МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет "Львівська політехніка"

Інститут ІКНІ Кафедра ПЗ



3BIT

До лабораторної роботи № 3 3 дисципліни : "Вступ до інженерії програмного забезпечення "

Лектор: ст. викл. каф. ПЗ

Левус €.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 створені 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 пройтися 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 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.
- 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

№ 3/п	Тестові дані	Фактичні результати	Очікувані результати	Ступінь критич ності	Тип звіту	Примітки	
	Димове тестування						
1	-	Запуск програми без помилок	Запуск програми без помилок	-	-	-	
2	-	Завершення роботи без помилок	Завершення роботи без помилок	-	-	-	
	Тест нормальних умов						

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

_	4	Правильне	Правильне	_	_	_
5		обмеження	обмеження			
	limitList.lsexe	списку до	списку до певної			
		певної к-ті	к-ті книг			
		книг	K-11 KHVII			
		КНИІ				
		limitListResult.txt				
	4	Створення	Створення	_	_	_
6		декількох	декількох			
	separateLists.lsexe	списків	списків			
		паралельно	паралельно			
		mapasiesibile	паралельно			
		separateListsResult.t xt				
7		Перемикання	Перемикання	-	-	-
'		між списками	між списками			
	switchingLists.lsexe	книг за	книг за			
		звичайних	звичайних умов			
		умов	9211 1 W 1111111 J 11102			
		J 1110 B				
		separateListsResult.t xt				
8	4	Правильне	Правильне	-	-	_
O	ann and Tallish lance	додавання	додавання книг			
	appendToList.lsexe	книг вручну	вручну			
		000	10 0			
		T. L. ID. III.				
		appendToListResult.t xt				
9	4	Вивід списку	Вивід списку	-	-	-
	listOfLists.lsexe	достуних	достуних			
	IISTOI LISTS. ISEAC	списків книг	списків книг			
		000				
		listOfLists.txt				
		IISTOLISTS.TXT				
I	l .	<u> </u>		<u> </u>	<u> </u>	

10	4	Вивід к-ті	Вивід к-ті книг у	-	-	-
10	2.15.15.1.	книг у	поточному			
	switchingLists.lsexe	поточному	списку			
		списку	J			
		000				
		switchingListsResult. txt				
11	4	Збереження	Збереження	-	-	-
11	saveToFile.lsexe	списку у	списку у			
	save forlie.isexe	текстовий	текстовий файл			
		файл	1			
		-				
		and the back				
		result.txt				
		000				
		saveToFileResult.txt				
		Save for hercesult.txt				
12		Помилка при	Нормальне	Критична	Verified	Якщо списки
	listDeletion.lsexe	видалені	видалення			не порожні,
		одного	списків			то проблеми
		порожнього				не виникає
		списку				
13		Отримання	Отримання	-	-	-
	helpTest.lsexe	довідки по	довідки по			
		командам	командам			
		оболонки	оболонки			
		000				
		helpTestResult.txt				
		Тест екстре	мальних(гранич	чних) умог	В	
1		Нормальне	Нормальне			
1	emptyFilesTest.lsexe	опрацювання	опрацювання			
	Chipty incorestines	порожніх	порожніх			
	000	текстових	текстових			
	empty.txt	файлів(або з	файлів(або з			
	, ,	порожніми	порожніми			
		стрічками) з	стрічками) з			
	whitespaces.txt	виводом	виводом			
		відповідного	відповідного			
		,, , , , , , , , , , , , , , , , , , , ,			l .	

		Honings	Hopings			
		повідомлення	повідомлення			
		emptyFilesTestResult				
		.txt				
2		Нормальне	Нормальне			
	emptyFilesTest.lsexe	опрацювання	опрацювання			
	1000	великих за	великих за			
		обсягом	обсягом			
	largefile.txt	тектових	тектових файлів			
		файлів				
		emptyFilesTestResult .txt				
3		Помилки не	Сортування	Не	Verified	-
	sortEmptyList.lsexe	виникає,	порожнього	критично		
	, ,	проте і	списку з			
		повідомлення	виводом			
		про	відповідного			
		сорутвання	повідомлення			
		порожнього				
		списку немає				
		000				
		sortEmptyListResult.				
		txt				
4	4	Помилки не	Фільтрація	Не	Verified	_
4	filterEmptyl ist leave	виника€,	порожнього	критично		
	filterEmptyList.lsexe	проте і	списку з	_		
		повідомлення	виводом			
		про	відповідного			
		фільтрацію	повідомлення			
		порожнього				
		списку немає				
		000				
		filterEmptyListResult.				
		ťxt				

