

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет “Львівська політехніка”

Інститут ІКНІ  
Кафедра ПЗ



**ЗВІТ**

До лабораторної роботи № 3

З дисципліни : “Вступ до інженерії програмного забезпечення ”

**Лектор:**

ст. викл. каф. ПЗ

Левус Є.В.

**Виконав:**

ст. ПЗ-15

Марущак А.С.

**Прийняв:**

ассист. каф. ПЗ

Самбір А. А.

“ ”

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**

#### **14. Поясніть, що означає думати про тестування на етапі визначення та аналізу вимог**

Це означає, що ще при створенні всіх вимог до програми треба завчасно думати про те, чи ці вимоги можна буде протестувати. Бо, наприклад вимогу «Програма має мати красивий інтерфейс» об'єктивно не можна верифікувати.

Саме тому, всі вимоги, що містяться в ТЗ(або іншому документі, що відображає вимоги до ПЗ) мають бути верифіковними.

#### **21. Як визначити повноту структурного тестування?**

Треба удостовіритись в тому, що тестові дані змушують працювати програму у всіх логічних відгалуженнях коду. Тобто, умовно кажучи, треба пройти всіма стежками, що передбачені кодом програми.

#### **47. Що означає властивість живучість ПЗ? Як її можна перевірити?**

Живучість – здатність системи функціонувати при вводі недопустимих даних або ж у напружених умовах. Задля її перевірки треба включити у тестові дані якомога більше неправильних та крайніх випадків.

### **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.









## Виконання роботи











## 1. Результати функціонального тестування:












## Результати функціонального тестування програми








# LibraryUtil, версія 0.0.1









<b>№ з/п</b>	<b>Тестові дані</b>	<b>Фактичні результати</b>	<b>Очікувані результати</b>	<b>Ступінь критичності</b>	<b>Тип звіту</b>	<b>Примітки</b>
<b>Димове тестування</b>						
<b>1</b>	-	Запуск програми без помилок	Запуск програми без помилок	-	-	-
<b>2</b>	-	Завершення роботи без помилок	Завершення роботи без помилок	-	-	-
<b>Тест нормальних умов</b>						

1	 randomCommandsTest.exe	<p>Нормальна робота команд. Але при наявності крапки з комою в кінці стрічки вважає його за порожню команду</p>  randomCommandsTestResult.txt	Правильна обробка скриптових файлів з різними функціями	Не критично	Помилка кодування	Помилку викликає наявність пробілу після крапки з комою.
2	 readingFile.exe	<p>Правильне зчитування правильних вхідних даних з файлу</p>  readingFileResult.txt	Правильне зчитування правильних вхідних даних з файлу	-	-	-
3	 sortingList.exe	<p>Правильне сортування правильних даних за всіма ознаками</p>  sortingListResult.txt	Правильне сортування правильних даних за всіма ознаками	-	-	-
4	 filteringList.exe	<p>Правильна фільтрація правильних даних за всіма ознаками</p>  filteringListResult.txt	Правильна фільтрація правильних даних за всіма ознаками	-	-	-





5	 limitList.lsexe	Правильне обмеження списку до певної к-ті книг  limitListResult.txt	Правильне обмеження списку до певної к-ті книг	-	-	-
6	 separateLists.lsexe	Створення декількох списків паралельно  separateListsResult.txt	Створення декількох списків паралельно	-	-	-
7	 switchingLists.lsexe	Перемикання між списками книг за звичайних умов  separateListsResult.txt	Перемикання між списками книг за звичайних умов	-	-	-
8	 appendToList.lsexe	Правильне додавання книг вручну  appendToListResult.txt	Правильне додавання книг вручну	-	-	-
9	 listOfLists.lsexe	Вивід списку доступних списків книг  listOfLists.txt	Вивід списку доступних списків книг	-	-	-

10	 switchingLists.lsexe	Вивід к-ті книг у поточному списку  switchingListsResult.txt	Вивід к-ті книг у поточному списку	-	-	-
11	 saveToFile.lsexe	Збереження списку у текстовий файл  result.txt  saveToFileResult.txt	Збереження списку у текстовий файл	-	-	-
12	 listDeletion.lsexe	Помилка при видаленні одного порожнього списку	Нормальне видалення списків	Критична	Помилка кодування	Якщо списки не порожні, то проблеми не виникає
13	 helpTest.lsexe	Отримання довідки по командам оболонки  helpTestResult.txt	Отримання довідки по командам оболонки	-	-	-
<b>Тест екстремальних(граничних) умов</b>						
1	 emptyFilesTest.lsexe  empty.txt  whitespaces.txt	Нормальне опрацювання порожніх текстових файлів(або з порожніми стрічками) з виводом відповідного	Нормальне опрацювання порожніх текстових файлів(або з порожніми стрічками) з виводом відповідного			

