CSE 102 Programming Assignment 2

Due:

Wednesday, 6-April-2022 by 23:59

Deliverables:

The following Java file should be submitted to MS Teams by the due date and time specified above. Submissions received after the deadline will be subject to the late policy described in the syllabus.

- Assignment02_{StudentNumber}.java
- ".git" folder of your assignment's git repository (see additional announcement for how to do this)

Specifications:

Overview: You will continue your program to maintain the account balances for bank customers. **Do not forget your headers with your assignments Git folder, Name and Date information.**

Requirements: Write and modify the following set of classes (All constructors should take parameters in the order given and initialize empty sets)
NOTE: For any parameter value passed to a method that is considered invalid, you should throw an Exception. If a specific Exception is not listed you should use an existing one that is a child of the RuntimeException class:

1. Bank
    a. Attributes
        i. Name: String, Address: String
        ii. Customers, Companies, and Accounts: types of your choosing
    b. Methods
        i. getName(), setName(), getAddress() and setAddress()
        ii. addCustomer(id: int, name: String, surname: String): None – creates a Customer using name passed and adds to the set
        iii. addCompany(id: int, name: String): None – creates a Company using name passed and adds to the set
        iv. addAccount(account: Account): None – adds passed account to set
            1. Note that an Account will be passed to this method, not created inside of this method
        v. getCustomer(id: int): Customer with the passed id
            1. raises CustomerNotFoundException with id parameter if Customer not found
        vi. getCustomer(name: String, surname: String): Customer with the passed name
            1. raises CustomerNotFoundException with name and surname parameters if Customer not found
        vii. getCompany(id: int): Company with the passed id

1. raises CompanyNotFoundException with id parameter if Company not found

viii. getCompany(name: String): Company with the passed name
1. raises CompanyNotFoundException with the name parameter if Company not found

ix. getAccount(accountNum: String): Account with the passed number
1. raises AccountNotFoundException if Account is not found

x. transferFunds(accountFrom: String, accountTo: String, amount: double): Transfers amount from
1. raises AccountNotFoundException if Account is not found
2. raises InvalidAmountException if amount cannot be transferred from accountFrom to accountTo

xi. closeAccount(accountNum: String): None – removes account from the set
1. raises AccountNotFoundException if Account is not found
2. raises BalanceRemainingException if account has a balance greater than 0

xii. toString(): String – displays the bank information in the following format (example shown at the end of this assignment description):
1. first line – Bank name {tab} Bank address
2. All companies first with the first line being one tab then company name and each line after being two tabs followed by account number, rate, and balance separated by tabs
3. All customers first with the first line being the customer name after one tab and each line after being two tabs followed by account number and balance separated by tabs

2. Account
   a. Attributes – no change (balance must be non-negative)
   b. Methods – all methods from Assignment 01 with following modifications
      i. deposit(amount: double): None
         1. raises InvalidAmountException if amount is negative
      ii. withdrawal(amount: double): None – decrease balance by amount
         1. raises InvalidAmountException if amount is negative or remaining balance is not enough
3. PersonalAccount – no change from Assignment 01
4. BusinessAccount – no change from Assignment 01 (rate must be positive)
5. Customer
   a. Attributes
      i. id: int – must be positive
      ii. Name: String
      iii. Surname: String
      iv. PersonalAccounts: Type of your choosing
   b. Methods – all methods from Assignment 01 with following modifications
      i. getId() and setId(id: int)

   ii. openAccount(acctNum: String): None – creates a PersonalAccount for the customer using the account number and adds it to the set
   iii. getAccount(accountNum: String): PersonalAccount with the passed number
      1. raises AccountNotFoundException if Account is not found
   iv. closeAccount(accountNum: String): None – removes account from the set
      1. raises AccountNotFoundException if Account is not found
      2. raises BalanceRemainingException if balance greater than 0
6. Company
   a. Attributes
      i. id: int – must be positive
      ii. Name: String
      iii. BusinessAccounts: Type of your choosing
   b. Methods – all methods from Project 01 with following modifications
      i. getId() and setId(id: int)
      ii. openAccount(acctNum: String, rate: double): None – creates a BusinessAccount for the Company using the account number and passed rate and adds it to the set
      iii. getAccount(acctNum: String): BusinessAccount with num passed
         1. raises AccountNotFoundException if Account is not found
      iv. closeAccount(accountNum: String): None – removes account from the set
         1. raises AccountNotFoundException if Account is not found
         2. raises BalanceRemainingException if balance greater than 0
7. Custom Exceptions (All must be instances of RunTimeException) – unless otherwise noted each exception class should have a single constructor using all attributes in the order they are given
   a. AccountNotFoundException
      i. Additional Attribute – acctNum: String
      ii. toString() “AccountNotFoundException: “ + Account Number
   b. BalanceRemainingException
      i. Additional Attribute – balance: double
      ii. toString() “BalanceRemainingException: “ + balance
      iii. getBalance(): double – returns balance from Exception
   c. CompanyNotFoundException
      i. Additional Attributes – id: int, name: String
      ii. Constructors
         1. One that takes only the id as a parameter and sets the name to null
         2. One that takes only the name as parameter and sets the id to any value
      iii. toString()

