Written Report

Jaydon Cameron Assignment 2 SENG2200

1. For each of the programs keep track of how much time you spend designing, coding and correcting errors, and how many errors you need to correct. Sum these numbers once your implementation is complete.

Designing

Date	Item	Time Spent (hours)
22/03/2022	Understanding program requirements	0.5
22/03/2022	Reviewed Polymorphism and Generics	1
23/03/2022	Looked at the documentation for the `Iterable<T>` interface. Iterables had not been covered in the lectures at the time of writing this.	0.6
23/03/2022	Implementing a method for reading input data specifications	0.5
23/03/2022	Looked at implementing a factory method for `PlanarShape`	1

Coding

Date	Item	Time Spent (hours)
22/03/2022	Created main class `A2`	0.1
22/03/2022	Created abstract class `PlanarShape` and implemented `Comparable<T>.compareTo(T o)`	0.5
22/03/2022	Copied `Point` class from A1	0.1
22/03/2022	Copied `Polygon` class from A1 and updated it to work with polymorphism	0.5
22/03/2022	Started using Google Java Style Guide as a guide for formatting my code and reformatted existing code	0.5
22/03/2022	Created `Circle` class and implemented methods	1

Date	Item	Time Spent (hours)
22/03/2022	Created `SemiCircle` class and implemented methods	0.5
22/03/2022	Copied `Node` class from A1 and updated it to work with the design of A2	0.3
23/03/2022	Created `LinkedList` class	0.1
23/03/2022	Implemented `append()` and `prepend()` methods	0.2
23/03/2022	Implemented `iterator()` method to return an `Iterator<T>` object that contains the methods `hasNext()` , `getNext()` , and `remove()` that uses the instance variable `current` to traverse the `LinkedList`	0.8
23/03/2022	Created `SortedLinkedList` class	0.05
23/03/2022	Updated `toString()` method in `SemiCircle` class to satisfy version 2.0.2 of the specifications	0.1
23/03/2022	Updated `toString()` method in `Polygon` class to satisfy version 2.0.2 of the specifications	0.1
23/03/2022	Updated `toString()` method in `Circle` class to satisfy version 2.0.2 of the specifications	0.1
23/03/2022	Overloaded `next()` and `reset()` in `LinkedList` to throw an `Exception`	0.5
23/03/2022	Overloaded `insert()` in `LinkedList` to direct all incoming data to `append()`	0.09
23/03/2022	Created `SortedLinkedList` class and overloaded `prepend()` and `append()` methods to throw an `Exception`	0.5
23/03/2022	Overloaded `insert()` in `SortedLinkedList` to direct any incoming data to the `insertInOrder()` method	0.1
23/03/2022	File reading method using switch statement	1
23/03/2022	Updated switch case to enhanced switch statement	0.1

Correcting Errors

Date	Error	Time Spent (hours)
23/03/2022	Design Error 1	0.1

2. Keep a log of what proportion of your errors come from design errors and what proportion from coding/implementation errors. Address any trends you noticed from your logs and results.

Design Errors

Date	Error	Time Spent (hours)
23/03/2022	Coding and implementation error 1	0.1
23/03/2022	Coding and implementation error 2	0.1
23/03/2022	Coding and implementation error 3	0.1
23/03/2022	Coding and implementation error 4	0.1

Coding and Implementation Errors

Date	Error	Reason
23/03/2022	`unreported error exception java.lang.Exception; must be caught or declared to be thrown` in LinkedList.java	Needed to declare that `append()` throws an `UnsupportedOperationException`
23/03/2022	`deprecated item is not annotated with @Deprecated` for 2 methods in LinkedList.java	The javadocs for `reset()` and `next()` included the `@deprecated` tag without the methods being annotated
23/03/2022	`Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException` in A2.java	No arguments were being passed from the command line. I needed to include an argument for the filename.
23/03/2022	`Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for length 2` for `LinkedList@1b2c6ec2` in A2.java	

Including `Triangle` and `Square`

> Provide a (*brief*) design of how you would further extend your **PA2** so that it specifically included `Triangle` and `Square` figures. Draw the UML class diagram for this new program (*intricate detail not required*). What attribute data do you need in each case?


Program with Square and Triangle - UML Class Diagram

Modeling an `Ellipse`

> Investigate the mathematical structure of an `Ellipse` on the Cartesian plane. How would you model the `Ellipse`? How would you then calculate its area and `originDistance()`? How would this be incorporated into your program? Draw another UML class diagram to show this.