CMSC203 Ho Chien, Lin

Project 7A

Program Design/ UML Diagram

|  |
| --- |
| Ship |
| -name:String  -year:String  -type:ShipType |
| +Ship(a:String, b:String, p:String)  +getName():String  +setName(name:String):void  +getYear():String  +setYear(year:String):void  +getType():ShipType  +setType(o:String):void  +writeString():String  +toString():String |

|  |
| --- |
| CargoShip |
| -tons:double |
| +CargoShip(a:String, b:String, p:String, c:double)  +getTons():double  +setTons(tons:double)  +writeString():String  +toString():String |

|  |
| --- |
| CruiseShip |
| -passenger:int |
| +CruiseShip(a:String, b:String, p:String, c:int)  +getPassenger():int  +setPassenger(passenger:int):void  +writeString():String  +toString():String |

|  |
| --- |
| WarShip |
| -guns:int  -aircrafts:int  -torpedoes:int |
| +WarShip(a:String, b:String, p:String, c:int, d:int, e:int)  +getGuns():int  +setGuns(guns:int):void  +getAircrafts():int  +setAircrafts(aircrafts:int):void  +getTorpedoes()  +setTorpedoes(torpedoes:int):void  +writeString():String  +toString():String |

|  |
| --- |
| Cruiser |
| -guns:int  -name:String  -year:String |
| +Cruiser(a:String, b:String, p:String, c:int, d:int, e:int)  +getName():String  +setName(name:String):void  +getYear():String  +setYear(year:String):void  +getGuns():int  +setGuns(guns:int):void  +writeString():String  +toString():String |

|  |
| --- |
| Carrier |
| -name:String  -year:String  -aircraft:int |
| +Carrier(a:String, b:String, p:String, c:int, d:int, e:int)  +getName():String  +setName(name:String):void  +getYear():String  +setYear(year:String):void  +getAircrafts():int  +setAircrafts(aircrafts:int):void  +writeString():String  +toString():String |

|  |
| --- |
| Destroyer |
| -guns:int  -name:String  -year:String |
| +Destroyer(a:String, b:String, p:String, c:int, d:int, e:int)  +getName():String  +setName(name:String):void  +getYear():String  +setYear(year:String):void  +getGuns():int  +setGuns(guns:int):void  +writeString():String  +toString():String |

|  |
| --- |
| Mine\_Sweeper |
| -guns:int  -name:String  -year:String |
| +Mine\_Sweeper(a:String, b:String, p:String, c:int, d:int, e:int)  +getName():String  +setName(name:String):void  +getYear():String  +setYear(year:String):void  +getGuns():int  +setGuns(guns:int):void  +writeString():String  +toString():String |

|  |
| --- |
| Submarine |
| -torpedoes:int  -name:String  -year:String  -count:int |
| +Submarine(a:String, b:String, p:String, c:int, d:int, e:int)  +getName():String  +setName(name:String):void  +getYear():String  +setYear(year:String):void  +getTorpedoes()  +setTorpedoes(torpedoes:int):void  +writeString():String  +toString():String  +getCount():int  +setCount(count:int):void |

|  |
| --- |
| ShipRegistry |
| -ship: ArrayList<Ship> |
| +ShipRegistry()  +addShip(name:String, type:String, year:String, tons:int, pax:int, guns:int, torpedoes:int, aircraft:int):void  +readFile(file:File):void  +writeFile(file:File):void  +writeNode():String  +printShip():String |

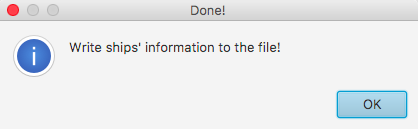
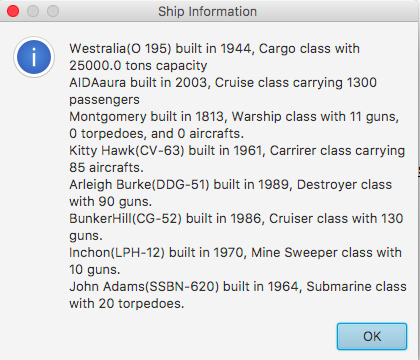
Algorithm

