CMPS 251 Tutorial 2 – Fall 2019

Exercise 1 - Grade Book App

Develop a Grade Book App to maintain students grades on an exam and display a grade report that includes the grades, class average, lowest grade and highest grade.

1. Create a Java package named `qu.gradebook` to place all the classes of this app.
2. Create **Course** class as per the class diagram in Figure 1. Provide getters and setters for each of the class attributes.
3. Create **Student** class as per the class diagram in Figure 1. Provide getters and setters for each of the class attributes.
4. Create **GradeBook** class as per the class diagram in Figure 1. Provide a getter and a setter for the course attribute. Implement the following methods:

`void addStudent(Student student)`	Adds a student to students list
`double getMinimum()`	Returns the highest grade
`double getMaximum()`	Returns the lowest grade
`double getAverage()`	Returns the average grade

`void displayGrades()`	Displays the course grades. Example display:
	`The grades for CMPS 151 Programming Concepts`
	`Id Firstname Lastname Grade`
	`-- --------- -------- -----`
	`1 Firstname1 Lastname1 27.46`
	`2 Firstname2 Lastname2 86.55`
	`3 Firstname3 Lastname3 1.86`
	`4 Firstname4 Lastname4 79.17`
	`5 Firstname5 Lastname5 31.51`

5. Test your implementation using ***GradeBookTest.java*** provided @ https://gist.github.com/erradi/cf16b6801c1175bc3e467a5cae68a01a

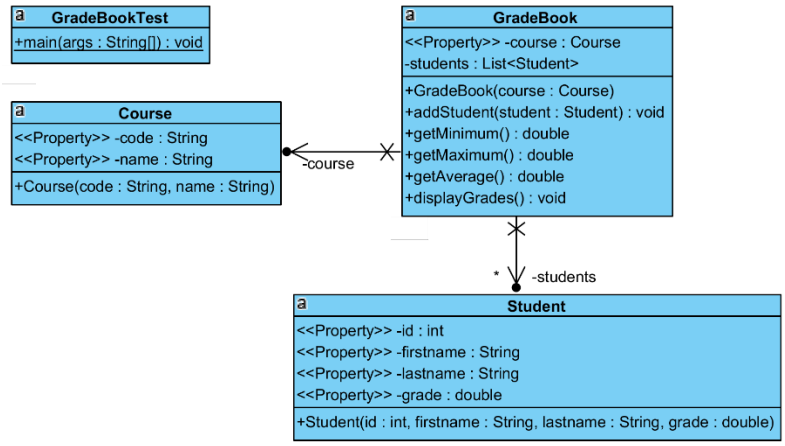

Figure 1. GradeBook App Class Diagram

Exercise 2 - Banking App

Develop a Banking App to manage bank accounts.

1. Create a Java package named `qu.bank` to place all the classes of this app.
2. Create **Account** class as per the class diagram in Figure 2. Provide getters and setters for each of the class attributes.

3. Create **Bank** class as per the class diagram in Figure 2. Note that all attributes and methods of this class are static. Implement the following methods:

void addAccount(Account account)	Adds an account to accounts list
Account getAccount(int acctId)	Get an account by acctId
double getBalance(int acctId)	Get an account balance by acctId
String deposit(int acctId, double amount)	Deposit an amount into an acct and returns a confirmation message.
String withdraw(int acctId, double amount)	Withdraw an amount from an acct and returns a confirmation message.
void addTestAccounts()	Implementation @ https://gist.github.com/erradi/17a43ba0de49ee351979d1b7ba5f1ce3
String getFormattedBalance(acctId)	Implementation @ https://gist.github.com/erradi/9d5190ccbc12487db39ca2b340879d7a

4. Test your implementation using ***BankUI.java*** provided @ https://gist.github.com/erradi/899c940781eae775f4e9a35e68608a08

