

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут КНІТ  
Кафедра ПЗ**

**ЗВІТ**

До лабораторної роботи №3

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

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

**Лектор:**

доцент кафедри ПЗ

Левус Е.В.

**Виконав:**

студ. групи ПЗ-15

Хробак О.М.

**Прийняв:**

викладач каф. ПЗ

Самбір А. А.

«\_\_\_\_» \_\_\_\_ 2022 р.

$\Sigma =$  \_\_\_\_\_

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**

**6. Що таке граничні випробування? Наведіть три приклади.** Граничні значення області зміни вхідних даних – ті, які мають сприйматися програмою як правильні дані. Наприклад, якщо програма приймає дискретне число від 0 до 100, то потрібно перевірити її роботу при введених значеннях 0, 1, 99, 100. Якщо ж очікуване число є неперервним у діапазоні від 0 до 100, то перевіряють поведінку програми при введених значеннях в околі 0 та 100 відповідно.

**28. Яке тестування називають позитивним? Яке тестування називають успішним?** Тестування для нормальних та екстремальних вхідних даних називають позитивним тестуванням. Успішним називають те тестування, під час проведення якого було виявлено помилки.

**43. Наведіть характеристики якості ПЗ згідно актуального стандарту ІПЗ.**

- Функціональність
- Надійність
- Зручність застосування
- Ефективність
- Супроводжуваність
- Переносність

## **Завдання**

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(границьких) 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.

## Хід роботи

### StudentsList, версія 1.0

№ з/п	Тестові дані	Фактичні результати	Очікувані результати	Ступінь критичн ості	Тип звіту	Примітки
Димовий тест						
1	Запуск програми	Програма успішно запускається без помилок та аварійних завершень	Запуск програми без помилок	--	--	--
Тест нормальних умов						
1	Додавання нового елемента, що відповідає формату, в кінець списку	Новий елемент успішно доданий в кінець списку	Успішне додавання елемента та коректний запис даних	--	--	Правильний формат: Ім`я та прізвище не більше 20 символів, дата народження у форматі dd mm yyyy Оцінки ( 5 цілих чисел в межах від 1 до 5 включно, розділені пробілами)
2	Сортування заповненого списку за прізвищем	Успішне сортування списку за прізвищем	Правильне сортування усіх даних в списку за даною умовою	--	--	--
3	Видалення елемента за індексом, якщо список не порожній і елемент з введеним індексом існує	Елемент з введеним індексом успішно видалений	Успішне видалення елемента з індексом, який ввів користувач	--	--	--
4	Видалення елементів, у яких за 1 і 3 екзамени є оцінка 3, якщо список не порожній і такі елементи існують	Елементи, які відповідають умові, успішно видалені	Успішне видалення елементів, які відповідають умові	--	--	--

5	Зміна елемента з введеним індексом, якщо список не порожній, елемент з введеним індексом існує, а новий елемент відповідає формату	Елемент з введеним індексом успішно змінений	Успішна зміна елемента за введеним індексом	--	--	Правильний формат: Ім`я та прізвище не більше 20 символів, дата народження у форматі dd mm yyyy Оцінки ( 5 цілих чисел в межах від 1 до 5 включно, розділені пробілами)
6	Пошук елемента за прізвищем, якщо такий елемент існує і список не порожній	Усі елементи з введеним прізвищем успішно знайдені і виведені в консоль	Виведення елементів з введеним прізвищем в консоль	--	--	--

#### Тест екстремальних умов

1	Додавання нового елемента або зміна ісуючого у такому форматі: ім`я та прізвище довжиною 20 символів, дата – 31.12.2022, оцінки – 5 5 5 5 5	Новий елемент успішно доданий в кінець списку, зміна пройшла успішно	Успішне додавання або зміна елемента та коректний запис даних	--	--	--
2	Спроба відкриття і зчитування даних з порожнього файлу	Програма виводить порожню таблицю	Вивід пустої таблиці в консоль	--	--	--
3	Спроба сортування порожнього списку	Програма завершує роботу коректно, але не виводиться ніяка інформація про те, що список порожній	Вивід відповідного повідомлення про те, що список порожній	Низький	Пропозиція	

