

2021

# Structures and Algorithms report

TESTING AND SELF REFLECTION

ROSS O'NEILL

## Introduction

This report will contain test logs for Steps 1,2 and 3 of the Structures and Algorithms coursework for the academic year of 2020/2021. The report will be broken up into 4 sections:

1. Test log for Step 1
2. Test log for Step 2
3. Test log for Step 3 – In the Step 3 project there is Junit testing. In this report, there will be a test log of methods that have not been testing using Junit.
4. A self-reflection of this coursework

The test logs will contain tables describing the test cases, data used, expected/actual outputs and any comments about the tests themselves. The self-reflection will be a brief description of what went well and what I could have done better in the coursework. Part of this coursework was also to create a GitHub repository and show evidence of use. The link for my repository for this project is:

<https://github.com/rossco122005/StructuresAndAlgorithmsCourswork>

## Step 1 Test Log

Test case	Method(s) Tested	Input	Expected Result	Actual Result	Comments
Adding Pet objects into the Pet binary search tree	public void insert(Pet pet)	3 Pet objects instantiated with the names: "dog", "cat", "Insect"	After each pet is added. An appropriate message of the pet being added is displayed	As expected	I have added the pets not in alphabetical order to check they will be displayed in alphabetical order in another test. I also made Insect have a capital I to show that the program will ignore case.
Trying to add a duplicate pet with the expectation of a NotUniqueException being thrown	public void insert(Pet pet)	2 Pet objects with the names: "cat", "insect"	A message showing these pets already exist	As expected	The program caught the lower case "i" in the insect object.
Trying to find a pet in the BST	public Pet find(Pet pet)	2 Pet objects with the names: "cat", "insect"	A message showing that the pet is found after each string is entered	As expected	The program caught the lower case "i" in the insect object.
Testing for a NotFoundException when trying to find a pet not on the system	public Pet find(Pet pet)	2 Pet objects with the names: "horse", "reptile"	A message showing that the pet types do not exists after each is entered	As expected	
Testing the system if it can display a specific pet type	public String displayPetDetails()	2 Pet objects with the names: "cat", "insect"	A message displaying the pet type along with a message showing there are currently no products as this will be added at a later stage	As expected	The program caught the lower case "i" and displayed "Insect"

Testing that the program will display all pet types in alphabetical order	public String displayInOrder()	3 Pet objects instantiated with the names: "dog", "cat", "Insect" inserted in this order	A message displaying the pets in alphabetical order	As expected	The Insect pet is displayed in the correct place as the program ignores case
Testing that the program will delete a pet and still display remaining pets in alphabetical order	public Pet remove(Pet pet) public String displayInOrder()	The Pet object "dog"	Displaying all pets will display "cat" and "Insect" in alphabetical order	As expected	The program will also delete Insect if the user enters "insect" as the pet to delete
Testing that a pet that has already been removed throws a NotFoundException	public Pet remove(Pet pet)	The Pet object "dog"	A message showing that the Pet dog does not exist	As expected	

## Step 2 Test Log

Test case	Method(s) Tested	Input	Expected Result	Actual Result	Comments
Adding products into the Linked List	public void addProduct(Comparable object)	4 Product objects with the following details:  Product 1: "1", "bowl", "10" Product 2: "2", "cage", "20" Product 3: "3", "food", "30" Product 4: "4", "ball", "40"	A message showing that the product has been added after each is added	As expected	
Trying to add a duplicate product with NotUniqueException being expected	public void addProduct(Comparable object)	Product object with the details: "1", "bowl", "10"	A message showing that the product already exists	As expected	The quantity field does not part of the compareTo check. This could be taken out for the program to be more efficient.  The program also allows for products of the same name under different ID numbers to be added. Scenarios where this is needed is expected.
Removing a product from the Linked List	public Comparable remove(Comparable object)	Product with the details: "3", "food", "30"	A message showing that the product has been removed and does not display when	As expected	The program does not ask for quantity as this is not needed to delete a node in the linked list

			showing all products in the Linked List		
Trying to remove a product that does not exist. NotFoundException is expected	public Comparable remove(Comparable object)  public Comparable findProduct(Comparable object)	Product with the details: "3", "food", "30"	Message showing product has not been found	As expected	
Trying to find a product and displaying the appropriate details	public Comparable findProduct(Comparable object)	Product object with the details: "4", "ball", "40"	All information regarding product displayed	As expected	
Trying to find an product that does not exist	public Comparable findProduct(Comparable object)	Product with the details: "3", "food", "30" (after removal)	Product not found message	As expected	

## Step 3 Test Log

Test case	Method(s) tested	Input	Expected Result	Actual Result	Comments
Displaying all pets and products stocked by the company in alphabetical order of Pet	public String displayAllPetsAndProducts()	<p>The following pets and products will be added in this order:</p> <p>Pet: "reptile"</p> <p>Product 1: "1", "cage", "20"</p> <p>Product 2: "2", "food", "30"</p> <p>Pet: "dog"</p> <p>Product 1: "1", "bowl", "20"</p> <p>Product 2: "2", "lead", "50"</p> <p>Pet: "cat"</p> <p>Product 1: "1", "litter", "20"</p> <p>Product 2: "2", "food", "40"</p>	All pets and products are displayed in alphabetical order of Pet	As expected	The products are being displayed in alphabetical order. I would recommend changing this to be ordered by product number in any further development
Displaying all products that are stocked for a specific pet	public String displayProductsForPet() public Pet find(Pet pet)	Pet object with the name: "cat"	All products stocked for cat are displayed	As expected	
Displaying all products that are stocked for a specific pet but no products exist	public String displayProductsForPet() public Pet find(Pet pet)	Pet object with the name: "cat" after a fresh run of the program where no	A message showing there are no products stored for the pet	As expected	

		products are added for "cat"			
--	--	---------------------------------	--	--	--

## Self-Reflection

I thoroughly enjoyed completing this coursework. It was enjoyable bringing all my knowledge from the module together to be able to logically plan and execute this project. If I was able to give my past self any advice now that I have completed the project, I would advise to take more time in the planning stage of this project. If I had spent more time coming up with a test plan for the project, I believe that Stage 1 would have gone smoother. My process and execution of Step 1 was quite clumsy, and I feel that spending more time thinking about what the outputs were going to be, would have helped me get started sooner and more effectively. Thankfully, once I had the custom binary search tree created for the pets, the rest of the project went relatively smoothly. I would also warn that underestimating this project could have been a dangerous path. I initially thought that the project was going to be much simpler than it was by looking at the brief, but with good planning and thought process, I realised quickly that it wasn't as simple as I thought.

As I have mentioned, after the binary search tree was created in Step 1, the rest of the project went relatively smoothly. I believe this came from the effort I put into the lectures and the labs throughout the course. From the beginning, I made sure that I typed out as much code as I possible when going through the self-assessments and the labs, even simple `System.out.println()` lines of code. I made sure that I tried my absolute best when creating solutions for the labs before going to the solutions provided by the university. It wasn't until Unit 8 that I began to need the solutions, and by typing out as much code as I could before going to the solutions, I found that my understanding of the solutions was clear. The practice of typing everything out helped me greatly when coming to complete this coursework. At no point was there any code that I did not understand, and any problems I came across were not syntax problems, they were logic problems. With this in mind, it is a practice I will take throughout my studies and my career. It helped me greatly in this course and I am sure it will in the future.