COMP-1502

ASSIGNMENT #2

Dept. of Mathematics & Computing

Mount Royal University
Khosro Salmani

Assignment #2 is prerequisite for assignment #3. We DO NOT provide the code later for the next assignment.

The Problem

In this assignment you will write software to for a toy store. Your program will allow users to find, purchase, and list toys, as well as add and remove toys from the database. You will use object-orients concepts including **inheritance, abstraction, and interfaces**.

Assignment Instructions

1. Use **only** Eclipse IDE (JDK 8 or 11).
2. The due date for this assignment is posted in Blackboard.

Requirements

Your project must have the following features and specifications:

Project Directory Structure

Follow this directory structure:

- bin/ (compiled java files)
- src/ (Java source files)
    - application/ (classes housing the "main" program)
    - controller/ (classes used to coordinate the "model"s and "view"s)
    - model/ (classes related to data objects, like toys)
    - view/ (classes related to user interface)
    - exceptions/ (custom exception classes)
- doc/ generated JavaDoc (Ensure the private option is checked and everything is included in the generated documentation.)
- lib/ any 3rd-party libraries. This folder can be empty, and eclipse should automatically add things here if needed
- res/ any resources or data files
- test/ unit tests

Data Storage

The database of toys should be saved to a file called "toys.txt" in the res/ folder. Each line in the file represents a different toy and each piece of information for a toy is separated by a semi-colon. Your program should read all the toys into a ****single**** ArrayList on startup, and save the (potentially modified) list of toys back to the file on exit.

A sample **`toys.txt`** file is provided for you.

Menu Options

- **Search Inventory and Purchase Toy**
    - Allows users to search for a toy by serial number (**SN**), **name**, or **type**. The program then shows matching products, along with current inventory counts for each. The customer can select one of the items to purchase **or** return to the search menu. Purchasing an item modifies the inventory count accordingly, and trying to purchase an item that is out-of-stock shows an error. The customer returns to the search menu after purchasing.
        - NOTE: searching by name returns all items containing the name (see sample runs below)
- **Add a New Toy**
    - Allows user to add a new toy to the database. Each type of toy requires different information (see formatting section).
        - the program should validate the serial number before using it (see toy attributes)
        - the serial number must be unique
- **Remove a Toy**
    - Allows user to remove a toy from the database. The program asks the user to enter the serial number, then shows the corresponding item and asks for confirmation before removing it.
- **Make a gift suggestion (OPTIONAL - WORTH BONUS MARKS!)**
    - Asks user for an age, type, or price range (the customer can leave 1-2 of these empty) and shows a list of item suggestions. The customer can then select an item from the list to purchase.
- **Save and Exit**
    - Saves the database to the text file (in the appropriate format) and terminates the program.

Note that all menus should be **case in-sensitive**, and **validate** user input

Toys and their attributes

There are four types of toys: figures, animals, puzzles, and board games. The toys have the following attributes:

- **Figures**
    - Serial Number: a unique 10-digit number for this figure. The first digit is 0 or 1
    - name: the name of the item
    - brand: the brand name
    - price: the cost to purchase the item
    - available-count: the number copies of this item currently in stock
    - age-appropriate: the suggested minimum age to use this item
    - classification: either **A**ction, **D**oll, or **H**istoric
- **Animals**
    - Serial Number: a unique 10-digit number for this figure. The first digit is 2 or 3

    - name: the name of the item
    - brand: the brand name
    - price: the cost to purchase the item
    - available-count: the number copies of this item currently in stock
    - age-appropriate: the suggested minimum age to use this item
    - material: a description of the material this toy is made from
    - size: **S**mall, **M**edium, or **L**arge
- **Puzzles**
    - Serial Number: a unique 10-digit number for this figure. The first digit is 4, 5, or 6
    - name: the name of the item
    - brand: the brand name
    - price: the cost to purchase the item
    - available-count: the number copies of this item currently in stock
    - age-appropriate: the suggested minimum age to use this item
    - puzzle-type: **M**echanical, **C**ryptic, **L**ogic, **T**rivia, or **R**iddle
- **Board Games**
    - Serial Number: a unique 10-digit number for this figure. The first digit is 7, 8, or 9
    - name: the name of the item
    - brand: the brand name
    - price: the cost to purchase the item
    - available-count: the number copies of this item currently in stock
    - age-appropriate: the suggested minimum age to use this item
    - number of players: a range, like 2-4 (note your program must store min and max players in separate properties
    - designer(s): a comma-separated list of designers

Use of Inheritance and Object-Oriented structure

- You must create a class for each toy type (located in the model package), and each must inherit from a `Toy` superclass which **cannot be instantiated** and has all the common attributes.
- Each toy class should have a constructor that accepts all the attribute values for that toy
- Each toy class should override the `toString()` method, so that the object's details can be printed in human-readable form

