

# Multiplicity

In Domain Model, Class Diagram and Code

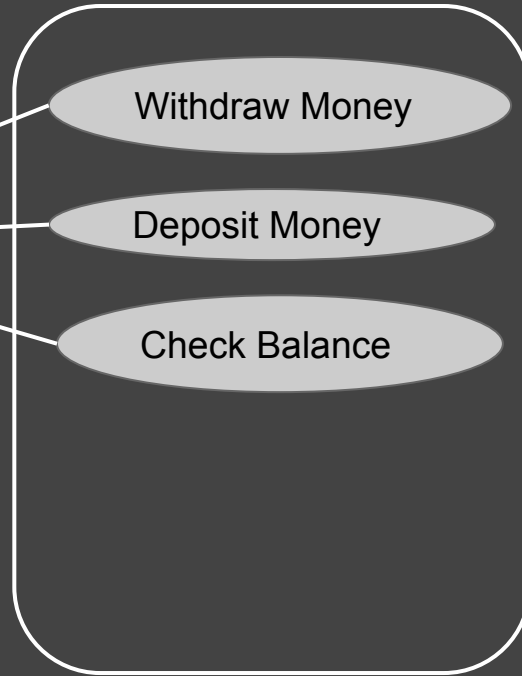
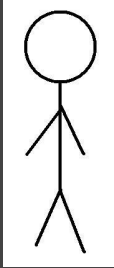
# Bank Account

A customer opens a bank account in a bank.

After opening the account the customer can withdraw, deposit and check his balance at any time he wants. The bank charges a fee of 10kr. for every withdraw a customer makes. The Customer can not withdraw more money than is on the account including the fee. The bank also has an interest rate of 8.5%, which is used for calculating a certain amount of interest on the amount deposited by the customer.

# Use Case Diagram

Primary Actor  
Customer



# Use Cases

## UC#1: Withdraw Money

### Actor: Customer

Customer requests to withdraw money from his account and specifies the amount.

The bank puts on a fee of 10 kr. when a withdrawal is made, so the customer gets the requested amount, and the fee is subducted on the remaining balance of the account.

# Use Cases

## UC#2: Deposit Money

### Actor: Customer

The Customer requests to deposit money. The bank calculates the interest based on the interest rate and the amount just deposited, the interest and the amount is summed. The sum is then added to the account balance.

# Use Cases

## UC#3: Check Balance

### Actor: Customer

Customer requests to see the balance of his account.

The account number and name of the customer together with the balance is displayed.

# Nouns

Customer, Money,  
Account, Amount,  
Bank, Fee, Balance,  
Error message,  
interest, interest rate,  
Account Number,  
Customer Name,

Customer  
    name  
Account  
    balance  
    accountNumber  
Bank  
    fee  
    interest rate

# Nouns

Customer, ~~Money~~,  
Account, ~~Amount~~,  
Bank, Fee, Balance,  
~~Error message~~,  
~~interest~~, interest rate,  
Account Number,  
Customer Name,

Customer  
    name  
Account  
    balance  
    accountNumber  
Bank  
    fee  
    interest rate



