Java Programming

Bauhaus-Universität Weimar

28. März 2019

Task 1
Have a look at the file `ExceptionHandling.java`. The class throws multiple exceptions on execution. Write `try-catch` blocks to handle those exceptions.
Have a closer look on the last lines of code. When trying to compute the squareroot of a negative number, no exception is thrown. Write your own exception for this case. Then, write a method that tries to compute the squareroot of a given input but throws your custom exception when the input is negative.

Task 2
Write a class `BankAccount`. This class should consist of the following:

1. `final String bic = ”ABCDDEFFXXX”` to represent the BIC of the account.
2. `String iban` to represent the IBAN of the account. By default the IBAN should be *DE00 1234 5678 0000 0000 00*.
3. `int securityNumber` to represent the PIN code.
4. `Person owner` to represent the owner of the bank account.
5. `double limit` to represent the limit of the account.
6. `double balance` to represent the current balance.
7. A constructor setting the owner, balance and limit of the account. In the constructor also the security number and the last 10 digits of the IBAN are set. You can use `new Random().nextInt()` to generate random integers. The security number should consist of 6 digits.
8. Another constructor just setting the owner and the balance. Again, security number and the last 10 digits of the IBAN should be set here. The limit remains 0.0.
9. Getter methods for: *balance, iban, owner* and *limit*.
10. `deposit(double amount, int securityNumber)` for depositing money. This should only work if the security number is correct.
11. `withdraw(double amount, int securityNumber)` for withdrawing money. Keep in mind that the user has a limit for withdrawing money. Again, check the security number.
12. `transfer(String remoteIban, double amount, int securityNumber)` for transfering money to another account. Again, keep in mind the security number.

Think about good return values for your methods.

Task 3
Have a look at the `TestSuite.java`. It already contains one test case for testing the initialization of the owner. Add at least **15** test cases. All tests should pass in the end. You should have **at least** one test case for every method. Try to think of edge cases.