5	limitEmptyList.lsexe	Обмеження порожнього списку з виводом відповідного повідомлення limitEmptyListResult.t xt	Обмеження порожнього списку з виводом відповідного повідомлення	-	-	-
		Tec	т критичних ум	ОВ		
1	invalidCommandTest. Isexe	Нормальна обробка неправильної команди в скріпті з виводом відповідного повідомлення invalidCommandTest Result.txt	Нормальна обробка неправильної команди в скріпті з виводом відповідного повідомлення			
2	invalidParametersTes t.lsexe	Нормальна обробка неправильног о параметру команди з виводом відповідного повідомлення invalidParametersTes tResult.txt	Нормальна обробка неправильного параметру команди з виводом відповідного повідомлення			
3	invalidParametersCo unt.lsexe	Помилка при передачі 2ох параметрів в функцію сортування	Нормальна обробка неправильної кті параметрів команди скрпіта з виводом відповідного	Критичн о	Verified	-

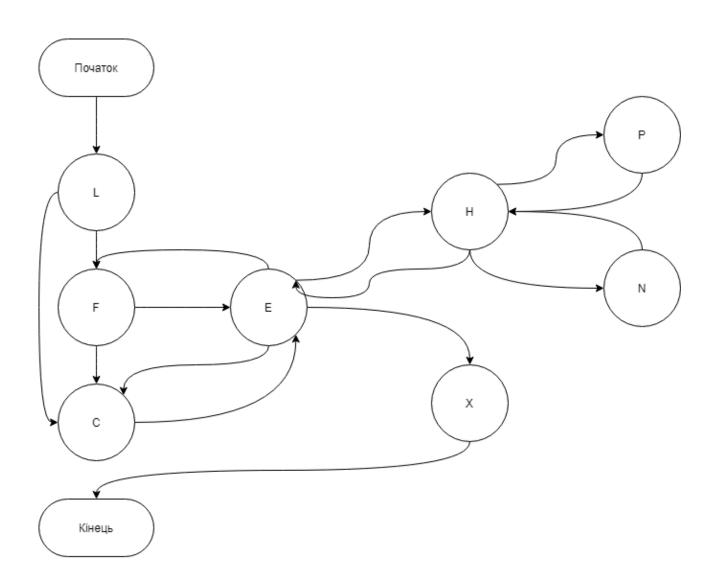
			повідомлення		
1		Нормальна	Нормальна		
4		обробка	обробка запуску		
	notExistingScriptFileT est.PNG	запуску	неіснуючого		
		неіснуючого	скриптового		
		скриптового	файлу з виводом		
		файлу з	відповідного		
		виводом	повідомлення		
		виводом відповідного	повідомлення		
		повідомлення			
		повідомлення			
		i i			
		notExistingScriptFileT estResult.PNG			
	9	Нормальна	Нормальна		
5	n at Cuistin a Tau A Cila T	обробка	обробка		
	notExistingTextFileT est.lsexe	зчитування з	зчитування з		
		неіснуючого	неіснуючого		
		текстового	текстового		
		файлу з	файлу з виводом		
		виводом	відповідного		
		відповідного	повідомлення		
		повідомлення			
		000			
		notEvictio = Ell=T==1D			
		notExistingFileTestR esult.txt			
1					l

Результати повторного тестування

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

2	listDeletion. Isexe	Нормальне видалення списків	Нормальне видалення списків	-	-	-
3	sortEmptyList.lsexe	Сортування порожнього списку з виводом відповідного повідомлення sortEmptyListResult.	Сортування порожнього списку з виводом відповідного повідомлення	-	-	-
4	filterEmptyList.lsexe	Фільтрація порожнього списку з виводом відповідного повідомлення filterEmptyListResult.	Фільтрація порожнього списку з виводом відповідного повідомлення	_	_	_
5	invalidParametersCo unt.lsexe	Нормальна обробка неправильної к-ті параметрів команди скрпіта з виводом відповідного повідомлення іnvalidParametersCo untResult.txt	Нормальна обробка неправильної кті параметрів команди скрпіта з виводом відповідного повідомлення	-	-	-

Логічна схема програми у вигляді графів управління



L – очистка екрану

F – виконання скрпітового файлу

С – виконання в консолі

Е – виконання однієї команди

H-отримання довідки по команді

Р – існуюча команда

N – неіснуюча команда

Х – вихід з програми

Результати структурного тестування

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

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.