#### Тест виняткових ситуацій

1	Додавання нового елемента у такому форматі: прізвище та ім`я більше 20	Програма аварійно завершує роботу, бо	Вивід відповідного повідомлення про те, що	Високий	Помилка кодування	
---	--	---------------------------------------	--	---------	-------------------	--

	символів, формат дати і оцінок правильний	перевищено довжину масива символів	введено неправильні дані			
2	Додавання нового елемента у такому форматі: дата записана в неправильному форматі, решта вхідних даних є коректними	Помилок не виникає, але програма виводить в таблицю некоректні результати	Вивід відповідного повідомлення про те, що введено неправильні дані	Високий	Помилка кодування	
3	Додавання нового елемента у такому форматі: оцінки записані в неправильному форматі, решта вхідних даних є коректними	Програма або аварійно завершує роботу (введено не цифри), або записує в поле Оцінки неправильні дані	Вивід відповідного повідомлення про те, що введено неправильні дані	Високий	Помилка кодування	

## Результати повторного тестування

### StudentsList, версія 1.1

№ з/п	Тестові дані	Фактичні результати	Очікувані результати	Ступінь критичності	Тип звіту	Примітки
Димовий тест						
1	Запуск програми	Програма успішно запускається без помилок та аварійних завершень	Запуск програми без помилок	--	--	--
Тест нормальних умов						
1	Додавання нового елемента, що відповідає формату, в кінець списку	Новий елемент успішно доданий в кінець списку	Успішне додавання елемента та коректний запис даних	--	--	Правильний формат: Ім`я та прізвище не більше 20 символів, дата народження у форматі dd mm yyyy

						Оцінки ( 5 цілих чисел в межах від 1 до 5 включно, розділені пробілами)
2	Сортування заповненого списку за прізвищем	Успішне сортування списку за прізвищем	Правильне сортування усіх даних в списку за даною умовою	--	--	--
3	Видалення елемента за індексом, якщо список не порожній і елемент з введеним індексом існує	Елемент з введеним індексом успішно видалений	Успішне видалення елемента з індексом, який ввів користувач	--	--	--
4	Видалення елементів, у яких за 1 і 3 екзамени є оцінка 3, якщо список не порожній і такі елементи існують	Елементи, які відповідають умові, успішно видалені	Успішне видалення елементів, які відповідають умові	--	--	--
5	Зміна елемента з введеним індексом, якщо список не порожній, елемент з введеним індексом існує, а новий елемент відповідає формату	Елемент з введеним індексом успішно змінений	Успішна зміна елемента за введеним індексом	--	--	Правильний формат: Ім`я та прізвище не більше 20 символів, дата народження у форматі dd mm yyyy Оцінки ( 5 цілих чисел в межах від 1 до 5 включно, розділені пробілами)
6	Пошук елемента за прізвищем, якщо такий елемент існує і список не порожній	Усі елементи з введеним прізвищем успішно знайдені і виведені в консоль	Виведення елементів з введеним прізвищем в консоль	--	--	--

#### Тест екстремальних умов

1	Додавання нового елемента або зміна	Новий елемент	Успішне додавання або	--	--	--
---	-------------------------------------	---------------	-----------------------	----	----	----