# Domain Model

## Costumer

name

## Account

Account Number  
Balance

## Bank

Fee  
Interest Rate

# Verbs = Actions = Methods

Withdraw

get fee

Deposit

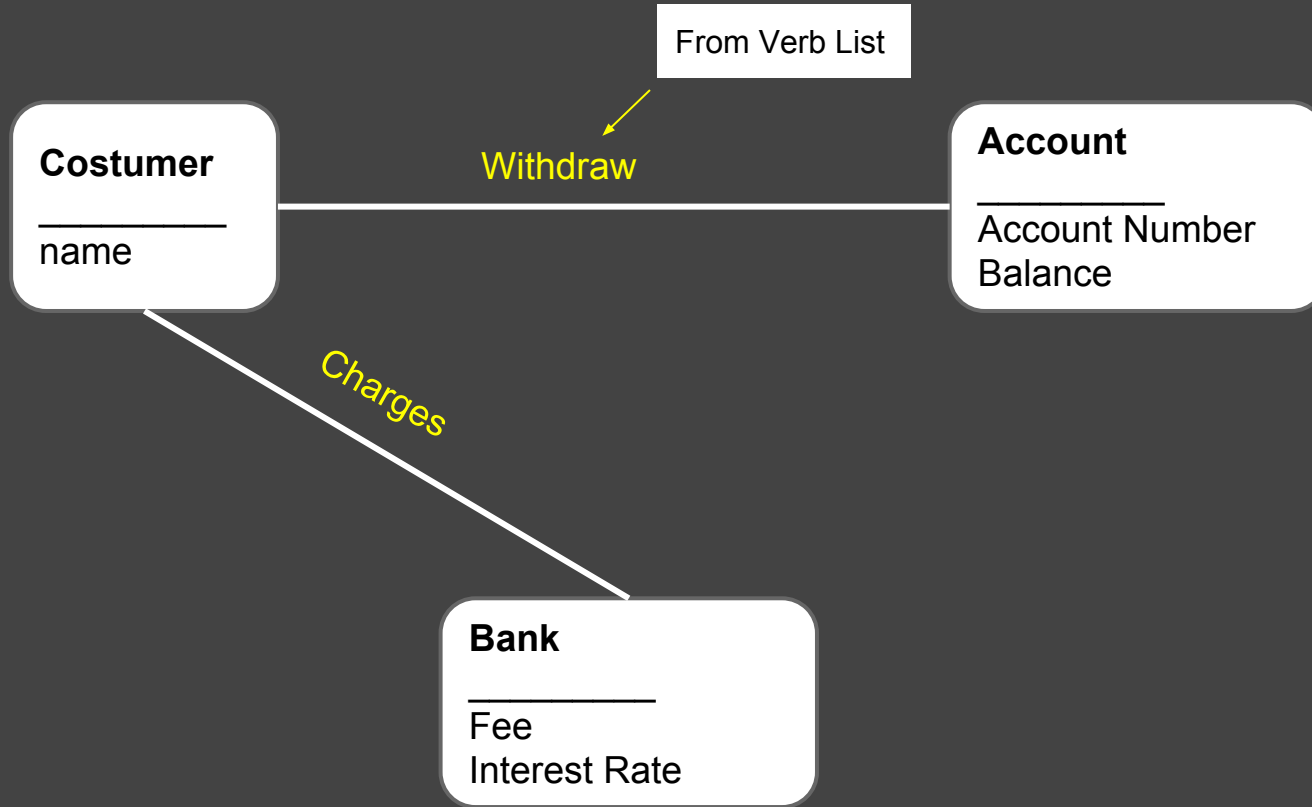
Calculate interest

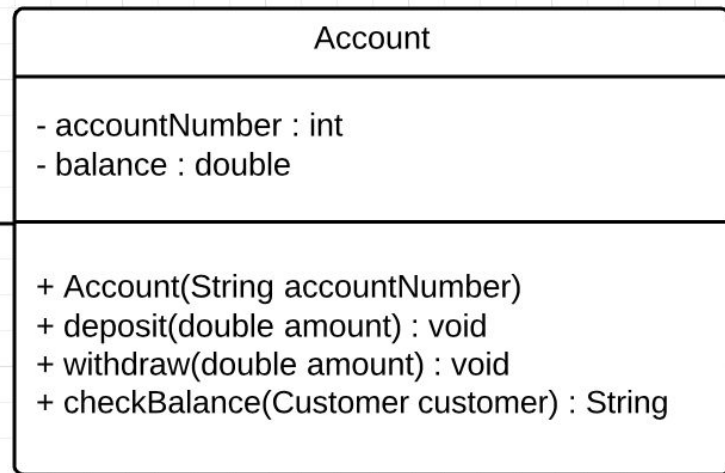
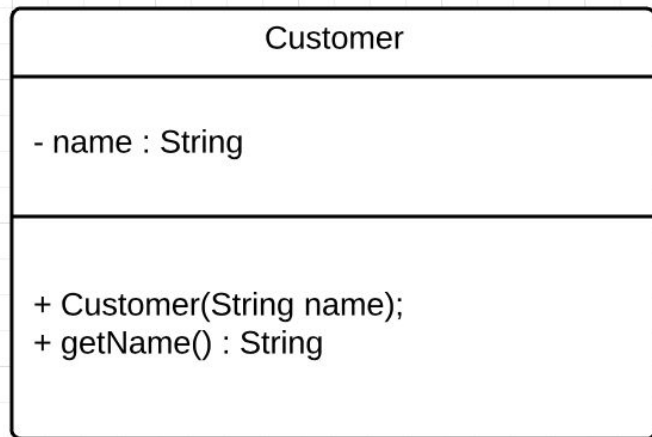
Add to balance

