package lab5;

import java.util.HashMap;

import java.util.Map;

import java.util.Random;

public class Bank {

private Random random = new Random();

private Map<String, Integer> rates = new HashMap<String, Integer>();

public Bank() {

rates.put("RUB", 1);

rates.put("USD", 60);

rates.put("EUR", 70);

}

public int rand(int min, int max) {

int randomNum = random.nextInt((max - min) + 1) + min;

return randomNum;

}

public double convert(int amount, String currIn, String currOut) throws Exception {

if (rates.containsKey(currIn) && rates.containsKey(currOut)) {

double rateIn = rates.get(currIn);

double rateOut = rates.get(currOut);

rateIn = (rateIn / 100) \* (rand(-20, 20) + 100);

double resultRate = rateIn / rateOut;

return resultRate \* amount;

} else {

throw new Exception();

}

}

}

package lab5;

import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

public class BankTest {

Bank bank;

@BeforeClass

public static void setUpBeforeClass() throws Exception {

}

@AfterClass

public static void tearDownAfterClass() throws Exception {

}

@Before

public void setUp() throws Exception {

bank = new Bank();

}

@After

public void tearDown() throws Exception {

}

@Test

public void test() {

}

}

package lab5;

public class MoneyPrinter {

public MoneyPrinter() {

}

public void print(String operation, String currency, int amount) {

System.out.println(operation + " (" + amount + " " + currency + ")");

}

}

package lab5;

import java.util.HashMap;

import java.util.Map;

public class Wallet {

private Map<String, Integer> wallet = new HashMap<String, Integer>();

private Bank bank;

private MoneyPrinter printer = new MoneyPrinter();

public Wallet(Bank bank) {

this.bank = bank;

}

public int size() {

return wallet.size();

}

public void addMoney(String curr, int amount) {

wallet.put(curr, this.getMoney(curr) + amount);

printer.print("Пополнение баланса", curr, amount);

}

public void removeMoney(String curr, int amount) throws Exception {

if (wallet.containsKey(curr)) {

int inWallet = this.getMoney(curr);

if (inWallet >= amount) {

printer.print("Снятие", curr, amount);

inWallet = inWallet - amount;

if (inWallet > 0) {

wallet.put(curr, inWallet);

} else {

wallet.remove(curr);

}

} else {

throw new Exception();

}

}

}

public int getMoney(String curr) {

if (wallet.containsKey(curr)) {

printer.print("Запрос баланса", curr, wallet.get(curr));

return wallet.get(curr);

} else {

return 0;

}

}

public double getTotalMoney(String curr) throws Exception {

double totalAmount = 0;

for (Map.Entry<String, Integer> entry: wallet.entrySet()) {

totalAmount += bank.convert(entry.getValue(), entry.getKey(), curr);

}

printer.print("Запрос баланса в валюте", curr, (int)totalAmount);

return totalAmount;

}

public String toString() {

String result = "{";

for (Map.Entry<String, Integer> entry: wallet.entrySet()) {

result += " " + String.valueOf(entry.getValue()) + " " + entry.getKey() + ",";

}

if (!result.equals("{")) {

result = result.substring(0, result.length() - 1) + " ";

}

result += "}";

return result;

}

}

package lab5;

import static org.junit.Assert.\*;

import java.util.HashMap;

import java.util.Map;

import java.util.Random;

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Assert;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

class BankStub extends Bank {

private Map<String, Integer> rates = new HashMap<String, Integer>();

public BankStub() {

rates.put("RUB", 1);

rates.put("USD", 60);

rates.put("EUR", 70);

}

public double convert(int amount, String currIn, String currOut) throws Exception {

if (rates.containsKey(currIn) && rates.containsKey(currOut)) {

double rateIn = rates.get(currIn);

double rateOut = rates.get(currOut);

double resultRate = rateIn / rateOut;

return resultRate \* amount;

} else {

throw new Exception();

}

}

}

public class WalletTest{

private Wallet wallet;

@BeforeClass

public static void setUpBeforeClass() throws Exception {

}

@AfterClass

public static void tearDownAfterClass() throws Exception {

}

@Before

public void setUp() throws Exception {

wallet = new Wallet(new BankStub());

}

@After

public void tearDown() throws Exception {

}

@Test

public void testAddMoney() {

wallet.addMoney("RUB", 300);

assertEquals(wallet.getMoney("RUB"), 300);

wallet.addMoney("RUB", 300);

assertEquals(wallet.getMoney("RUB"), 600);

}

@Test

public void testRemoveMoney() throws Exception {

wallet.addMoney("RUB", 100);

wallet.removeMoney("RUB", 20);

assertEquals(wallet.getMoney("RUB"), 80);

}

@Test(expected=Exception.class)

public void testRemoveMoneyException() throws Exception {

wallet.addMoney("RUB", 20);

wallet.removeMoney("RUB", 100);

}

@Test

public void testSize() throws Exception {

assertEquals(0, wallet.size());

wallet.addMoney("RUB", 100);

assertEquals(1, wallet.size());

wallet.removeMoney("RUB", 100);

assertEquals(0, wallet.size());

}

@Test

public void testToString() {

assertEquals("{}", wallet.toString());

wallet.addMoney("GBP", 500);

wallet.addMoney("RUB", 300);

wallet.addMoney("USD", 100);

assertEquals("{ 500 GBP, 100 USD, 300 RUB }", wallet.toString());

}

@Test

public void testGetTotalMoney() throws Exception {

wallet.addMoney("RUB", 300);

wallet.addMoney("USD", 100);

assertEquals((int)6300, (int)wallet.getTotalMoney("RUB"));

}

@Test

public void test() {

//fail("Not yet implemented"); // TODO

}

}