

CSE 1322L - Lab 11

Introduction

In this lab, you will write a program that allows a library to schedule boxes of books to be delivered to students. As students are entered into the program, they are added into a queue, which will determine the order that their boxes are delivered. The books in the boxes are put together as a stack: the first book to be put in will be at the bottom whereas the last book will be at the top. While the system will start with some pre-made boxes, it must give the user the option to create new boxes from scratch.

Requirements

The features described below must be in your program:

- **BoxOfBooks** class: this class stores the names of the books which the box holds. It behaves like a stack.
 - A String field called "boxName"
 - An arraylist of Strings called "books". This field will hold the contents of our stack: the first element is always the top of the stack
 - **BoxOfBooks(String)**: initializes "books" and assigns the argument to "boxName"
 - "boxName" has a getter
 - **void addBook(String)**: works as a stack's push(), adding the argument to the first position of "books"
 - **String retrieveBook()**: works as a stacks's pop(), removing and returning the element at the first position of "books". If "books" is empty, return null instead
 - **BoxOfBooks copy()**: Returns a new BoxOfBooks whose "name" and the contents of its "books" matches this object's "name" and "books"
- **Student** class
 - A String field called "name"
 - A BoxOfBooks field called "boxOfBooks"
 - **Student(String, BoxOfBooks)**: initializes both fields with the given arguments
 - **String unpackBoxOfBooks()**: pops all the elements in "boxOfBooks", concatenating them into a single String as they are popped, which is then returned. The elements must be separated by a new line
 - **String toString()**: this override returns the following string (replace the curly braces and their contents with the appropriate information):

```
Student: {name}
Books:
{unpackBoxOfBooks() }
```

- QueueOfStudents class: this class stores our students as they are added to the system. It behaves like a queue
 - An arraylist of Students called “students”. This field holds the contents of our queue: the first element is the head of the queue; the last element is its tail
 - **QueueOfStudents()**: initializes “students”
 - **void registerForPickup(Student)**: works as a queue’s enqueue(), adding the argument to the back of “students”
 - **Student deliverBoxOfBooks()**: works as a queue’s dequeue(), removing and returning the element at the front of “students”. If “students” is empty, return null instead
- Driver class
 - Create an arraylist of BoxOfBooks called “boxes”
 - Add the BoxOfBooks below to “boxes”. The BoxOfBooks’ name is followed by its contents, which must be pushed in the order they appear below:
 - Programming: “Databases made Simple”, “Macros and Scripts”, “Programming Basics”
 - Chemistry: “Organic Chemistry”, “Inorganic Chemistry”, “The Periodic Table of Elements”
 - Mathematics: “Linear Algebra”, “Calculus I and II”, “Pre-Calculus”
 - Create a QueueOfStudents called “queueOfStudents”
 - In a loop, implement the menu options below:
 1. **Add student**: Prompt the user for the name of the student. Print the name of all the BoxOfBooks inside “boxes” then ask the user which BoxOfBooks they would like to give to the student. Create a Student object with the name above and a copy of the BoxOfBooks they requested. Finally, enqueue the Student in “queueOfStudents”
 2. **Create box of books**: Prompt the user for the name of the box and then create a BoxOfBooks object with that name. Add said box to “boxes”. In a loop, keep prompting the user for names of books, pushing them into the newly created object. If the user enters “-done”, return to the main menu
 3. **Deliver all boxes**: Dequeue and print all students in “queueOfStudents”, then terminate the program

Deliverables

- Lab11.java (driver)
- BoxOfBooks.java
- Student.java
- QueueOfStudents.java

Considerations

- Despite what the deliverables above say, you can submit all 4 classes in a single file.
- Some of the code in your driver might throw exceptions. While you won't be explicitly required to catch unchecked exceptions, you should strive to do so to make your program more robust.
- Normally, BoxOfBook's copy() would instead be an override of clone(), which is an [Object method](#). You are being instructed to do copy() to simplify the lab
 - What bug could you run into if, instead of giving a Student a copy() of a BoxOfBooks, you assigned to their "boxOfBooks" the exact box inside of "boxes"?

Sample Output (user input in red)

[Book Delivery Scheduler]

1. Add student
2. Create box of books
3. Deliver all boxes

Enter option: **1**

Enter name of student: **Alice**

0: Programming

1: Chemistry

2: Mathematics

Give them which box? **1**

Student added to queue.

1. Add student
2. Create box of books
3. Deliver all boxes

Enter option: **1**

Enter name of student: **Bob**

0: Programming

1: Chemistry

2: Mathematics

Give them which box? **0**

Student added to queue.

1. Add student
2. Create box of books
3. Deliver all boxes

Enter option: **2**

Enter box name: **Music**

Enter name of book or '-done': **Harmonies and Cadences**

Book added to box.

Enter name of book or '-done': **Major and Minor Scales**

Book added to box.

Enter name of book or '-done': **Reading Music Sheets**

Book added to box.

Enter name of book or '-done': **-done**

Box of Books registered for future use.

1. Add student

2. Create box of books

3. Deliver all boxes

Enter option: **1**

Enter name of student: **Charlie**

0: Programming

1: Chemistry

2: Mathematics

3: Music

Give them which box? **3**

Student added to queue.

1. Add student

2. Create box of books

3. Deliver all boxes

Enter option: **1**

Enter name of student: **David**

0: Programming

1: Chemistry

2: Mathematics

3: Music

Give them which box? **2**

Student added to queue.

1. Add student

2. Create box of books

3. Deliver all boxes

Enter option: **3**

Here are all the students and the contents of their boxes:

Student: Alice

Books:

The Periodic Table of Elements

Inorganic Chemistry

Organic Chemistry

Student: Bob

Books:

Programming Basics

Macros and Scripts

Databases made Simple

Student: Charlie

Books:

Reading Music Sheets

Major and Minor Scales

Harmonies and Cadences

Student: David

Books:

Pre-Calculus

Calculus I and II

Linear Algebra

Shutting off...