Table of Contents

[Initial Diagram And Design 1](#_Toc511769279)

[Final UML class diagram 2](#_Toc511769280)

[Project Design 3](#_Toc511769281)

[Walkthrough 3](#_Toc511769282)

[Explaining how the graph is created 3](#_Toc511769283)

[Explaining how the schedule is maintained and updated 3](#_Toc511769284)

[Explaining the comparator for the priority queue 3](#_Toc511769285)

[Explaining how the A\*search chooses the best node and maintains the search 3](#_Toc511769286)

[Explaining how the path travelled can be checked as a list of directional edges 3](#_Toc511769287)

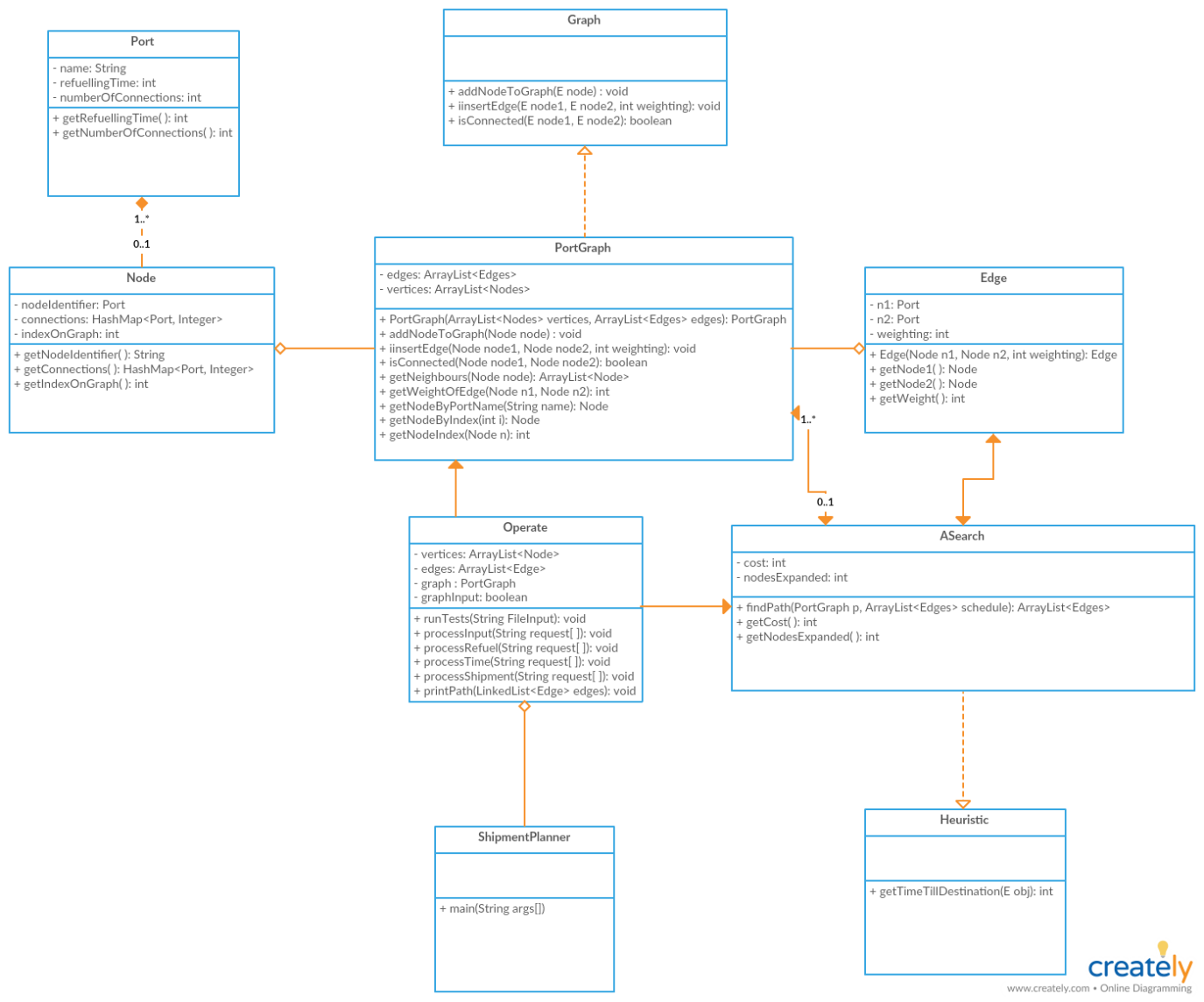
[Explaining how the A\*search knows its done based on the current path 3](#_Toc511769288)

[CRC Cards 3](#_Toc511769289)

[Run-time complexity analysis 3](#_Toc511769290)

