Classes, Constructors, and Inheritance

An Example with Bank Accounts

2/16/2022

Getting Started:

As starting point for this lab an account.js file has been provided containing an `Account` class. Read through the code, making sure you understand what everything does.

An Account simply encapsulates its number and balance. Getters are provided for both number and balance, but because they should not be modified directly no setters are created.

There are two methods that can change the balance, `deposit(amount)` which add money into the account, and `withdraw(amount)` that removes money from the account.

Lastly there is a `toString()` method that creates a string representation of an Account.

Exercises:

a) **Use the Mocha test file, bankTests.js,** to verify that that everything in accounts.js works as expected.

b) Extend the `Account` class by creating a class called `SavingsAccount` in a file called `savingsaccount.js`. In addition to the attributes of `Account`, `SavingsAccount` should have an interest variable, which is set in the constructor and has a getter and a setter mehtod. It should also have an `addInterest()` method which deposits the interest amount into the account. The calculation for the amount is balance * interest / 100 . Be sure to also overwrite the `toString()` method, and test with the Mocha tests in bankTests.js for the methods in `SavingsAccount`.

c) Create a `CheckingAccount` class by extending `Account`. In addition to the attributes of an `Account`, it should have an overdraft limit variable. The overdraft amount indicates how much a person is allowed to temporarily withdraw beyond what they have. In other words, it's the amount that an account is allowed to go into the red (negative balance). Be sure to set this value in the constructor and create a getter and a setter for it. Also make sure that you override the `withdraw(amount)` method and the `toString()` method. Test with bankTests.js.

d) Next create a `Bank` class, a `Bank` object should have an array of `Account` objects, and have `addAccount(), addSavingsAccount(interest), addCheckingAccount(overdraft)` methods each of which returns the number of the created account. Also add a `closeAccount(number)` method that closes (removes from the array) the account with that number, and a `accountReport()` method that returns a String

with each account on its own line. Use a static `nextNumber` variable on the Bank class to know what the number for the next account will be. Test with bankTests.js.

e) Create an `endOfMonth()` method on the Bank class, and on `Account`, `SavingsAccount`, and `CheckingAccount`. The method on the `Bank` class should go through the array calling `endOfMonth()` on each of the accounts collecting their output. For normal `Account`s the `endOfMonth()` method should return an empty string. For `SavingsAccount`s it should call the `addInterest()` method and return a string specifying how much interest was added to this account (see example below), and for `CheckingAccount`s it should check if the balance is below zero, and if so return a string with a warning (see example below). Test with bankTests.js.

Interest added SavingsAccount 2: balance: 102.5 interest: 2.5
Warning, low balance CheckingAccount 3: balance: -100 overdraft limit: 500