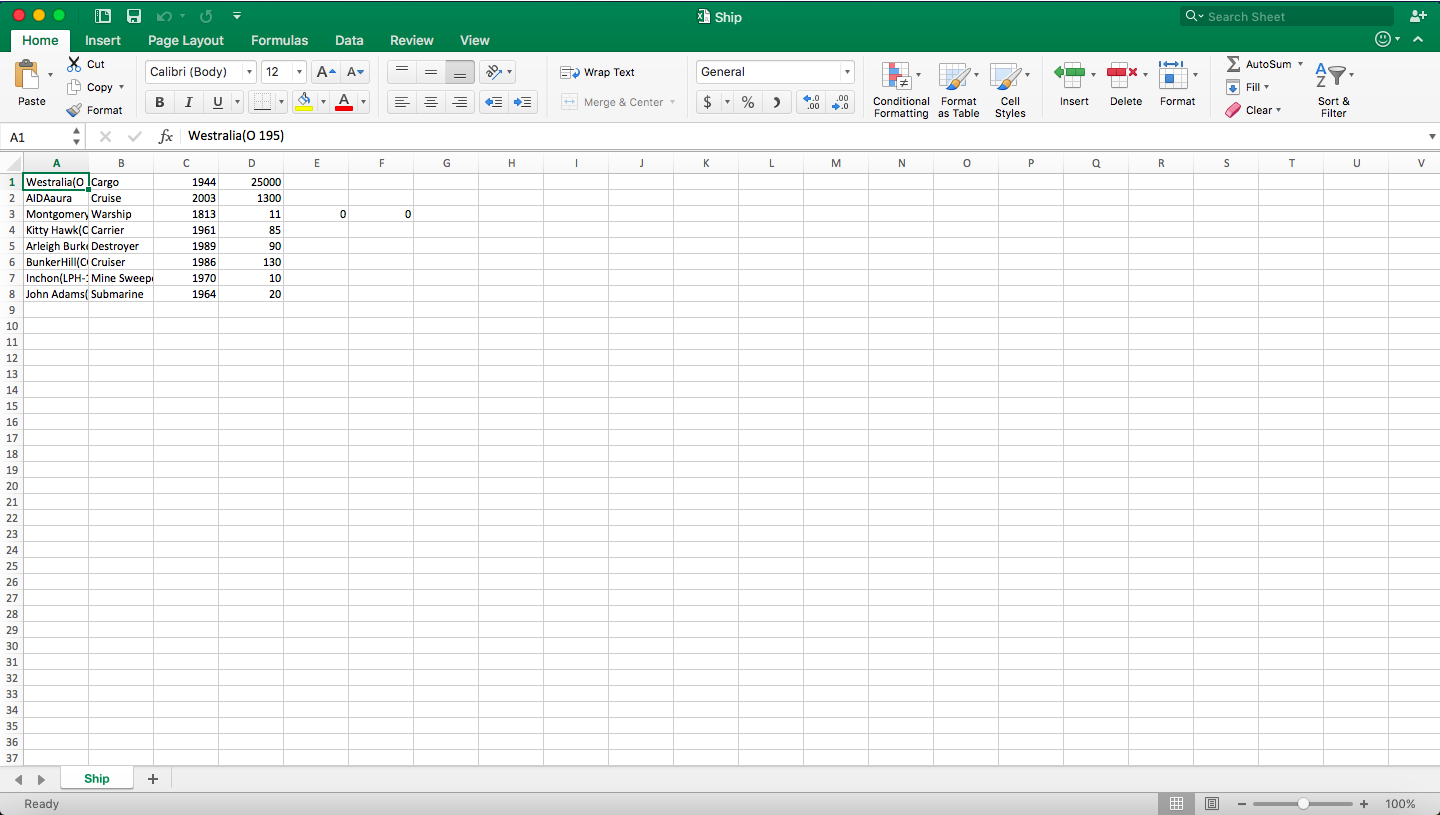
1. Create Ship Class
2. Create CargoShip, CruiseShip and WarShip Class
3. Create Cruiser, Carrier, Mine\_Sweeper, Destroyer and Submarine Class
4. Create ShipRegistry Class

