Java Lab 20: More I/O

This lab practices writing and reading objects, plus a few miscellaneous nio methods.

1. Create project Lab20. Download Lab20.zip; unzip it and copy the class into this project: the Cargo class and its children, the CargoSizeable interface, the CargoEnum, CargoContainer, and Lab20Main, which contains the main program. The Cargo class already implements CargoSizeable; make it also implement Serializable.

2. Add the method public ArrayList<Cargo> CargoContainer.getCargoList( ), that returns its ArrayList (breaks encapsulation).

3. Add a new class named CargoFileOperations with String data member filename and the following methods:

- overloaded constructor that sets filename

- public void writeList(ArrayList<Cargo>). This should declare an ObjectOutputStream that wraps a FileOutputStream; use a try-catch block here. It should write out, as objects, all of the ArrayList, using filename as the parameter to FileOutputStream's constructor and close the file at the end. Each ObjectOutputStream method used should have its own try-catch block with their own error messages (like “Output file failed to open”), so that you know which thing failed (that is, don't use one try-catch block for the whole thing), so print a message in the catch that tells what went wrong and then exit.

- public ArrayList<Cargo> readList( ). This should declare an ArrayList<Cargo>, initially empty. Declare an ObjectInputStream and a FileInputStream as separate variables. Wrap the second inside the first. Write a while loop that reads from filename, checking the FileInputStream variable's available( ) > 0 to quit the loop; close the file at the end. Use try-catch blocks – one around the new’ing of the two streams, and one around the while loop. Don't forget to cast the objects read back to type Cargo. Return the ArrayList.

4. In the main program, after the call to cargoContainer.display(), create a CargoFileOperations object; write out the Cargo items using its writeList( ) method. Then read them back using its readList( ) method, and display the returned objects. You should verify that CargoContainer's display method showed the same data.

5. Add a public void display( ) method to CargoFileOperations. In it, declare a path object using Paths.get( ) on its filename and display) its toString(), absolute path, and getRoot. Declare a File object and display isDirectory( ) and getAbsolutePath( ). Finally, display the return values of these Files methods on the Path variable: isExecutable( ), isReadable(), isWriteable(). Make sure each display has a simple label, as in "isExecutable() returns true". Call display() at the end of main.