1. if name is not null, “CompanyNotFoundException: name - “ + name
2. if name is null, “CompanyNotFoundException: id - “ + id

d. CustomerNotFoundException
   i. Additional Attributes – id: int, name: String, surname: String
   ii. Constructors
      1. One that takes only the id as a parameter and sets the name and surnmname to null
      2. One that takes the name and surname as parameters and sets the id to any value
   iii. toString()
      1. if name/surname is not null, “CustomerNotFoundException: name - “ + name + “ “ + surname
      2. if name/surname is null, “CustomerNotFoundException: id - “ + id

e. InvalidAmountException
   i. Additional Attribute – amount: double
   ii. toString() “InvalidAmountException: “ + amount

Design: Your program does not require a main method. You are only responsible for creating the six (6) classes and five (5) Exceptions described above.
Code: The file you submit will be named Assignment02_{StudentNumber}. You should put all java classes for this assignment inside of this file as discussed in class.
Test: You are responsible for testing your program. It is important to not rely solely on the examples presented in this Assignment description. It would be a very good idea to write your own test cases for this assignment.

Grading:
MS Teams Submission: If anything is ambiguous, it is your responsibility to ask questions. It is also your responsibility to complete this assignment in a timely manner. Questions regarding this assignment will likely not be answered if received after 17:00 on the due date of the assignment.
Quiz: There will be a quiz based on this assignment given on 22-April. The result of this quiz will be used to determine your grade on this assignment. **Note**: if you do not take the quiz, your score on this assignment **will be 0**.
Git Repository: For this assignment, you must provide the git repository folder. You should commit changes to your git repository often but not excessively. Try committing your changes after each unit of work. A unit of work for assignments like these can be writing a class, so you don’t need to commit every time you make little changes, but you are **not** required or limited to make commits only after you finished a class, you can make commits more often. Be sure to add comments for your commits. Commit comments should describe what you have done with that commit in a short, one-line sentence. Don’t forget to make your final commit. It should represent the finished assignment. **Note**: if you do not submit your Git repository folder, your score on this assignment **will be 0**.

Example: The lines of code below were used to create the following output:

```
Bank b = new Bank("My Bank", "My Bank's Address");
b.addCompany(1, "Company 1");
b.getCompany(1).openAccount("1234", 0.05);
b.addAccount(b.getCompany(1).getAccount("1234"))
b.getAccount("1234").deposit(500000);
b.getCompany(1).getAccount("1234").deposit(500000);
b.getCompany(1).openAccount("1235", 0.03);
b.addAccount(b.getCompany(1).getAccount("1235"));
b.getCompany(1).getAccount("1235").deposit(25000);
b.addCompany(2, "Company 2");
b.getCompany(2).openAccount("2345", 0.03);
b.addAccount(b.getCompany(2).getAccount("2345"));
b.getCompany(2).getAccount("2345").deposit(350);
b.addCustomer(1, "Customer", "1");
b.addCustomer(2, "Customer", "2");
Customer c = b.getCustomer(1);
c.openAccount("3456");
c.openAccount("3457");
c.getAccount("3456").deposit(150);
c.getAccount("3457").deposit(250);
c = b.getCustomer(2);
c.openAccount("4567");
c.getAccount("4567").deposit(1000);
b.addAccount(c.getAccount("4567"));
c = b.getCustomer(1);
b.addAccount(c.getAccount("3456"));
b.addAccount(c.getAccount("3457"));
System.out.println(b.toString());
```

```
My Bank  My Bank's Address
   Company 1
      1234  0.05  1000000.0
      1235  0.03  25000.0
   Company 2
      2345  0.03  350.0
   Customer 1
      3456  150.0
      3457  250.0
   Customer 2
      4567  1000.0
```