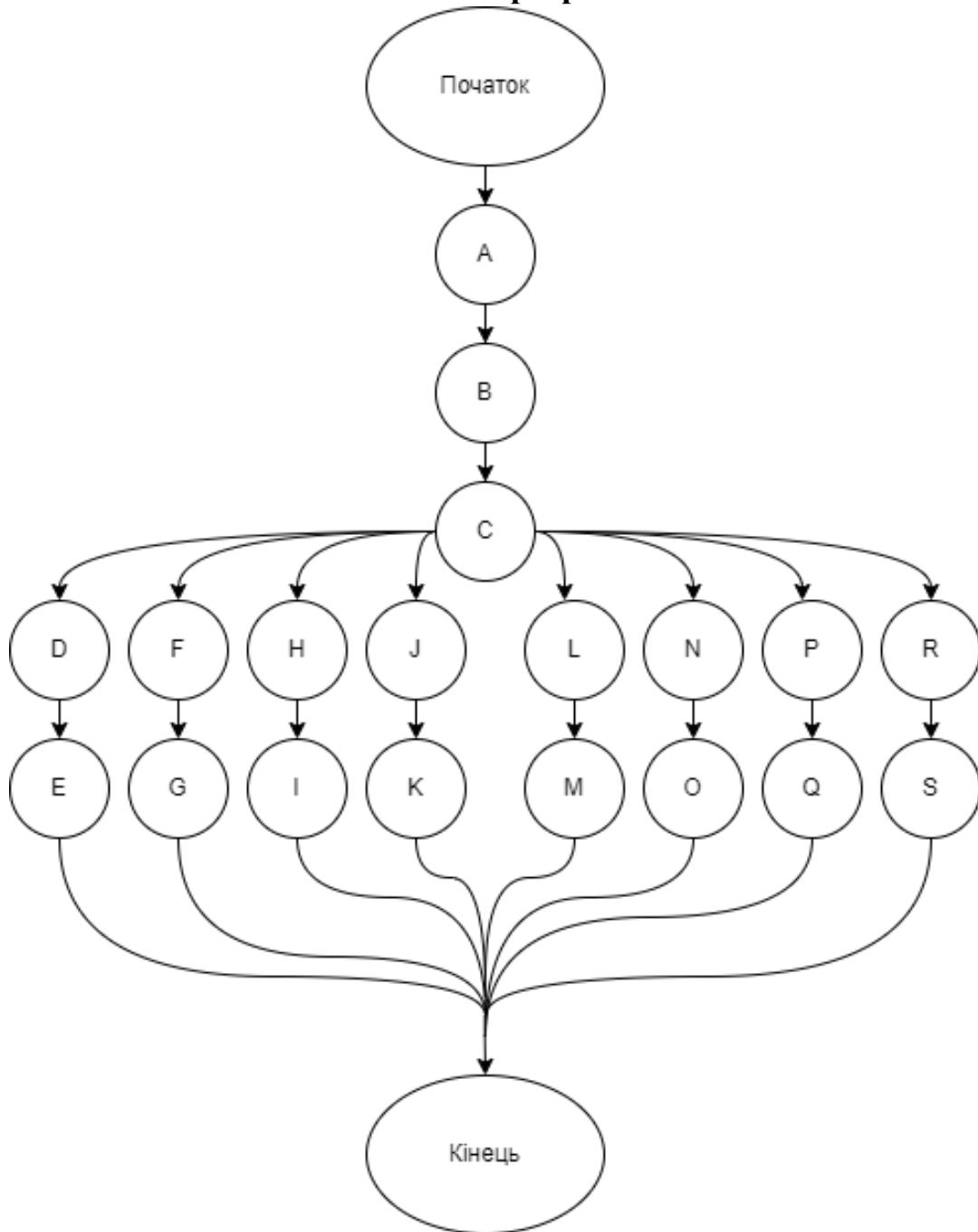
	існуючого у такому форматі: ім`я та прізвище довжиною 20 символів, дата – 31.12.2022, оцінки – 5 5 5 5 5	успішно доданий в кінець списку, зміна пройшла успішно	зміна елемента та коректний запис даних			
2	Спроба відкриття і зчитування даних з порожнього файлу	Програма виводить порожню таблицю	Вивід пустої таблиці в консоль	--	--	--
3	Спроба сортування порожнього списку	Вивід в консоль інформації про те, що список порожній	Вивід відповідного повідомлення про те, що список порожній	--	--	--