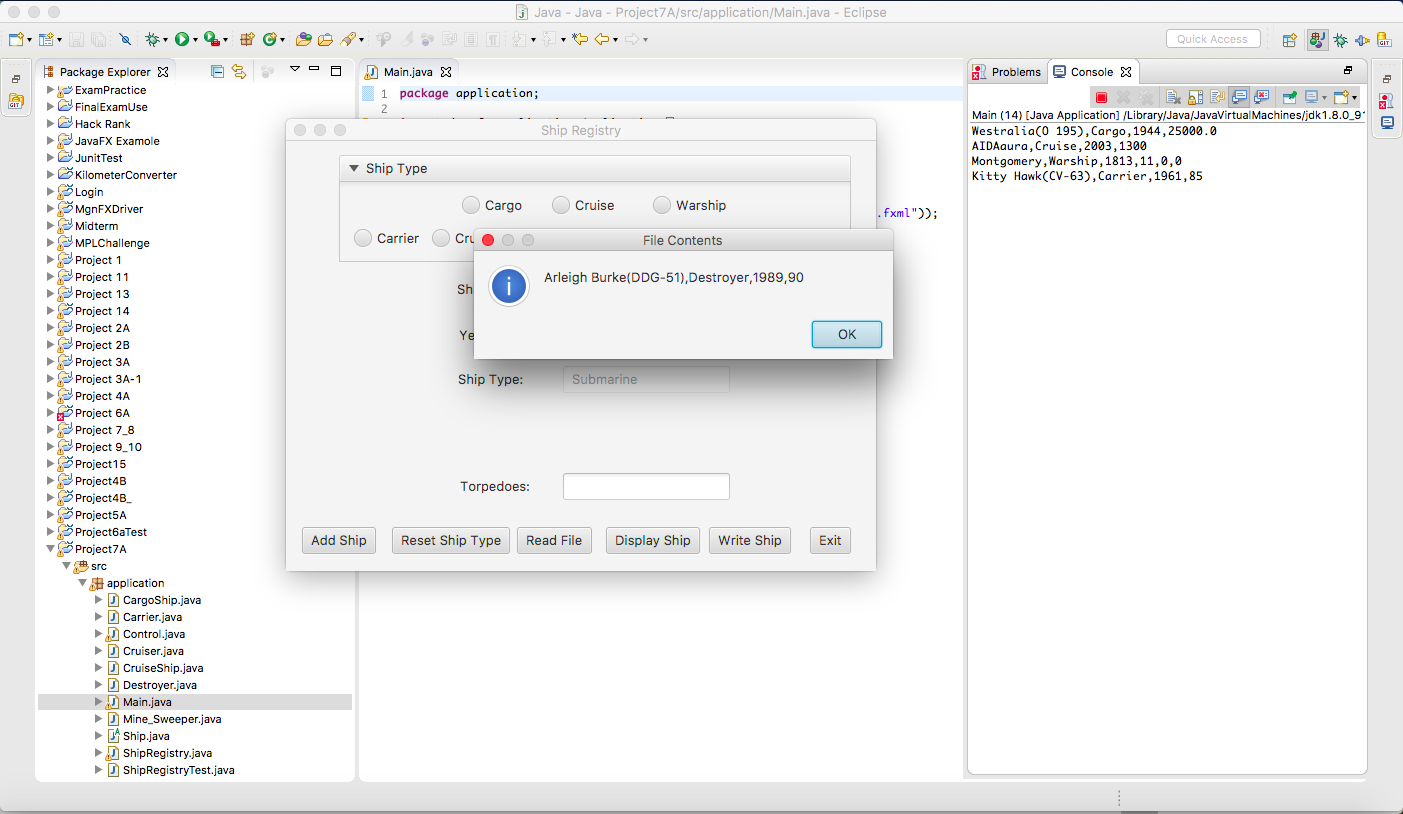
Test Case

|  |
| --- |
| Data |
| "Westralia (O 195)", "Cargo", "1944", 25000, 0, 0, 0, 0  "AIDAaura", "Cruise", "2003", 0, 1300, 0, 0, 0  "Montgomery", "Warship", "1813", 0, 0, 11, 0, 0  "Kitty Hawk (CV-63)", "Carrier", "1961", 0, 0, 0, 0, 85  "Arleigh Burke (DDG-51)", "Destroyer", "1989", 0, 0, 90, 0, 0  "Bunker Hill (CG-52)", "Cruiser", "1986", 0, 0, 130, 0, 0  "Inchon (LPH-12)", "Mine Sweeper", "1970", 0, 0, 10, 0, 0  "John Adams (SSBN-620)", "Submarine", "1964", 0, 0, 0, 20, 0 |