		повідомлення  emptyFilesTestResult.txt	повідомлення			
2	 emptyFilesTest.lsexe   largefile.txt	Нормальне опрацювання великих за обсягом текстових файлів   emptyFilesTestResult.txt	Нормальне опрацювання великих за обсягом текстових файлів			
3	 sortEmptyList.lsexe	Помилки не виникає, проте слід додати повідомлення про сортування порожнього списку   sortEmptyListResult.txt	Сортування порожнього списку з виводом відповідного повідомлення	Не критично	Пропозиц ія	-
4	 filterEmptyList.lsexe	Помилки не виникає, проте слід додати повідомлення про фільтрацію порожнього списку	Фільтрація порожнього списку з виводом відповідного повідомлення	Не критично	Пропозиц ія	-









		 filterEmptyListResult.txt				
5	 limitEmptyList.lsexe	Обмеження порожнього списку з виводом відповідного повідомлення   limitEmptyListResult.txt	Обмеження порожнього списку з виводом відповідного повідомлення	-	-	-
<b>Тест критичних умов</b>						
1	 invalidCommandTest.lsexe	Нормальна обробка неправильної команди в скріпті з виводом відповідного повідомлення   invalidCommandTestResult.txt	Нормальна обробка неправильної команди в скріпті з виводом відповідного повідомлення			
2	 invalidParametersTest.lsexe	Нормальна обробка неправильного параметру команди з виводом відповідного повідомлення   invalidParametersTestResult.txt	Нормальна обробка неправильного параметру команди з виводом відповідного повідомлення			
3	 invalidParametersCount.lsexe	Помилка при передачі 2ох параметрів в	Нормальна обробка неправильної к-	Критично	Помилка кодування	-



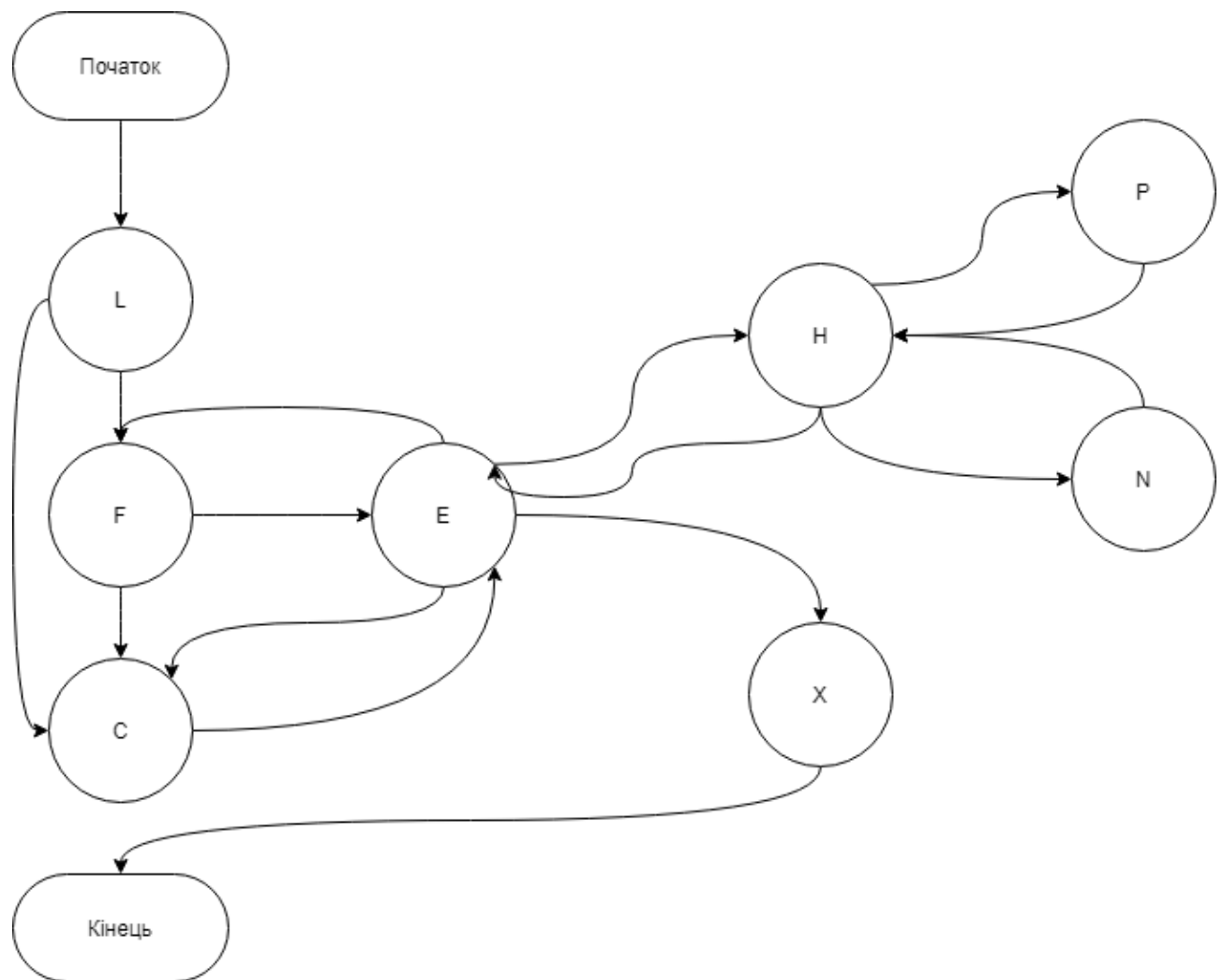
		функцію сортування	ті параметрів команди скріпта з виводом відповідного повідомлення			
4	 notExistingScriptFileTest.PNG	Нормальна обробка запуску неіснуючого скриптового файлу з виводом відповідного повідомлення   notExistingScriptFileTestResult.PNG	Нормальна обробка запуску неіснуючого скриптового файлу з виводом відповідного повідомлення			
5	 notExistingTextFileTest.lsexe	Нормальна обробка зчитування з неіснуючого текстового файлу з виводом відповідного повідомлення   notExistingFileTestResult.txt	Нормальна обробка зчитування з неіснуючого текстового файлу з виводом відповідного повідомлення			

