Jason Harkness

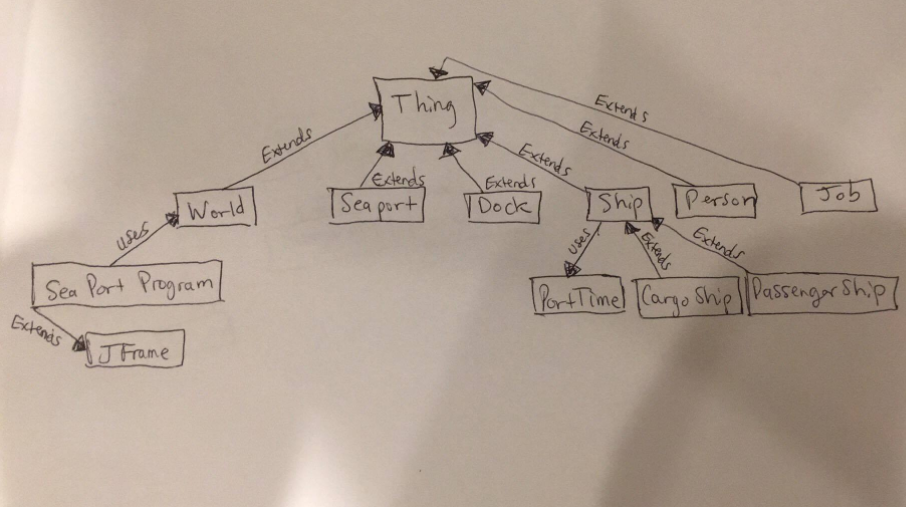
November 22, 2017

Project 2

CMSC 335 7981

**Design**

UML Diagram



**Classes, Variables and Methods**

Class Thing: Is the parent class for World, Seaport, Dock, Ship, Person, and Job. It stores variables for index, name, and parent. There is a constructor with Scanner input to store data for each variable and getter methods for each variable

SeaPortProgram extends JFrame

hmsp: HashMap<Integer, Seaport>

hmd: HashMap<Integer, Dock>

hms: HashMap<Integer, Ship>

hmp: HashMap<Integer, Person>

variables used by the GUI interface

world: World

Thing implement Comparable <Thing>

index: int

name: String

parent: int

World extends Thing

ports: ArrayList <SeaPort>

time: PortTime

SeaPort extends Thing

docks: ArrayList <Dock>

que: ArrayList <Ship> // the list of ships waiting to dock

ships: ArrayList <Ship> // a list of all the ships at this port

persons: ArrayList <Person> // people with skills at this port

Dock extends Thing

ship: Ship

Ship extends Thing

arrivalTime, dockTime: PortTime

draft, length, weight, width: double

jobs: ArrayList <Job>

PassengerShip extends Ship

numberOfOccupiedRooms: int

numberOfPassengers: int

numberOfRooms: int

CargoShip extends Ship

cargoValue: double

cargoVolume: double

cargoWeight: double

Person extends Thing

skill: String

CompareShip implements Comparator<Ship>

s: String

Job extends Thing - optional till Projects 3 and 4

duration: double

requirements: ArrayList <String>

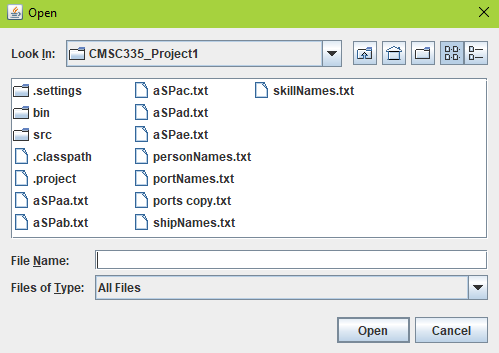
// should be some of the skills of the persons