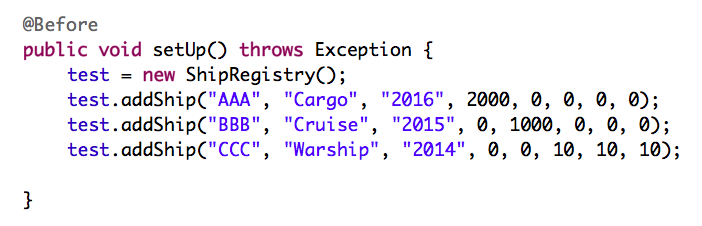
Screen Shots



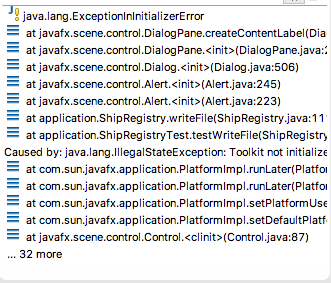




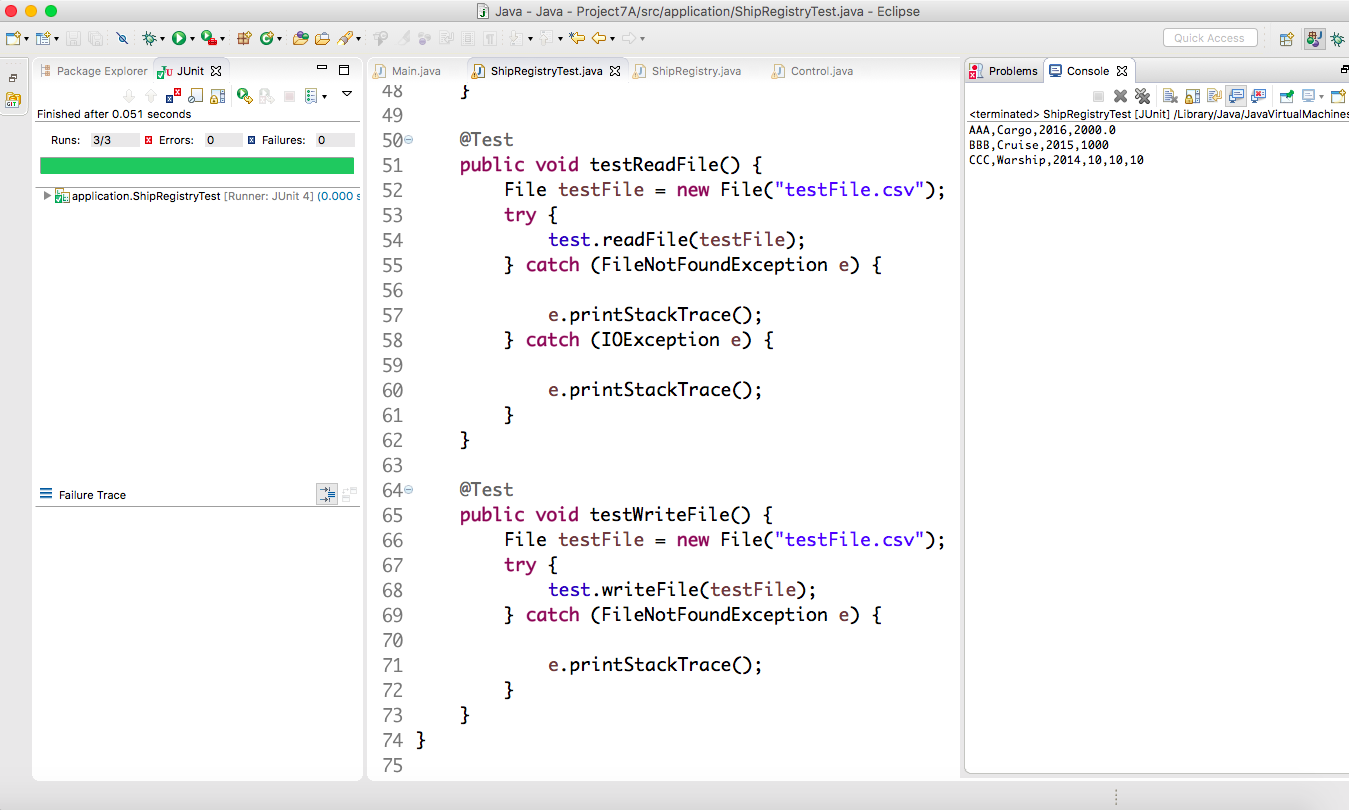
Junit Test Case



Because I add Alert DialogPane in both my writeFile and readFile method. When I run the test, it will reveal this error info.



If I remove Alert DialogPane in both methods, the test will run well.



Assumptions

1. The application is easy for user to use
2. User can choose which file they want to write in

Learning Experience

This is the last project I did in this semester. Can’t image time fly so fast. Combine this project with 6A and 5A. Now, I can understand how to create a JavaFX application and use inheritance to build a program. Like we had finished a simple air reservation application in early semester. I can think how to build this application more useful in real life.

During this semester I had learned many concepts of object-oriented programming. This is very important for developer to design real software application in real life. The last class I learned some basic algorithm and sorting methods. I feel things become more interesting than I previously thought. Even though there will have a lot of challenges in the future.