### Результати повторного тестування

№ з/п	Тестові дані	Фактичні результати	Очікувані результати	Ступінь критич ності	Тип звіту	Примітки
1	 randomCommandsTest.lsexe	Правильна обробка скриптових файлів з різними	Правильна обробка скриптових файлів з різними функціями	-	-	-

		функціями  randomCommandsTestResult.txt				
2	 listDeletion.lsexe	Нормальне видалення списків  listDeletionResult.txt	Нормальне видалення списків	-	-	-
3	 sortEmptyList.lsexe	Сортування порожнього списку з виводом відповідного повідомлення  sortEmptyListResult.txt	Сортування порожнього списку з виводом відповідного повідомлення	-	-	-
4	 filterEmptyList.lsexe	Фільтрація порожнього списку з виводом відповідного повідомлення  filterEmptyListResult.txt	Фільтрація порожнього списку з виводом відповідного повідомлення	-	-	-
5	 invalidParametersCount.lsexe	Нормальна обробка неправильної к-ті параметрів команди скріпта з виводом відповідного повідомлення	Нормальна обробка неправильної к-ті параметрів команди скріпта з виводом відповідного повідомлення	-	-	-

### Логічна схема програми у вигляді графів управління



L – очистка екрану

F – виконання скриптового файлу

C – виконання в консолі

E – виконання однієї команди

H – отримання довідки по команді

P – існуюча команда

N – неіснуюча команда

X – вихід з програми

Результати структурного тестування

№ з/п	Маршрут	Результат
1	(Початок)-L-F-E-X-(Кінець)	Успішне виконання
2	(Початок)-L-F-E-H-P-E-X- (Кінець)	Успішне виконання
3	(Початок)-L-F-E-H-N-E-X- (Кінець)	Успішне виконання
4	(Початок)-L-C-E-X-(Кінець)	Успішне виконання
5	(Початок)-L-C-E-H-P-E-X- (Кінець)	Успішне виконання
6	(Початок)-L-C-E-H-N-E-X- (Кінець)	Успішне виконання
7	(Початок)-L-F-E-F-C-E-X- (Кінець)	Успішне виконання
8	(Початок)-L-F-E-F-C-E-H-P- E-X-(Кінець)	Успішне виконання
9	(Початок)-L-F-E-F-C-E-H-N- E-X-(Кінець)	Успішне виконання
10	(Початок)-L-F-E-H-P-E-F-C- E-X-(Кінець)	Успішне виконання
11	(Початок)-L-F-E-H-P-E -F-C- E-H-P-E-X-(Кінець)	Успішне виконання
12	(Початок)-L-F-E-H-P-E -F-C- E-H-N-E-X-(Кінець)	Успішне виконання
13	(Початок)-L-F-E-H-N-E-F-C- E-X-(Кінець)	Успішне виконання
14	(Початок)-L-F-E-H-N-E -F-C- E-H-P-E-X-(Кінець)	Успішне виконання
15	(Початок)-L-F-E-H-N-E -F-C- E-H-N-E-X-(Кінець)	Успішне виконання

**Висновок:** Виконавши цю лабораторну роботу, ми вивчили основні методи та принципи тестування на прикладі створенної власної програми. Мною особисто було знайдено декілька критичних і не критичних помилок, тому тестування можна вважати успішним.