
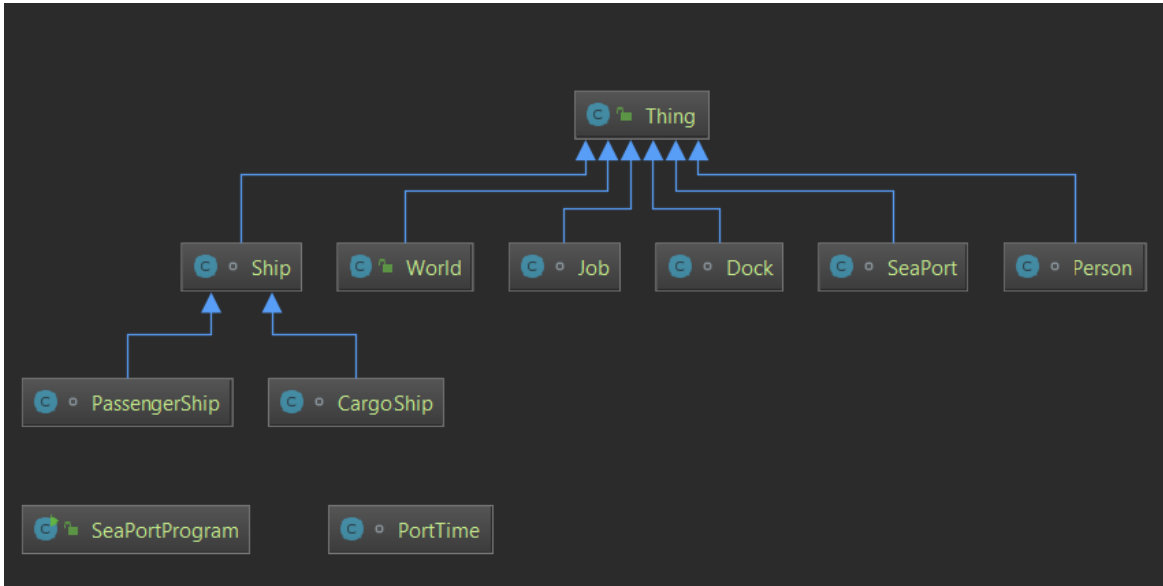


Name: William Crutchfield

Date: 01/26/19

Course: CMSC 335 – Object-Oriented and Concurrent Programming

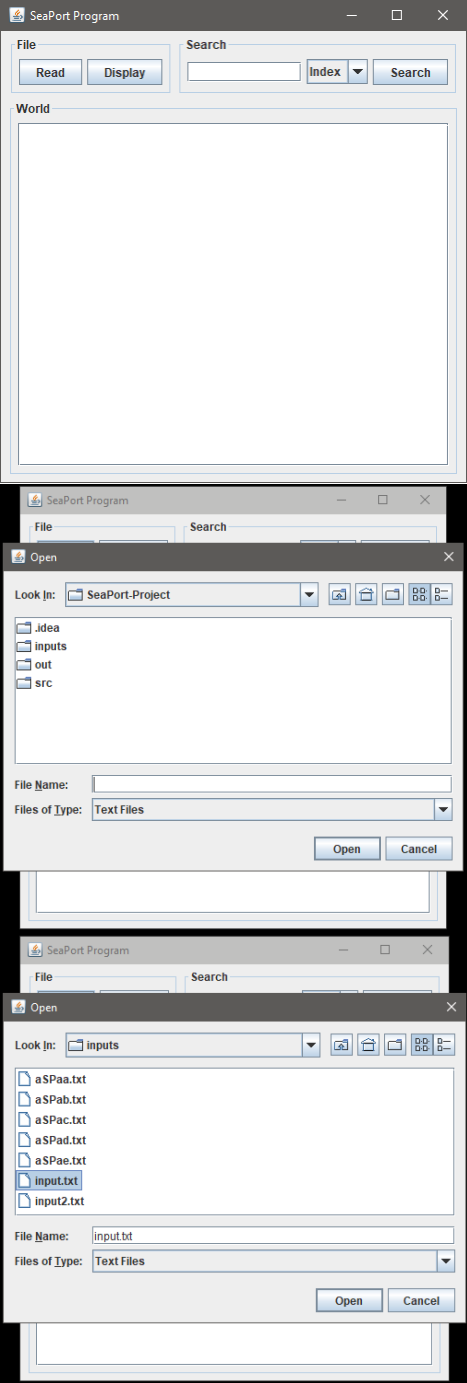
Project: 1

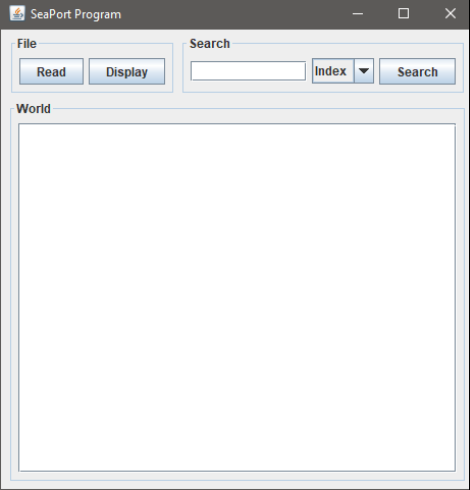
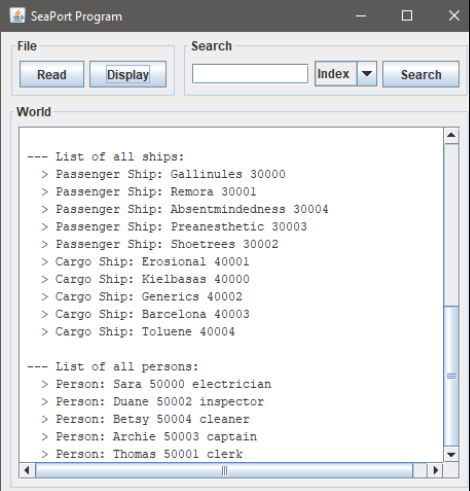
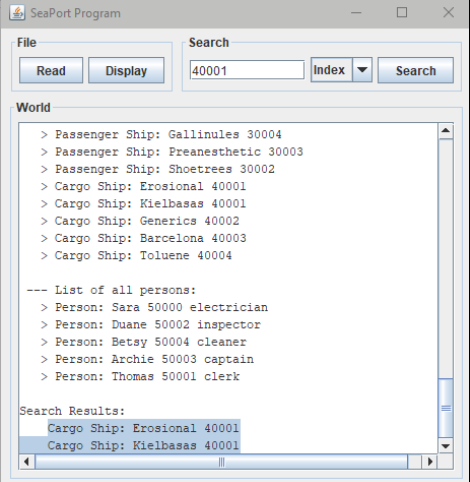
1	<p><b>Source code, data files, and configuration files (if any)</b></p> <p>Insert a zipped file of NetBeans ALL project files (so that it could be unzipped and loaded into NetBeans IDE again), zipped file of all data files, and zipped file of configuration files (if any). :</p> <p> src.zip</p>
2	<p><b>Design</b></p> <p>Insert here UML Class diagram, explain classes, variables, methods, explain how classes tie to the requirements of the project:</p>  <pre>classDiagram     class Thing     class Ship     class World     class Job     class Dock     class SeaPort     class Person     class PassengerShip     class CargoShip     class SeaPortProgram     class PortTime      Thing &lt; -- Ship     Thing &lt; -- World     Thing &lt; -- Job     Thing &lt; -- Dock     Thing &lt; -- SeaPort     Thing &lt; -- Person     Ship &lt; -- PassengerShip     Ship &lt; -- CargoShip     class SeaPortProgram     class PortTime</pre> <p>This UML Class diagram shows how the classes are connected. As we can see, all classes are a child of the Thing class, except for the SeaPortProgram, and PortTime classes. We can also derive that the PassengerShip and CargoShip classes are children of the Ship class.</p>
3	<p><b>User Guide</b></p> <p>Explain how a user starts &amp; runs your project, and any specific features with screenshots:</p> <p>In order to run this project, download the src.zip file and extract. Once extracted, open the src files up in any IDE. Lastly, run the SeaPortProgram.java file.</p>

