Software Engineering and Programming Basics - WS2021/22
Assignment 2

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Professorship of Software Engineering

11| 2021

Organisational

Deadline

14.11.2021 - 23:59

Submission

To submit your answers, please use the test item titled 'Submission' in the menu of Assignment 2. Click on 'Start Test'. You can pause the test and have your answers be saved temporarily by clicking on 'Suspend test'. You can continue the same attempt by clicking on 'Continue test'. Click on 'Finish test' to submit your answers. You have two attempts. The latest attempt will be the one that is graded.
You also need to adhere to the General Assignment Instructions.

Questions

Since this is a PVL, it is important that all students are able to access all necessary information. If you have any questions, please ask them in the in the course forum in the thread 'Assignment 2: Questions', which will be created for this purpose.

Task

Your task is to model a simple bank account with some of its basic functionalities. What follows is a description of the class and its methods which you need to implement. Please work with care and accuracy: Use the exact names for the class and the methods that are given on this task sheet, keep the order of the parameters as they appear in the description and make sure that your syntax is correct.

- Create a class called **Account**. An Account has three attributes: An **accountNumber** which is represented by a whole number to uniquely identify the account. A **pin** which is also represented by a whole number and serves as a password. And the **balance** which is the amount of money that is currently stored in the Account.
  - The class Account has one constructor that receives an accountNumber and a pin which has at least four digits (you can assume that only correct numbers will be given).
  - However, due to security reasons, the pin must not be saved as a clear number! To encrypt the pin, calculate its checksum[1] and save the checksum in the pin attribute.
  - The balance of a new Account should be initialized with 0.0.
- The class has public getter methods for all attributes that return the value of the corresponding attribute, i.e. **getAccountNumber**, **getPin**, and **getBalance**.

The class also has the following public methods:

- A method called **credit** which receives an *amount* of money. It adds the money to the current balance of the Account.
- A method called **debit** which receives an *amount* of money and a *pin*. It first checks whether the given pin matches the pin of the account by comparing its checksum to the one stored in the pin attribute. It then needs to check whether the balance of the account is high enough to debit the desired amount of money. If both checks are successful, the desired amount of money is debited and subtracted from

[1] Also known as digit sum: https://en.wikipedia.org/wiki/Digit$_s um$

the Account's balance. In this case, the method returns 'true'. If any of the checks fail and the money can't be debited, return 'false'.

- A method called **transferTo** which receives an Account of a *recipient*. an *amount* of money that should be transferred, and the *pin* of the sender's account for validation. First, like in the debit method, the method needs to check whether the given pin matches the Account's pin by comparing the given pin's checksum with the stored one. It then also needs to check whether the balance of the account is high enough to transfer the desired amount of money. If both checks are successful, the money is transferred to the recipient Account. That means the amount needs to be subtracted from the calling Account's balance and added to the balance of the recipient Account. If the transfer is successful, return 'true'. If any of the checks fail and the money cannot be transferred, return 'false'.

Example

```
Account petersAccount = new Account(1010101, 4567);
petersAccount.getAccountNumber(); // returns 1010101
petersAccount.getPin(); // returns 22

petersAccount.credit(5000.0);
petersAccount.getBalance(); // returns 5000.0

petersAccount.debit(150.0, 4567); // returns true
petersAccount.debit(150.0, 4577); // returns false (wrong pin)
petersAccount.debit(5500.0, 4567); // returns false (not enough money on the account)
petersAccount.getBalance(); // returns 4850.0 (150 were debited once)

Account sallysAccount = new Account(1010102, 9984);

petersAccount.transferTo(sallysAccount, 563.5, 4567); //returns true
petersAccount.getBalance(); //returns 4286.5
sallysAccount.getBalance(); //returns 563.5
```