**Modified Bank Account Classes**

Ethan Turkeltaub

See as a Gist with syntax highlighting: <https://gist.github.com/1417378>

**Bank.java**

import java.util.\*;

/\*\*

\* A Java representation of a Bank.

\*

\* @author Ethan Turkeltaub

\* @version 0.1.0

\*/

public class Bank {

/\*

\* Instance variables.

\*/

private String name;

private Account[] accounts;

/\*

\* Constructors.

\*/

public Bank() {

name = "Farmington Bank";

Account account = new Account();

Account[] contents = new Account[10];

for(int i = 0; i < 10; i++) {

contents[i] = account;

}

accounts = contents;

}

public Bank(String bankName, Account[] bankAccounts) {

name = bankName;

accounts = bankAccounts;

}

public Bank(int numAccounts, double startingAmount) {

Account[] contents = new Account[numAccounts];

for(int i = 0; i < numAccounts; i++) {

Account account = new Account(startingAmount);

contents[i] = account;

}

accounts = contents;

}

/\*

\* Methods.

\*/

public double totalFunds() {

double totalFunds = 0;

for(Account a : accounts) {

totalFunds = totalFunds + a.getBalance();

}

return totalFunds;

}

public void chargeFees(double fees) {

for(Account a : accounts) {

a.withdraw(fees);

}

System.out.println("Removed " + fees + " dollars from each account.");

}

public void addInterest(double interest) {

for(Account a : accounts) {

a.deposit(a.getBalance() \* (1 + interest));

}

System.out.println("Added " + interest + "% interest.");

}

public Account[] returnAccounts() {

return accounts;

}

public void printCompanyReport() {

System.out.println("======= REPORT FOR: " + name + " ========");

for(Account a : accounts) {

System.out.println("");

System.out.println("Account: " + a.getOwner());

System.out.println(" balance: " + a.getBalance());

System.out.println(" pin: " + a.getPIN());

}

System.out.println("Total funds: " + this.totalFunds());

}

}

**Account.java**

/\*\*

\* A Java representation of a bank account.

\*

\* @author Ethan Turkeltaub

\* @version 0.1.0

\*/

public class Account {

/\*

\* Instance variables.

\*/

private String owner;

private double balance;

private int pin;

private int id;

private static int nextID = 0;

/\*

\* Constructors.

\*/

public Account() {

owner = "Joe Shmoe";

balance = 1000.0;

pin = 1234;

nextID++;

id = nextID;

}

public Account(double inBalance) {

owner = "Joe Shmoe";

balance = inBalance;

pin = 1234;

}

public Account(String inOwner, double inBalance, int inPIN) {

owner = inOwner;

balance = inBalance;

pin = inPIN;

}

/\*

\* Methods

\*/

public void getInfo() {

System.out.println(owner + "'s account has a balance of " + balance + " dollars and a PIN of " + pin);

}

public String getOwner() {

return owner;

}

public double getBalance() {

return balance;

}

public int getPIN() {

return pin;

}

public void deposit(double amount) {

balance = balance + amount;

//System.out.println("Balance is now " + balance + " dollars.");

}

public void withdraw(double amount) {

balance = balance - amount;

//System.out.println("Balance is now " + balance + " dollars.");

}

public void changePIN(int newPIN) {

pin = newPIN;

System.out.println("New PIN is " + pin + " .");

}

public boolean isEqual(Account other) {

return this.getBalance() - other.getBalance() < 0.0000001;

}

}

**BankTest.java**

/\*\*

\* The test class BankTest.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class BankTest extends junit.framework.TestCase

{

/\*\*

\* Default constructor for test class BankTest

\*/

public BankTest() {

}

/\*\*

\* Sets up the test fixture.

\*

\* Called before every test case method.

\*/

protected void setUp() {

}

/\*\*

\* Tears down the test fixture.

\*

\* Called after every test case method.

\*/

protected void tearDown() {

}

public void testAddFiveHundredToBankOne() {

Bank one = new Bank();

Account[] oneAccounts = one.returnAccounts();

Account account = oneAccounts[5];

account.deposit(500.0);

assertEquals(1500.0, account.getBalance());

}

public void testAddFiveHundredToBankTwo() {

Account account = new Account();

Account[] accounts = { account, account, account, account, account };

Bank two = new Bank("Bank of Mexico", accounts);

Account[] twoAccounts = two.returnAccounts();

Account testAccount = twoAccounts[2];

testAccount.deposit(500.0);

assertEquals(1500.0, testAccount.getBalance());

}

public void testAddFiveHundredToBankThree() {

Bank three = new Bank(10, 500.0);

Account[] threeAccounts = three.returnAccounts();

Account account = threeAccounts[2];

account.deposit(500.0);

assertEquals(1000.0, account.getBalance());

}

public void testTotalFunds() {

Account account = new Account();

Account[] accounts = { account, account, account, account, account };

Bank one = new Bank();

Bank two = new Bank("Bank of Mexico", accounts);

Bank three = new Bank(10, 500.0);

assertEquals(10000.0, one.totalFunds());

assertEquals(5000.0, two.totalFunds());

assertEquals(5000.0, three.totalFunds());

}

}