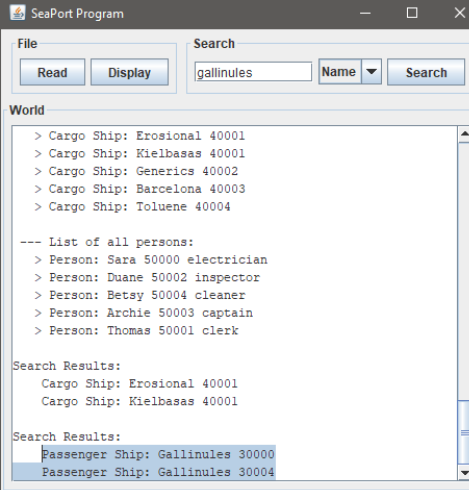
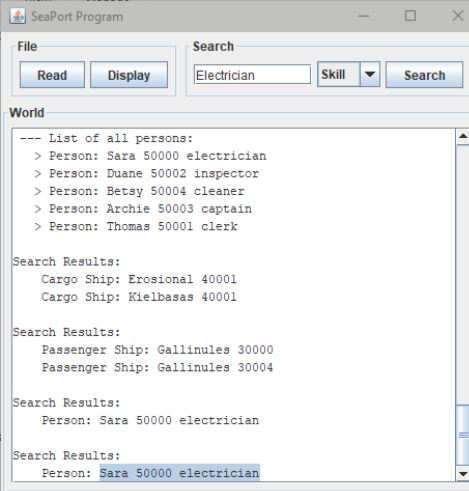
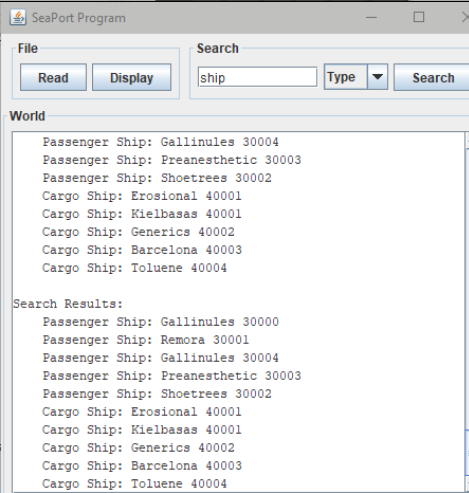
4

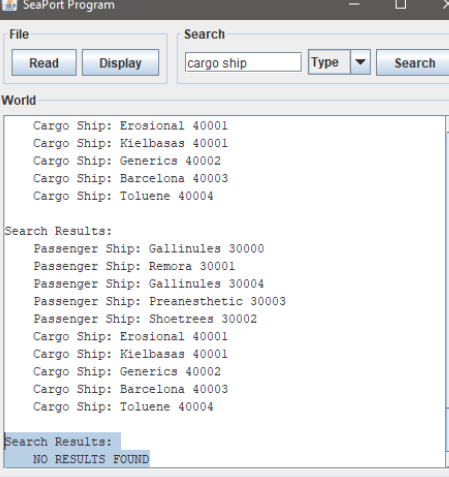
**Test Plan**

Complete this table and extend it with your test cases:

Test ID	Test Case	Selected Input	Expected Output	Actual Output (Screenshots)	Pass/Fail
1	Input.txt toString	Input.txt	Input.txt toString		Pass, we can read in a file, and output its toString method in the Text Area

				<div></div> <p>At first, it appears that the file has not been read in. But in order to confirm that the toString works, we will have to click on the display button.</p> <div></div>	
2	Multiple index search (input2.txt)	40001 – Index Search	Erosional 40001 + Kielbasas 40001	<div></div>	Pass, we can search for multiple indexes at once

3	Multiple Name search – ignore String case (input2.txt)	Gallinules – Name Search	Gallinules 30000 + Gallinules 30004		Pass, we can search for multiple names at once, and case doesn't affect the results
4	Skill search – ignore String case (input2.txt)	Electrician	Sara 50000 electrician		Pass, we can search for skill, case doesn't affect the results
5	Type search – ignore case (input2.txt)	ship – type search	10 unique ships		Pass, we can search for multiple types at once. This allows us to isolate a single object type.

	6	Type search – (input2.txt)	Cargo Ship – type search	5 unique ships		Fail, we can not currently search by ship subtype. This will need to be implemented on the next project
5	<b>Reflection and Lessons Learned</b> Reflect on your experience completing this project and the lessons you learned:  This was a great, and challenging, project! I believe that I have met all of the requirements, and I am excited to continue with this project series. I think I could improve some of my methods in my World.java class, in terms of efficiency. But, since we will be implementing maps in the second project of this series, I did not tinker with it any more. This was a great challenge, and refresher for me, before continuing with this course. Please let me know if there is anything I could improve upon!					