University of HUDDERSFIELD

CFS2160: Programming Stream

Tutorial/Practical 8

Using Java Classes

Introduction

This week we will develop the Bank Account class from last week[1].

As usual, activities marked ⇶ should be included in your logbook for assessment. All code *from now on* should be submitted as *listings*, using the layouts previously given.

Purpose

Today we will enhance the `BankAccount` class discussed in the lectrure to protect the values in its instance variables, mainly the balance. We will then test it by developing a second class that creates and uses a single `BankAccount` object.

You saw all the code needed to do all this in the lecture this morning.

If in doubt - Ask!

Activities

1. Make a new folder (package) in IntelliJ and copy the lecture `BankAccount` class from GitHub into it.
   You *will* need to change the package at the top of this class to match your own directory structure. This is important. If you do not do this, IntelliJ will be confused about which `BankAccount` class you are using later on.

2. Refactor the `deposit` method so that it protects the value of the balance (that is, it rejects deposits of zero or less). It should return a Boolean to indicate whether or not the deposit succeeded. Do the same with the `withdraw` method (which should also not allow withdrawals of zero or less) and also `addInterest` (which should not allow the interest rate to be zero or negative, and which should not apply interest to any negative balance).

[1] By the end of the module there will have been an awful lot of bank accounts. Sorry.

3. Refactor the `withdraw` method so that the balance is not allowed to go below zero *unless* the customer has an overdraft[2].

4. ✂
   Include a listing of the new class (including all the enhancements from steps 2 and 3 above) in your logbook. Include brief comments on each method to show what changes you have made.

5. ✂
   In the same package (folder), implement a `BankAccountDemo` class. This class should create a BankAccount object, and should include examples of the methods in the class being used. This new class will just have a single `main` method.
   Include a listing of this program in your logbook, along with its output. Use comments in the program to show what you are testing.

6. (For those who want a challenge.)
   Add a third constructor to the BankAccount class. This should take a single parameter, which is the name of the customer. It should allocate a random account number - there is some code below that will help. The balance should be set to zero, and it should be assumed that the customer does not have an overdraft.

Possibly Useful Code[3]

```
package week08;

import java.util.Random;

public class BankAccountRandom {

    private String accountNumber;

    public BankAccountRandom () {
        this.accountNumber = generateAccountNumber ();
    }

    private String generateAccountNumber ()
    {
        String acc = "";
        Random rg = new Random ();

        for (int i = 0; i < 9; i ++) {
            acc += rg.nextInt (10);
        }

        return acc;
    }
}
```

[2] You may well need one of "or", which is || or "and", which is &&.
[3] Yes, that method is private. It's private because we only want it to be used within this class.