Check Balance

Charges

# Domain Model

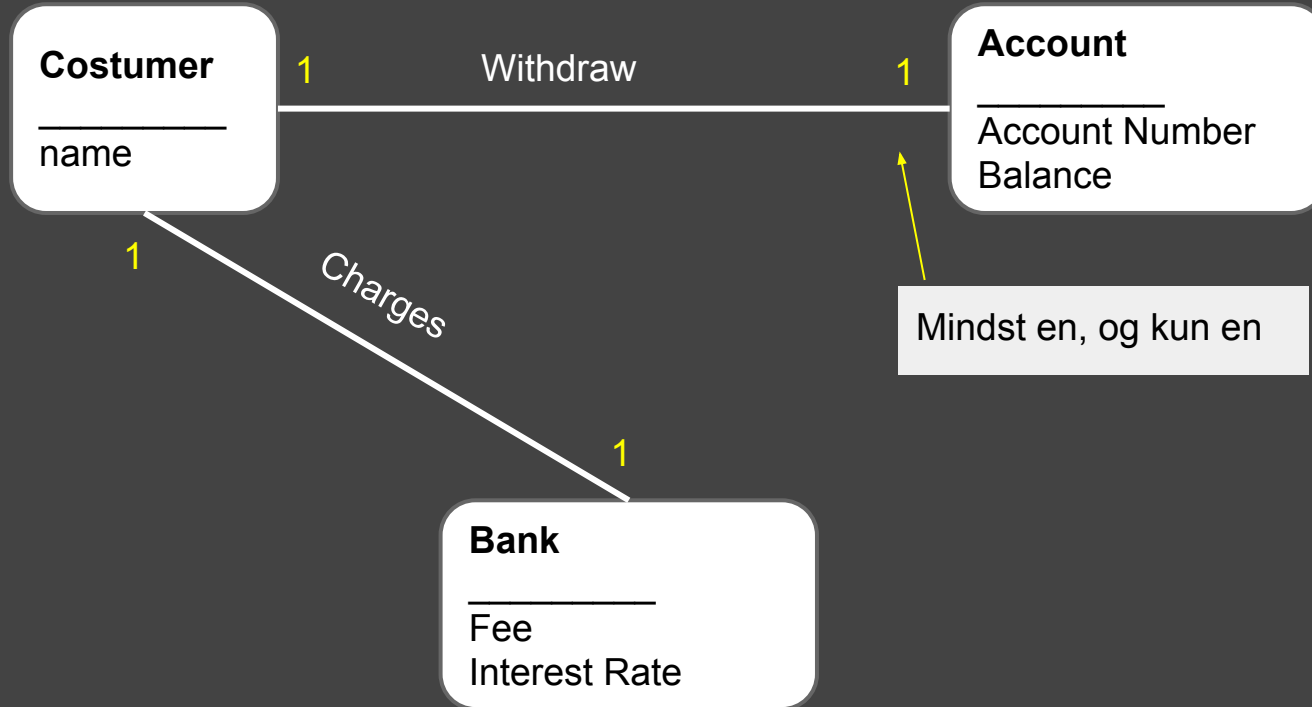




Kode

# Multiplicity

# Domain Model

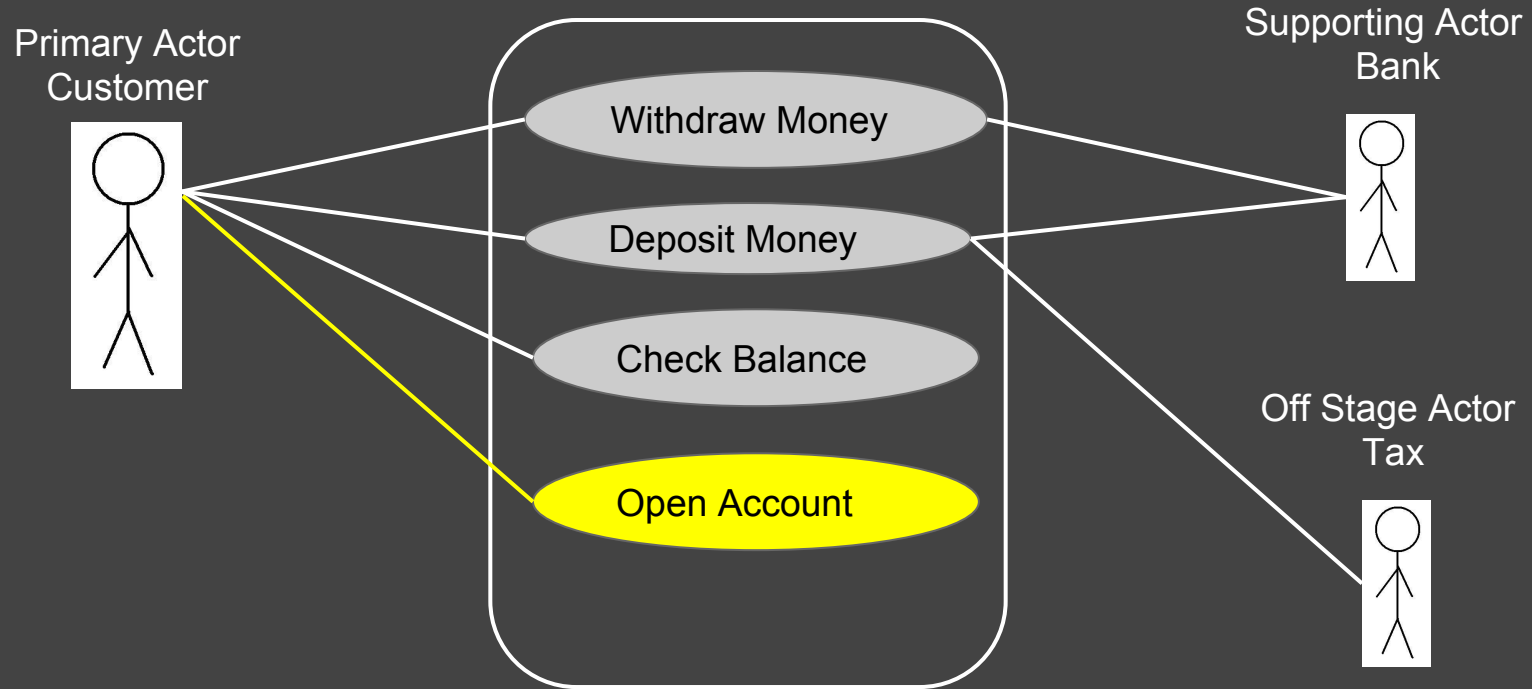


# Klassediagram 1-1 forbindelse



# Kode 1-1 forbindelse

# Use Case Diagram



# Use Cases

## UC#3: Open Account

Actor: Customer

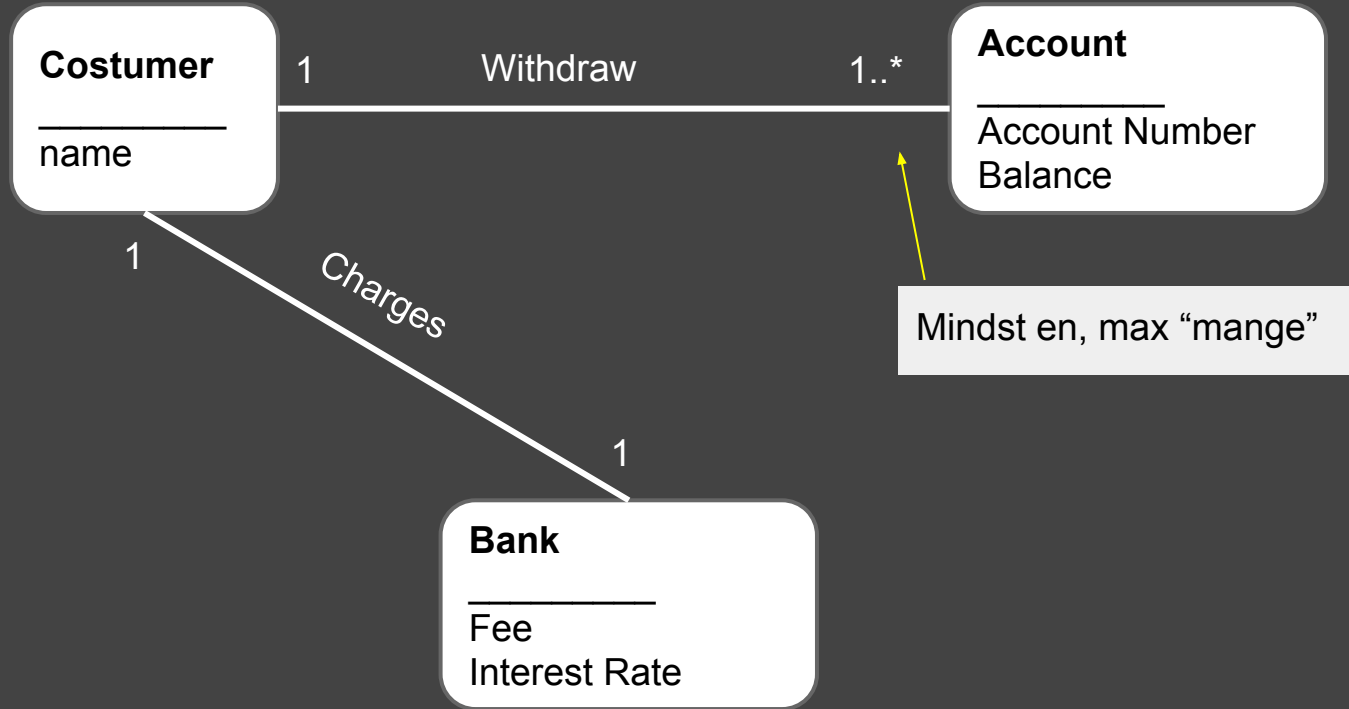
### Main Success Scenario:

Customer enters the bank and requests to open an account. The Bank deals with this request ... (according to the bank regulations and rules). Customer now has an account and can [Withdraw](#), [Deposit](#) and [Check](#) balance of his account.

### Alternative Flow:

1. If customer already have an account the banks rules and regulations differ. (description of that). The customer now have 2 accounts, and pays 100kr a year for having this extra account.
2. If customer earlier have applied for an account and was rejected the request. The bank looks at the reason and checks if the situation changed. If not the customer is again rejected. If changes has occurred goto Main Success Scenario.

# Domain Model - En til mange



# Klassediagram 1-\* forbindelse

**Kode 1-\* forbindelse**

# Øvelse: Car Wash

1. Lav en Domænemodel for jeres Car Wash projekt
2. Tilføj navngivne relationer og Multiplicity
3. Lav et klassediagram med fokus på de klasser der har en 1 - \* relation
4. Implementer det i kode