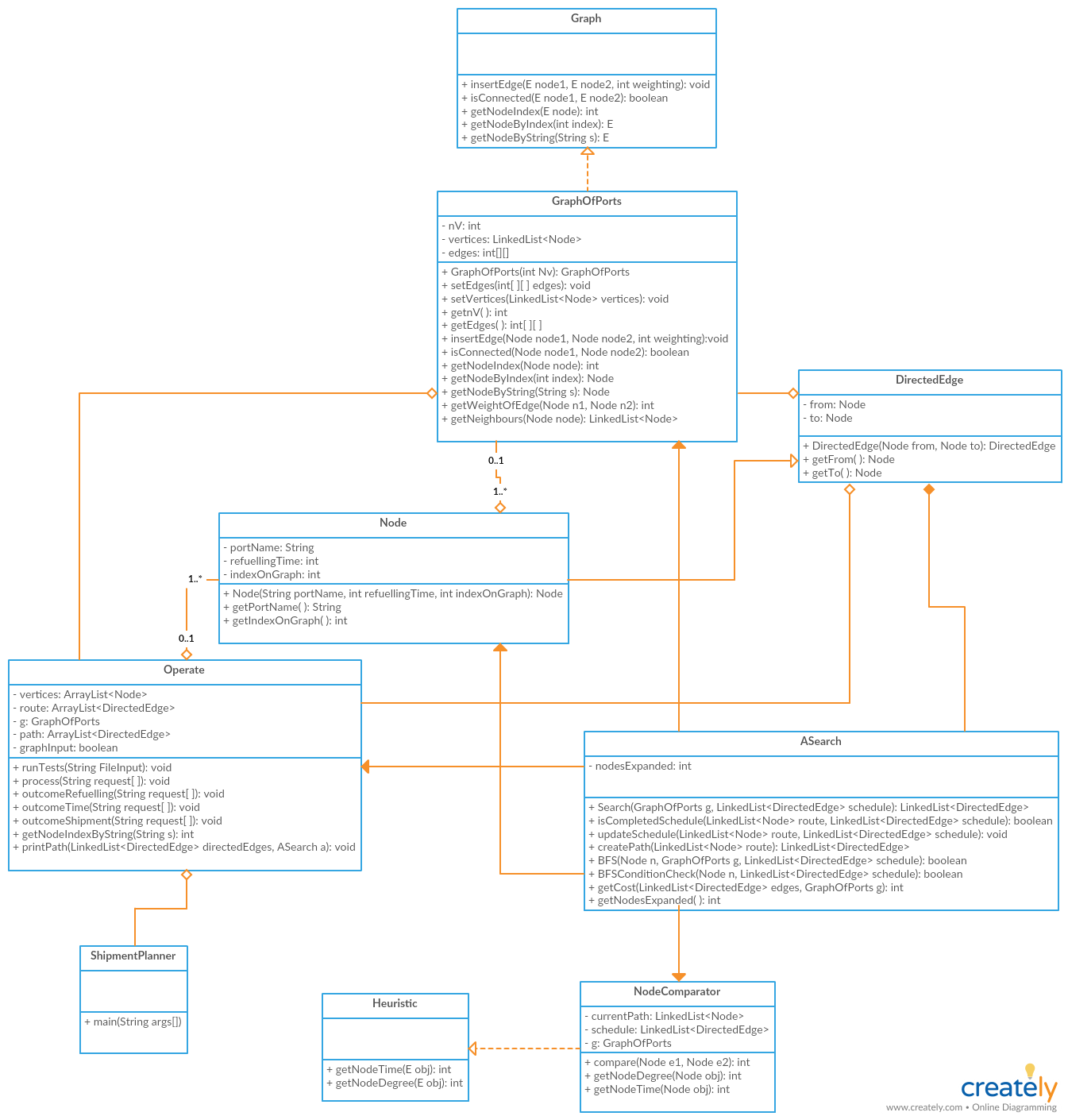
# Initial Diagram And Design

This UML diagram is the first one I made before I started any coding, I tried using this exact design however ran into some problems following it exactly, the final UML diagram (after coding) is very similar with some slight changes however the overall design is almost exactly the same (removed some of the redundant classes) I also realised I needed to make a custom class for my comparator to keep my priority queue updated



# Final UML class diagram

After I made a few minor changes to the design, and making it actually work above design is almost exactly correct however the port class is redundant and I didn’t include the comparator class to make my custom comparator that uses the heuristic as a scoring system. Also a few changes were made to the fields and methods but the overall design is very similar (also changed class names). Methods were added (some helper functions and so) to make it possible as the above design didn’t really encounter problems till coding and debugging. The reason so many lines cross over is the operate class which does all the testing technically relies on all the other classes and uses them, there is a connection but if they weren’t considered and it only relied on ASearch and Graph, this diagram would have a lot less collisions which were unavoidable.



# Project Design

# Walkthrough

# Explaining how the graph is created

# Explaining how the schedule is maintained and updated

# Explaining the comparator for the priority queue

# Explaining how the A\*search chooses the best node and maintains the search

# Explaining how the path travelled can be checked as a list of directional edges

# Explaining how the A\*search knows its done based on the current path

# CRC Cards

# Run-time complexity analysis