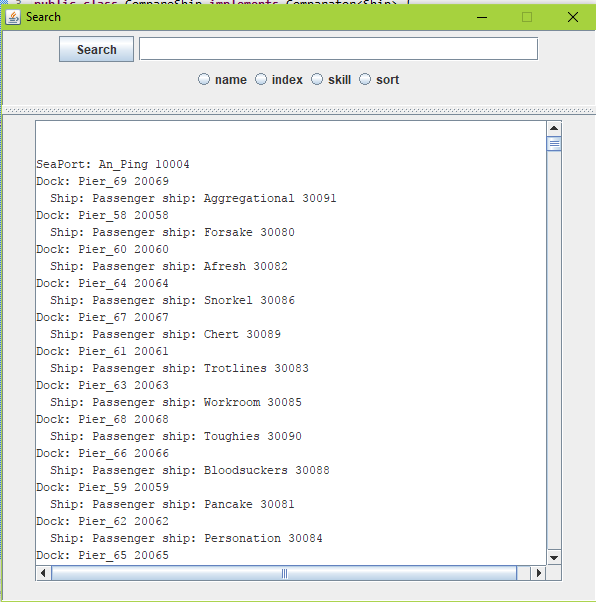
PortTime

time: int

**Users Guide**

The user can run the program through an IDE such as Eclipse or Netbeans. All documents required for the program are provided and should be placed in the same src folder. When the program is compiled and run a JFileChooser dialogue box is displayed and the user must select a file such as aSPae.txt which contains the input for the program. When the open button is clicked, another window is displayed allowing the user to search the data file by name, index, or skill and also sort ships by attribute. If a match is found in the data file it will be displayed in the JScrollPane. I created a data structure of 4 hashmaps. Loops were created like in the arraylists from project 1 to fill each hashmap according to their type. The hmsp hashmap of type Seaport stores all information for all objects. Searching for an object by index is much quicker than by searching through the entire array list. I implemented comparators to support sorting. Ships are sorted by weight, length, width, and draft then displayed to the window. Below are screenshots of the dialogue box to choose the file and the search window:



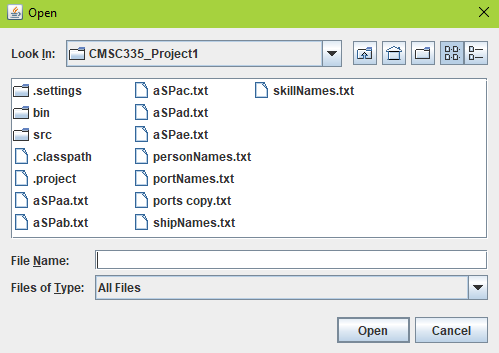


**Test Plan**

Input: User runs program

Expected Results: Program compiles and outputs file chooser window

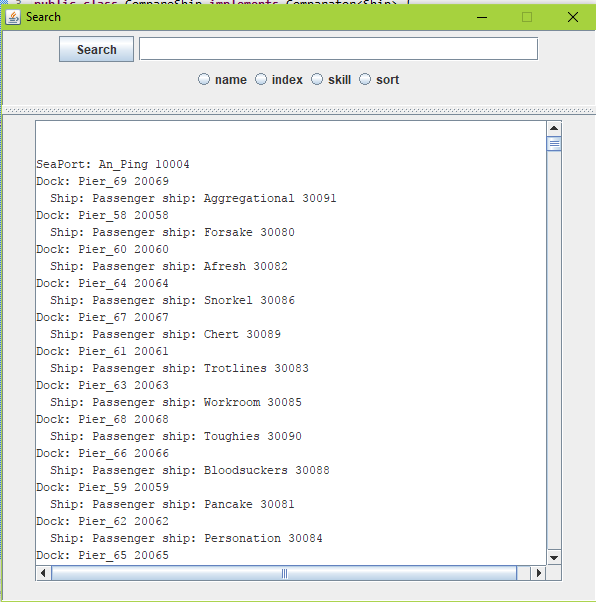
Actual Results: PASS



Input: User chooses file aSPae.txt for data input in file chooser and clicks open

