King Saud University
College of Computer and Information Sciences
CSC111
Lab Final Exam

Write a Java program that manages bills in a billing system. Here are the UML diagrams:

Bill	3 points
- id: int - status: String - amount: double - balance: double	1
+ Bill() + Bill(id: int, status: String, amount: double, balance: double) + setters + getters	0.5 0.5 0.5 0.5

BillingSystem	8 points
- bills[]: Bill - numOfBills: int + MAX_SIZE: public static final int	1
+ BillingSystem () + getNumOfBills(): int + addBill(b: Bill): boolean + removeBill(i: int): void + displayBill(i: int): void + findBill(id: int): int + addBalance(id: int, amount: double): void + payBill(id: int): boolean	1 0.5 1 1 0.5 1 1 1

TestBilling	4 points
+ main(): void	4

The Class: Bill

As shown in the UML diagram, write the class Bill that has the attributes:

- **id**: the id of the bill
- **status**: the status of the bill: "**Paid**" or "**NotPaid**".
- **amount**: the amount of the bill.
- **balance**: represents the money available to pay the bill

The methods of this class are:

- **Bill**(): A default constructor that sets id to -1, status to "**NotPaid"**, amount and balance to 0.
- **Bill**(id, status, amount, balance): A constructor that initializes new bill with the initial values from the user.
- **Setter methods** (one for each): That sets the values for: id, status, amount, and balance.

- **Getter Methods** (one for each): That returns the values for: id, status, amount, and balance.

The Class: BillingSystem

The class **BillingSystem** contains an array of objects **bills[ ]** that contains all the bill on a company. It also contains the variable **numOfBills** that represents the current number of bills stored in the array of objects **bills[]**.

Note that the maximum number of bills in the list is 100 (hint: **MAX_SIZE** constant).

The methods of this class are:

- `BillingSystem()`: a constructor that initializes the attributes and creates an array of bills of size 100
- `getNumOfBills()`: returns the current number of bills stored in the array.
- `addBill (Bill b)`: this method will add a bill. If it is not possible to add the bill, you should print an error message "**ERROR Adding**".
- `removeBill(int i)`: remove the bill with index 'i' and put the last bill in its place. Otherwise print "**ERROR Removing**".
- `displayBill(int i)`: displays the details for the bill with index 'i' in the array, otherwise print "**ERROR Printing**". It should display the bill information as follows:

```
The bill id: 1234 with amount: 900.0SR is NotPaid. The balance is 1200.0SR.
```

- `findBill(int id )`: returns the index of the bill to the given id. If no such bill exist then return -1.
- `payBill(int id )`: if the bill with this id is NotPaid and has enough balance to pay the bill, change the status to Paid, the amount to 0, and decrease the balance by amount. If the id is not found you should print "Bill not found", if you find the id but the status is Paid you should print "Bill is already Paid". If you find the id and the status is NotPaid but you don't have enough money in the balance to pay the amount, you should print "Balance :800 SR is not enough to pay the amount 1200 SR" (Note: the numbers in amount and balance are examples)
- `addBalance (int id , double money)`: find the bill that has the received id and add the received money to the balance of the bill. Print "**ERROR Bill not found! Cannot add balance"** if the id is not found. If the money was added correctly print "**The amount: 1000.0 is added to your balance. the new balance is 1500.0" (note: the number 1000 and 1500 are just examples)**

The Main Class: TestBilling

- **main**: the main method does the following:
  1. Create a **BillingSystem** object.
  2. (1 point) Ask the user to enter bill information. Keep reading and storing bill information in array bills until user enters (-1) as id of bill.
  3. (1 point) Ask the user to enter the id of the bill he wants to pay then display the output of this operation.

4. (0.5 points) ask the user if he wants to add balance
    a. if (yes) ask the user to enter id and amount to add to the balance then display the output of the operation.
    b. If (no), print "OK – You don't want to add balance".
5. (1 point) Finally, display all bills.

Sample run:

```
enter id (-1 to exit) , status , amount , balance: 1234 NotPaid 1200 600
enter id (-1 to exit) , status , amount , balance: 2222 Paid 1000 2000
enter id (-1 to exit) , status , amount , balance: 3333 NotPaid 1400 1600
enter id (-1 to exit) , status , amount , balance: -1
Do you want to add balance?: (1-Yes/2-No): 1
Please enter the id and the amount of money you want to add: 1234 1100
The amount: 1100.0 is added to your balance. the new balance is: 1700.0
enter the id of the bill you want to pay : 1234
Done!
The bill id: 1234 with amount: 0.0SR  is Paid. The balance is 500.0SR.
The bill id: 2222 with amount: 1000.0SR  is Paid. The balance is 2000.0SR.
The bill id: 3333 with amount: 1400.0SR  is NotPaid. The balance is 1600.0SR.
```

Sample run:

```
enter id (-1 to exit) , status , amount , balance: 1234 NotPaid 1200 600
enter id (-1 to exit) , status , amount , balance: 2222 Paid 1000 2000
enter id (-1 to exit) , status , amount , balance: 3333 NotPaid 1400 1600
enter id (-1 to exit) , status , amount , balance: -1
Do you want to add balance?: (1-Yes/2-No): 2
OK – You don't want to add balance:
enter the id of the bill you want to pay : 1234
The balance: 600.0 cannot cover the bill: 1200.0
The bill id: 1234 with amount: 1200.0SR  is NotPaid. The balance is 600.0SR.
The bill id: 2222 with amount: 1000.0SR  is Paid. The balance is 2000.0SR.
The bill id: 3333 with amount: 1400.0SR  is NotPaid. The balance is 1600.0SR.
```

Sample run:

```
enter id (-1 to exit) , status , amount , balance: 1234 NotPaid 1200 600
enter id (-1 to exit) , status , amount , balance: 2222 Paid 1000 2000
enter id (-1 to exit) , status , amount , balance: 3333 NotPaid 1400 1600
enter id (-1 to exit) , status , amount , balance: -1
Do you want to add balance?: (1-Yes/2-No): 2
OK – You don't want to add balance:
enter the id of the bill you want to pay : 2222
Bill is already Paid!
The bill id: 1234 with amount: 1200.0SR  is NotPaid. The balance is 600.0SR.
The bill id: 2222 with amount: 1000.0SR  is Paid. The balance is 2000.0SR.
The bill id: 3333 with amount: 1400.0SR  is NotPaid. The balance is 1600.0SR.
```

Sample run:

```
enter id (-1 to exit) , status , amount , balance: -1
Do you want to add balance?: (1-Yes/2-No): 1
Please enter the id and the amount of money you want to add: 1111 1000
ERROR Bill not found! Cannot add balance
enter the id of the bill you want to pay : 1234
Bill not found!
```