**Тест виняткових ситуацій**

1	Додавання нового елемента у такому форматі: прізвище та ім`я більше 20 символів, формат дати і оцінок правильний	Виводиться повідомлення про те, що введено неправильні дані	Вивід відповідного повідомлення про те, що введено неправильні дані	--	--	--
2	Додавання нового елемента у такому форматі: дата записана в неправильному форматі, решта вхідних даних є коректними	Виводиться повідомлення про те, що введено неправильні дані	Вивід відповідного повідомлення про те, що введено неправильні дані	--	--	--
3	Додавання нового елемента у такому форматі: оцінки записані в неправильному форматі, решта вхідних даних є коректними	Виводиться повідомлення про те, що введено неправильні дані	Вивід відповідного повідомлення про те, що введено неправильні дані	--	--	--

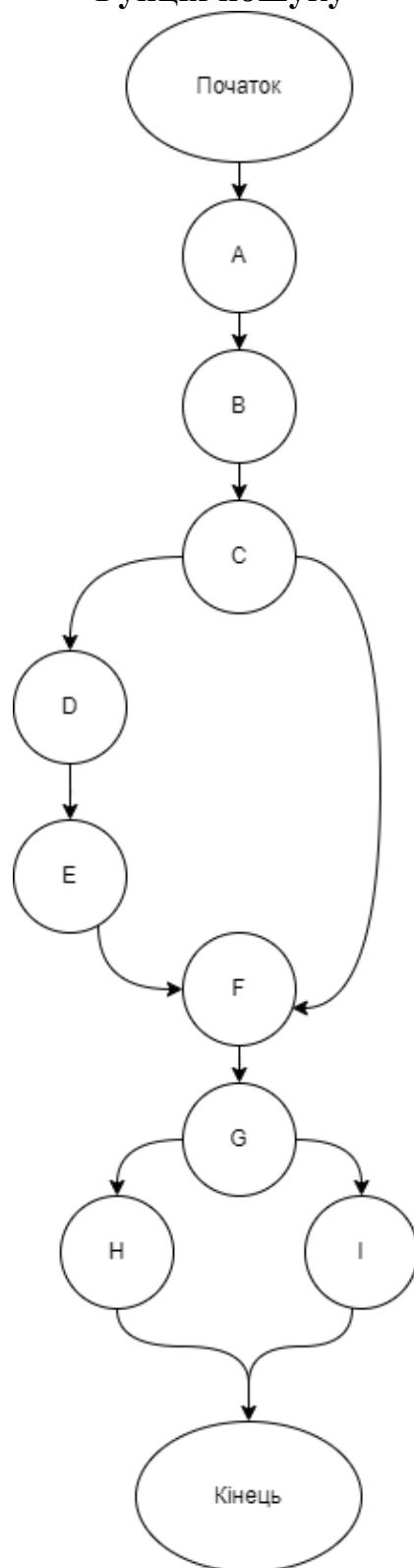
## Тестування логічної схеми програми (головна програма та дві функції-підпрограми)