Testing

We expect to see reasonable JUnit tests written to ensure that the behavior of the existing classes (`Animal, BoardGame, Figure`, …)

Exceptions

The program should throw (custom) exceptions if:

- The input price is negative when the user creates a new toy
- The minimum number of players is greater than the maximum, when adding a new board game

Submission Instructions

- Push all git commits before the due date. The contribution of each team member must be visible in Git history. Paste the link to your github repo into the D2L assignment to finalize your submission.
- You must demo your project to your instructor during a demonstration to be scheduled during tutorial/lab time. This demo is mandatory. **No demo = a grade of 0**

Team work

- Before starting, understand the requirements and meet with your project partner to divide tasks equally. Assign each task to a team member in a planning document in your repository (a readme file would be a good place).
- Both team members must contribute to the git project. The contribution of each member must be visible in the git commit history.
- Both members of a team must be knowledgeable about the whole project. This must be evident during the assignment demo. Your instructor may ask questions or make requests of either team member.

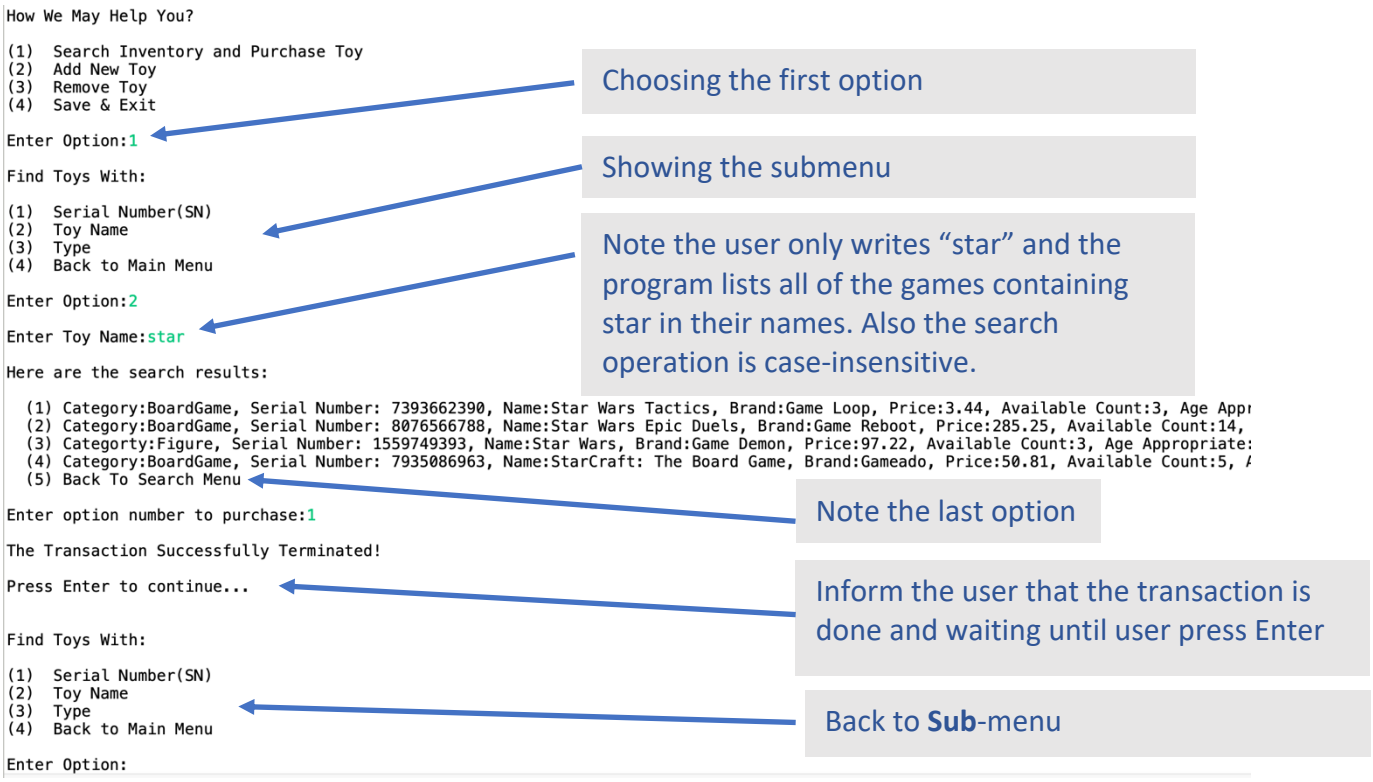
Sample Runs (some) not everything is covered in these images

Main Menu:

```
**************************************************
*        WELCOME TO TOY STORE COMPANY!           *
**************************************************

How We May Help You?

(1)  Search Inventory and Purchase Toy
(2)  Add New Toy
(3)  Remove Toy
(4)  Save & Exit

Enter Option:
```


