src/main/java/de/uni_passau/fim/prog1/bank/Account.java

```
package de.uni_passau.fim.prog1.bank;

/**
 * Represents a bank account. Each account has an account holder
 * (= the person the account belongs to), a balance (= the current
 * amount of money on the account) and a counter for the number of
 * transactions (deposits and withdrawals) performed in the current
 * month. At the end of the month, the bank charges a fixed monthly
 * fee for the account plus a fee per transaction performed.
 */
class Account {

    public static final float MONTHLY_COST  = 1.50f;
    public static final float COST_PER_TRANSACTION = 0.1f;

    public String name;
    public float balance;
    public int transactionCount;

    /**
     * Constructs an account from the name of the account holder.
     * The balance and the count for transactions to charge are
     * both set to zero.
     *
     * @param name the account holder's name
     */
    public Account(String name){
        this.name = name;
        balance = 0f;
        transactionCount = 0;
    }

/**
 * Deposits a given amount on this account.
 *
 * @param amount the amount to deposit on the account
 */
public void deposit(float amount){
    balance += amount;
    transactionCount++;
}

/**
 * Withdraws the given amount from this account.
 * When the balance is not sufficient to withdraw the requested amount,
 * the balance does not change but the attempt to withdraw counts as
 * two transactions in the monthly costs for the account.
 *
 * @param amount the amount to withdraw from the account
 * @return {@code true} if the amount could be withdrawn;
 * otherwise {@code false}
 */
public boolean withdraw(float amount){
    if(balance >= amount){
        transactionCount++;
        balance -= amount;
        return true;
    }
    else{
        transactionCount += 2;
        return false;
    }
}

/**
 * Returns the balance of this account.
 *
 * @return the balance of the account
 */
public float getBalance() {
    return balance;
}

/**
 * Charge the account with the costs at the end of a month.
 * The fixed monthly costs and the per transaction charge
 * for each transaction recorded are withdrawn from the account.
 * The counter for transactions to charge is reset to zero.
 */
//Attention! Balance can reach into negative value after method
public void monthlyStatement(){
    balance -= (MONTHLY_COST + (transactionCount * COST_PER_TRANSACTION));
    transactionCount = 0;
}

/**
 * Returns a string representation of the account. The returned
 * string contains the name of the account holder, the current
 * balance and the number of transactions.
 *
 * @return a {@link String} describing the account
 */
@Override
public String toString() {
    String str = "";
    str = name + " hat gerade " + balance + " Euro auf dem Konto und hat in diesem Monat " + transactionCount + " Transaktionen durchgeführt.";
    return str;
}
}
```

src/main/java/de/uni_passau/fim/prog1/bank/Transfer.java

```
package de.uni_passau.fim.prog1.bank;

/**
 * Represents a money transfer between two accounts.
 */
class Transfer {

    public Account sender;
    public Account receiver;
    public float amount;

    /**
     * Construct a bank transfer. The sending account, the receiving account
     * and the amount to transfer need to be given.
     *
     * @param sender the account which sends the money
     * @param receiver the account which receives the money
     * @param amount how much money to transfer
     */
    public Transfer(Account sender, Account receiver, float amount){
        this.sender = sender;
        this.receiver = receiver;
        this.amount = amount;
    }

    /**
     * Executes the transfer. The money is only
     * deposited on the receiving account if the withdrawal
     * from the sending account has been successful.
     */
    public void execute(){
        if(amount > 0) {    //Schützt vor Angriffen, wo ein negativer Betrag gesendet wird, der Sender also praktisch Geld vom receiver bekommt
            if (sender.getBalance() >= amount) {
                sender.withdraw(amount);
                receiver.deposit(amount);
            }
            else {
                System.out.println("Der Sender hat zu wenig Geld");
            }
        }
        else{
            System.out.println("Sie können keinen negativen Betrag senden!");
        }
    }
}
```

src/main/java/de/uni_passau/fim/prog1/bank/Bank.java

```
package de.uni_passau.fim.prog1.bank;

class Bank {
    public static void main(String[] args) {

        Account a = new Account("Anna Consultant");
        Account b = new Account("Betty Programmer");
        Account k = new Account("Karl Knappbeikasse");

        a.deposit(1000);
        b.deposit(500);
        k.deposit(20);

        a.withdraw(50);  // should succeed
        k.withdraw(100); // should fail

        // Show information about accounts
        System.out.println(a);
        System.out.println(b);
        System.out.println(k);

        Transfer t1 = new Transfer(k, a, 200);
        t1.execute();   // should not transfer any money
        System.out.println(a);
        System.out.println(k);

        Transfer t2 = new Transfer(b, k, 200);
        t2.execute();   // should transfer money
        System.out.println(b);
        System.out.println(k);

        a.monthlyStatement();
        b.monthlyStatement();
        k.monthlyStatement();

        System.out.println(a);
        System.out.println(b);
        System.out.println(k);

    }
}
```