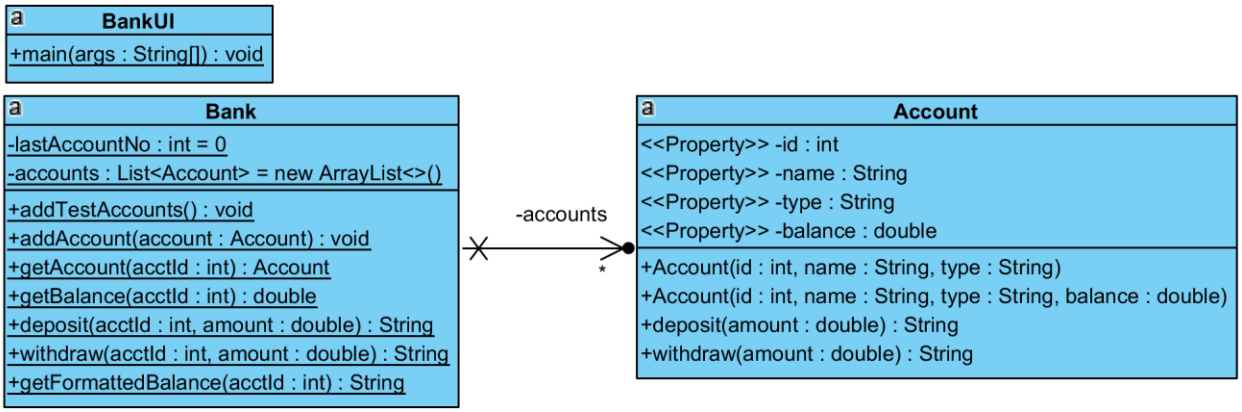

Figure 2. Bank App Class Diagram

Exercise 3 – Book Store App

Develop a Book Store App to manage buying books.

1. Create a Java package named `qu.bookstore` to place all the classes of this app.
2. Create **Book** class as per the class diagram in Figure 3. Provide getters and setters for each of the class attributes.
3. Create **`CartItem`** class as per the class diagram in Figure 3. Provide getters and setters for each of the class attributes. Implement *getTotal()* method as `quantity * book.getPrice()`
4. **`BookCatalog`** implementation is provided @ https://gist.github.com/erradi/026d57f4c7f713aa27ba69f484ab0732
5. Create **`ShoppingCart`** class as per the class diagram in Figure 3. Implement the following methods:

void addItem(CartItem cartItem)	Adds cartItem to cartItems list
double getTotal()	Get the total of items in the cartItems list

6. Test your implementation using ***BookStoreUI.java*** provided @ https://gist.github.com/erradi/ebb34b6f99ddcfda34aa0327233033dc

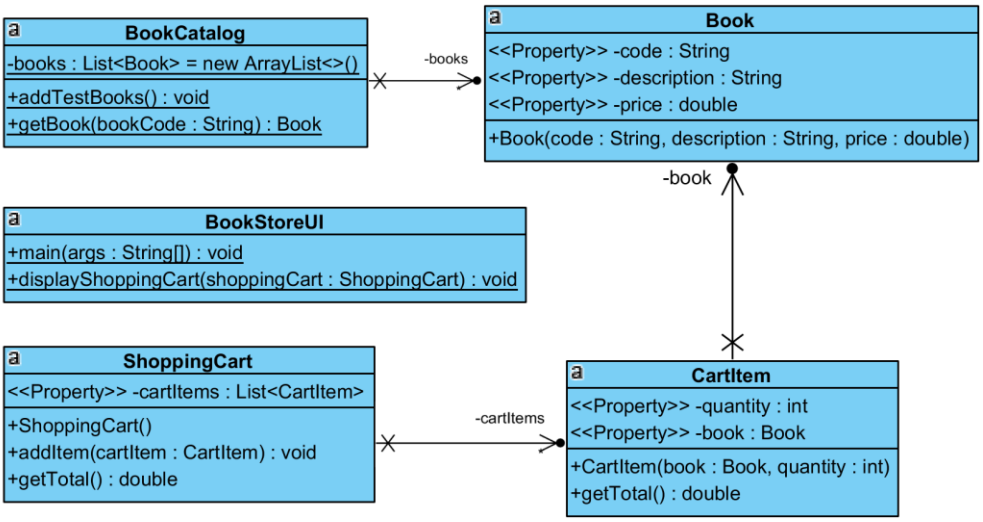

Figure 3. Book Store App Class Diagram