Головна програма



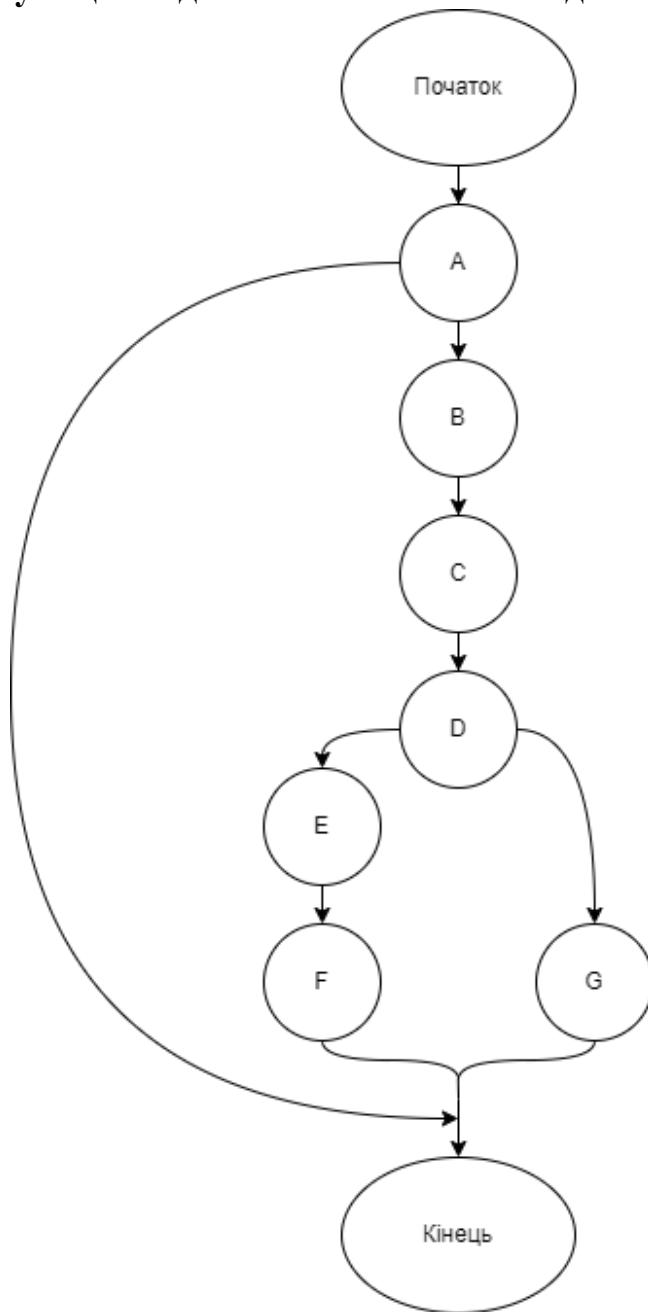
№ з/п	Маршрут	Результат
1	(початок)-A-B-C-D-E-(кінець)	Сортувати список за прізвищем
2	(початок)-A-B-C-F-G-(кінець)	Видалення елемента за індексом
3	(початок)-A-B-C-H-I-(кінець)	Видалення елемента за умовою
4	(початок)-A-B-C-J-K-(кінець)	Додавання елемента в кінець списку
5	(початок)-A-B-C-L-M-(кінець)	Зміна елемента за індексом
6	(початок)-A-B-C-N-O-(кінець)	Пошук елемента за прізвищем
7	(початок)-A-B-C-P-Q-(кінець)	Збереження списку у файлі
8	(початок)-A-B-C-R-S-(кінець)	Вивід повідомлення, що введено неправильні дані

## Функція пошуку



№ з/п	Маршрут	Результат
1	(початок)-A-B-C-D-E-F-G-H- (кінець)	Вивід в консоль результатів пошуку
2	(початок)-A-B-C-F-G-I- (кінець)	Вивід повідомлення про те, що елемента з таким прізвищем немає у списку

### Функція видалення елемента за індексом



№ з/п	Маршрут	Результат
1	(початок)-A-B-(кінець)	Вивід повідомлення про те, що список порожній
2	(початок)-A-B-C-D-E-F-(кінець)	Видалення елемента з списку
3	(початок)-A-B-C-D-G-(кінець)	Вивід повідомлення про те, що немає елемента з таким індексом, який ввів користувач

### Висновки

Під час виконання лабораторної роботи я вивчив основні методи та принципи тестування на прикладі створеної власної програми. Провів функціональне і структурне тестування. Під час тестування виявив декілька помилок, усунув їх і провів повторне тестування нової версії програми.