Welcome page and the main menu

Search:


How We May Help You?
(1) Search Inventory and Purchase Toy
(2) Add New Toy
(3) Remove Toy
(4) Save & Exit
Enter Option:1
Find Toys With:
(1) Serial Number(SN)
(2) Toy Name
(3) Type
(4) Back to Main Menu
Enter Option:2
Enter Toy Name:star
Here are the search results:
(1) Category:BoardGame, Serial Number: 7393662390, Name:Star Wars Tactics, Brand:Game Loop, Price:3.44, Available Count:3, Age Appr
(2) Category:BoardGame, Serial Number: 8076566788, Name:Star Wars Epic Duels, Brand:Game Reboot, Price:285.25, Available Count:14,
(3) Categorty:Figure, Serial Number: 1559749393, Name:Star Wars, Brand:Game Demon, Price:97.22, Available Count:3, Age Appropriate:
(4) Category:BoardGame, Serial Number: 7935086963, Name:StarCraft: The Board Game, Brand:Gameado, Price:50.81, Available Count:5, A
(5) Back To Search Menu
Enter option number to purchase:1
The Transaction Successfully Terminated!
Press Enter to continue...
Find Toys With:
(1) Serial Number(SN)
(2) Toy Name
(3) Type
(4) Back to Main Menu
Enter Option:
Choosing the first option
Showing the submenu
Note the user only writes “star” and the program lists all of the games containing star in their names. Also the search operation is case-insensitive.
Note the last option
Inform the user that the transaction is done and waiting until user press Enter
Back to **Sub**-menu

Adding a new Toy

```
How We May Help You?

(1)  Search Inventory and Purchase Toy
(2)  Add New Toy
(3)  Remove Toy
(4)  Save & Exit

Enter Option:2

Enter Serial Number:9999999999

Enter Toy Name:My Toy

Enter Toy Brand:My Brand

Enter Toy Price:24.99

Enter Available Counts:5

Enter Appropriate Age:5

Enter Minimum Number of Players:3

Enter Maximum Number of Players:5

Enter Designer Names(Use ',' to separate the names if there is more than one name):Khosro Salmani, John

New Toy Added!

Press Enter to Continue...

How We May Help You?

(1)  Search Inventory and Purchase Toy
(2)  Add New Toy
(3)  Remove Toy
(4)  Save & Exit

Enter Option:
```

Adding a new Toy!

Removing a Toy

```
How We May Help You?

(1)  Search Inventory and Purchase Toy
(2)  Add New Toy
(3)  Remove Toy
(4)  Save & Exit

Enter Option:3

Enter Serial Number:9999999999
This Item Found:

    Category:BoardGame, Serial Number: 9999999999, Name:My Toy, Brand:My Brand, Price:24.99, Available Count:5, Age Appropriate:5 Min

Do you want to remove it (Y/N)?y

Item Removed!

Press Enter to continue...

How We May Help You?

(1)  Search Inventory and Purchase Toy
(2)  Add New Toy
(3)  Remove Toy
(4)  Save & Exit

Enter Option:
```

Removing a Toy!

Save and Exit

```
How We May Help You?

(1)  Search Inventory and Purchase Toy
(2)  Add New Toy
(3)  Remove Toy
(4)  Save & Exit

Enter Option:4

Saving Data Into Database...


********** THANKS FOR VISITING US! **********
```

Save and Exit!

Input Validation

```
*****************************************************
*        WELCOME TO TOY STORE COMPANY!              *
*****************************************************

How We May Help You?

(1)  Search Inventory and Purchase Toy
(2)  Add New Toy
(3)  Remove Toy
(4)  Save & Exit

Enter Option:q

This is Not an Integer Number! Try again.


Enter Option: 7

This is not a valid option! Try again.

Enter Option:
```

```
*****************************************************
*        WELCOME TO TOY STORE COMPANY!              *
*****************************************************

How We May Help You?

(1)  Search Inventory and Purchase Toy
(2)  Add New Toy
(3)  Remove Toy
(4)  Save & Exit

Enter Option:2

Enter Serial Number:1147205649

A Toy With This Serial Number Already Exists! Try Again.

Enter Serial Number:11qqq5649

The Serial Number should only contain digits! Try again.


Enter Serial Number:123456789

The Serial Number's length MUST be 10 digits! Try again.


Enter Serial Number:
```

Marking Rubric

Item	Points
Program requirements met (menus, game, reports work as expected)	40
Object-oriented structure (Problem broken down into classes/methods. Correct use of inheritance/abstraction)	20
Documentation & comments (Javadoc, comments, appropriate/conventional names, etc)	20
Unit testing (tests work properly, good test coverage)	20
TOTAL	**100**

** The maximum mark for this assignment is 100 and the bonus can boost your mark if you lost any marks in other items only in this assignment.