та виведення виведення на консоль | Високий | - | - |
| **Тест нормальних умов** | | | | | |  |
| 1 | Додавання нової книги до списку до посортовано- го списку | Вивід зчитаного списку.Потім списку з новим елементом | Вивід зчитаного списку,потім списку з новим елементом | Високий | - | Список сортується за зростанням рейтингово-го балу |
| 2 | Видалення зі списку студентів з рейтинговим балом вищим за 4.5 | Вивід на консоль відкоректованого списку | Вивід на консоль відкоректованого списку | Середній | - | Рейтинго-вий бал- це сума оцінок за екзамени розділені на кількість екзаменів та помножений на спеціальний коефіцієнт |
| **Тест граничних умов** | | | | | |  |
| 1 | Спроба зчитати дані з порожнього файлу | Вивід повідомлення про те що файл є порожнім | Вивід повідомлення про те що файл є порожнім. | Високий | - |  |
| 2 | Додавання студента з лише оцінками “1” | Успішне додавання студента до списку | Успішне додавання студента до списку | Низький | - |  |
| 3 | Додавання студента у порожній список | Успішне додавання студента до списку | Успішне додавання студента до списку | Середній | - |  |
| 4 | Видалення всіх студентів | Вивід повідомлення про видалення всіх студетів | Вивід повідомлення про видалення всіх студетів | Середній | - |  |
| **Тест виняткових умов** | | | | | |  |
| 1 | Спроба відкрити файл, шлях до якого не існує. | Виведення повідомлення, що такого файлу не існує | Виведення повідомлення, що такого файлу не існує | Середній | - |  |
| 2 | Ведення некоректних  даних | Вивід повідомлення про некоректність введених даних. | Вивід повідомлення про некоректність введених даних. | Низький | - |  |

**Результат структурного тестування**



**Граф управління**



**Можливі шляхи обходу:**

**1)** [початок]-[А]-[B]-[C]-[H]-[I]-[J]-[K]-[кінець] – *вставка єдиного елементу до списку та видалення його за умовою, вивід повідомлення про видалення всіх елементів списку.*

**2)**  [початок]-[А]-[B]-[D]-[E]-[F]-[кінець] – *зчитування некоректних даних з файлу, вівід повідомлення про некоректність даних та вихід з програми .*

**3)**  [початок]-[А]-[B]-[D]-[E]-[G]-[H]-[I]-[J]-[K]-[кінець] - *зчитування коректних даних з файлу, їх сортування та видалення всіх за умовою, вивід повідомлення про видалення всіх елементів списку.*

**4)**  [початок]-[А]-[B]-[D]-[E]-[G]-[H]-[I]-[J]-[L]-[кінець] - *зчитування коректних даних з файлу, їх сортування та видалення всіх за умовою, друк відредагованого списку.*

**5)** [початок]-[А]-[B]-[C]-[H]-[I]-[J]-[L]-[кінець] – *вставка єдиного елементу до списку та видалення його за умовою, вивід списку.*

**Висновок**: під час виконання цієї лабораторної роботи я вивчила основи тестування на основі власної програми, переробила програму, засвоїла шляхи усунення можливих помилок, які можна буде використовувати для своїх майбутніх програм, в яких теж можуть виникати помилки, які можна уникнути за допомогою тестування.