Student Name: _____

Exam - after 2 weeks - EXAMPLE 1

Time: 2 hours
Closed books! if you forgot something and need help, please ask the instructor.

Question 1

Implement a class named “ATM” with the following data:

(1) Id of the ATM - String
(2) Current amount of bills for each of the possible bills: 20, 50, 100, 200
    (a) Create an enum for Bill to represent the different bills
    (b) Each bill would be identified by an Enum value representing the bill
    (c) For each enum value, the ATM should manage the amount of such bill

Please make sure to use proper access modifiers (private, public etc.)

ATM should implement the interface Comparable<ATM>

- Comparing two ATMs, is done according to the monetary value of the bills they hold

ATM should also implement the methods:

- Map<Bill, Integer> getBills();
- int getBillCount(Bill bill);
- List<Bill> getAvailableBills(); -- returns only the Bills that have amount > 0
- long totalMoneyValue();

Implement also the following static method:

[2] shouldFill

This method gets as an input:

- ArrayList<ATM> -- a list of ATMs

The method shall return as an output some representation of all ATMs that need to be refilled (e.g. the ID of the ATM or any other representation that you would choose) and for each - the amount of bills to fill per each kind of bill. This should be done based on a static representation inside class ATM for the minimal required amount in an ATM, for each bill.

Question 2

Implement a Stack class for ints.
Stack shall have the following methods:

- Ctor getting capacity for the Stack
- void push(int value) throws FullStackException
- int pop() throws EmptyStackException

Implement now a BetterStack class, inheriting from Stack.
BetterStack shall add the following methods:

- int getSum()
- double getAverage() throws EmptyStackException

Question 3

Implement a class Book, holding: name, author, catalog number (long)
Implement a class BookInstance, holding: Book, instance id (int), ref to Borrower
Implement a class Person, holding: name, id (long)
Implement a class Borrower, inheriting from Person, holding: a Map of borrowed books (BookInstance) mapping to Date borrowed (java.util.Date)

Implement a class Library, holding:

- Books: a Map of Book to List<BookInstance>
- Borrowers: a Map of borrower id to List<BookInstance>

Implement in class Library a method:

- forAllBorrowers(ApplyOnBorrower apply);

- ApplyOnBorrower is an interface you need to create, with the following method:
    - void applyFunction(Borrower b);

Present two usages for the method forAllBorrowers:

- Print for all borrowers: name of borrower: number of books currently borrowed
- Print the names of all borrowers who hold more than 4 books

You can use anonymous inner class!