Expected Results: data from file is displayed in new search window

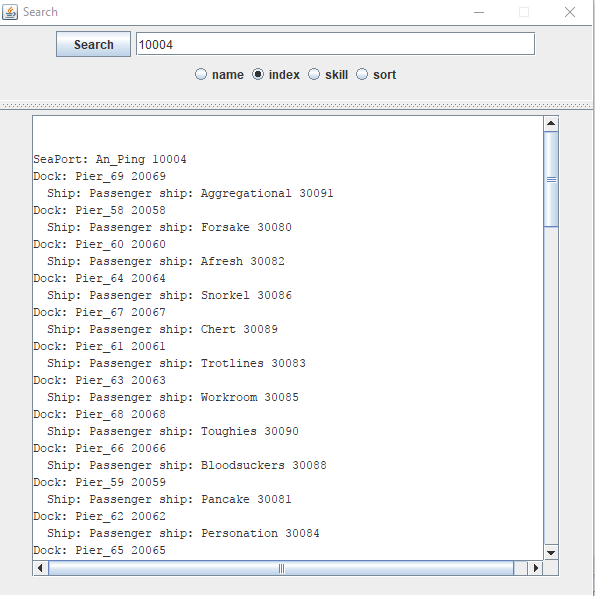
Actual Results: PASS



Input: User searches by index 10004 for seaport

Expected Results: List of Seaport 10004 with docks, ships and persons

Actual Results:

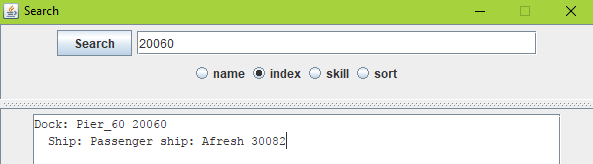


Input: User searches dock 20060

Expected Results:

Dock: Pier\_60 20060

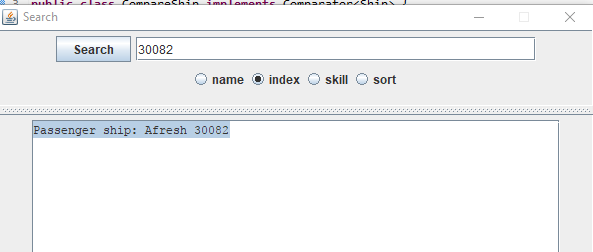
Ship: Passenger ship: Afresh 30082Actual Results:



Input: Users searches index 30082

Expected Results: Passenger ship: Afresh 30082

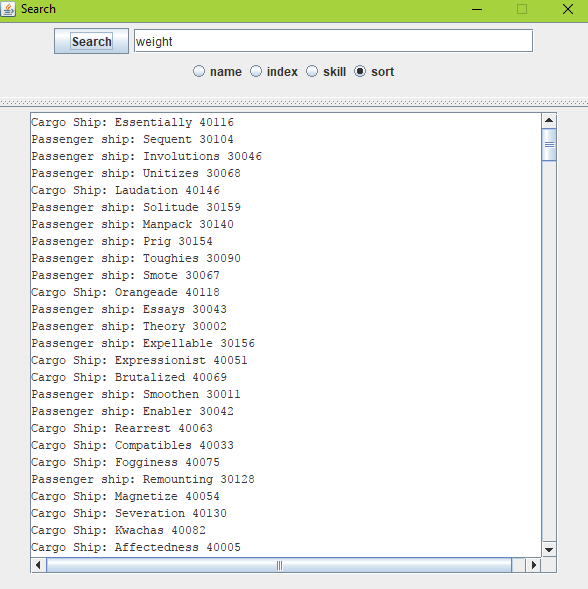
Actual Results:



Input: User sorts by weight

Expected results: Ships are sorted by weight

Actual Results:



Input: User sorts by length

Expected results: Ships are sorted by height

Actual Results:

