Introduction

In this assignment, you will take the roll of an up and coming programming contractor. Your task will be to build a simple application. The script will take the form of a fictional story, to assist in your understanding of the task. This application is detailed in a specification given to you by the story's protagonist. For this scenario, you will need to build up a system of classes and interfaces that meet the requirements that you have been given. For this assignment you will be focusing on accessor methods (getter's and setter's) as well building an understanding of Lists and basic programming logic.

Story

Hello my dear programmer friend,

I have come to understand that you might be in a position to help me. You see, I feel as if I am in a bit of a rut and have decided that I need to make a life altering change. I am going to resign my position as the head of the Soup canning factory and pursue my dreams, taking on the role of a celebrity insect therapist, in a distant country.

[1]From now on, classes and interfaces will be shortened to simply "classes"

Unfortunately, this means that I will have to move house, which is something I have very little experience in doing.

As such, I was hoping that I could enlist your skills to help me. I would like you to build a software application model to help me organise the packing of my life belongings, and to do so as efficiently as humanly possible.

Should you accept, there are a few things that I need you to model:

- To begin with I need something to store all of my items in. I have 3 different kinds of storage containers that I intend to use, and they come in various different sizes.
  - A moving truck that I will take all of my items in.
  - A cardboard box. This multi-purpose storage container be packed with many different items (including other boxes! How cool).
  - A duffle bag. I have a few of these that I will use to pack all of my personal items.
- I have some furniture that I would like to take to the next house with me too. This would go into the moving truck last as they are quite heavy.
- As mentioned before, I have some personal items that I would like to take with me. This includes items like clothes and laptops. I've also got some items that I will be storing for some friends so I need to know who owns what.

Because I am not the best at packing, (and why I am asking you to create this application) sometimes I may make mistakes and pack things in the wrong order. The application will need to be aware of this. Would you be able to implement a system where if I pack too much into a container or put something where it's not supposed to be, that the application will let me know?

Awesome, thanks so much!

- Protagonist

Supplied Material

- This task sheet.
- Code specification document (Javadoc).[2]
- Gradescope, a website where you will submit your assignment.[3]
- A starting template for your assignment code, available for download on Blackboard. The files in this template provide a minimal framework for you to work from, and build upon. These files have been provided so that you can avoid (some of) the critical mistakes described in Appendix A.

  Each of these files:
  - is in the correct directory (do not change this!)
  - has the correct package declaration at the top of the file (do not change this!)
  - has the correct public class or public interface declaration. Note that you may still need to make classes abstract, extend classes, implement interfaces etc., as detailed in the Javadoc specification.

[2]Detailed in the `Javadoc` section
[3]Detailed in the `Submission` section

As the first step in the assignment (after reading through the specifications) you should download the template code from Blackboard. Once you have created a new project from the files you have downloaded, you should start implementing the specification.

Javadoc

Code specifications are an important tool for developing code in collaboration with other people. Although assignments in this course are individual, they still aim to prepare you for writing code to a strict specification by providing a specification document (in Java, this is called Javadoc).

You will need to implement the specification precisely as it is described in the specification document.

somewhere, and open `doc/index.html` with your web browser.

Tasks

1. Fully implement each of the classes and interfaces described in the Javadoc.
2. Write JUnit 4 tests for all the methods in the following classes:
   - `Book` (in a class called `BookTest`)
   - `MovingTruck` (in a class called `MovingTruckTest`)