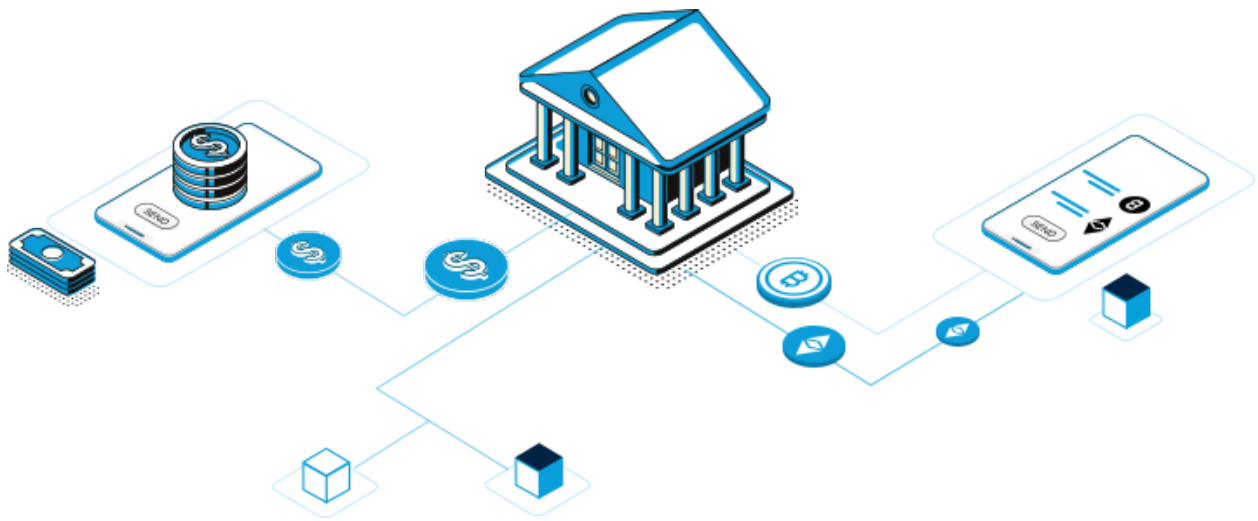


Bank of MR.S



Inledning

Projektuppgiften går ut på att ta fram ett enkelt system för en fiktiv bank.

Banken har ett antal kunder och varje kund kan ha ett eller flera olika Bankkonton.

Programmet kommer bestå av 3 olika klasser som ska definieras: Bank, Account och Customer.

Vi ska med hjälp utav dem tre olika klasserna skriva ut en lista med bankens kunder (personnummer, för och efternamn) till en textfil. Kunna lägga till nya kunder med unika personnummer. Gör det möjligt att enbart ändra namn dock så ska personnumret vara detsamma. Ett alternativ att kunna avsluta en kunds konton=kontot tas bort samt info. Generera ett nytt Konto till en befintlig kund med ett unikt kontonummer (1001, 1002, osv). Möjlighet att avsluta konton och därefter saldo skrivs ut samt kontot tas bort.

För en specifik kund ska du kunna få fram följande:

information om vald kund inklusive alla konton (kontonummer, saldo, kontotyp)

Kunna sätta in pengar på ett eller flera konton

Ta ut pengar från kontot (men bara om saldot täcker uttagsbeloppet)

Teknologierna som ska användas är Java och Git.

Klasserna som ska implementeras är följande: Bank, Customer och Account. Dessa ska utföra olika uppgifter dessa är..

Börja med att implementera klassen Account som ska hantera följande information:

- Saldo
- Kontotyp (konto (Account))
- Kontonummer (det kan inte finnas flera konton med samma kontonummer).

Man ska kunna utföra transaktioner (insättning/uttag), hämta kontonummer, samt presentera kontot (kontonummer, saldo, kontotyp). Implementera metoder som säkerställer ovanstående krav i klassen Bank nedan (Bank inkluderar förslag på metoder. Komplettera dessa med fler metoder om det behövs).

Klassen Customer ska hantera följande information:

- Kundens namn
- Personnummer
- En lista med kundens alla konton

Man ska till exempel kunna ändra kundens namn samt hämta information om kunden (personnummer, för- och efternamn samt hämta information om kundens konton (kontonummer, saldo, kontotyp)). Dessutom ska man kunna hantera kundens konton.

Implementera metoder som säkerställer ovanstående krav i klassen Bank nedan (Bank inkluderar förslag på metoder. Komplettera dessa med fler metoder om det behövs).

Klassen Bank ska innehålla en lista med alla inlagda kunder. Klassen ska innehålla ett antal publika metoder som hanterar kunder och dess konton (se ovan). Du kommer troligtvis att skapa ett antal hjälpmetoder, privata metoder, men de publika metoderna som ska finnas i Bank är följande:

```
public List<String> GetCustomers()
```

- Returnerar en List<String> som innehåller en presentation av bankens alla kunder (personnummer och namn)

```
public boolean AddCustomer(String name, long pNr)
```

- Skapar upp en ny kund med namnet name samt personnummer pNr, kunden skapas endast om det inte finns någon kund med personnummer pNr. Returnerar true om kund skapades annars returneras false.

```
public List<String> GetCustomer(long pNr)
```

- Returnerar en List<String> som innehåller informationen om kunden inklusive dennes konton.

Första platsen i listan är förslagsvis reserverad för kundens namn och personnummer sedan följer

informationen om kundens konton.

```
public boolean ChangeCustomerName(String name, long pNr)
```

- Byter namn på kund med personnummer pNr till name, returnerar true om namnet ändrades annars

returnerar false (om kunden inte fanns).

```
public List<String> RemoveCustomer(long pNr)
```

- Tar bort kund med personnummer pNr ur banken, alla kundens eventuella konton tas också bort

och resultatet returneras. Listan som returneras ska innehålla information om alla konton som togs

bort, saldot som kunden får tillbaka samt vad räntan blev.

```
public int AddAccount(long pNr)
```

- Skapar ett konto till kund med personnummer pNr, returnerar kontonumret som det skapade kontot

fick alternativt returneras -1 om inget konto skapades.

```
public String GetAccount(long pNr, int accountId)
```

- Returnerar en String som innehåller presentation av kontot med kontonummer accountId som tillhör

kunden pNr (kontonummer, saldo, kontotyp, räntesats).

```
public boolean Deposit(long pNr, int accountId, int amount)
```

- Gör en insättning på konto med kontonummer accountId som tillhör kunden pNr, returnerar true om

det gick bra annars false.

```
public boolean Withdraw(long pNr, int accountId, int amount)
```

- Gör ett uttag på konto med kontonummer accountId som tillhör kunden pNr,

returnerar true om det gick bra annars false

```
public String CloseAccount(long pNr, int accountId)
```

- Stänger ett konto med kontonummer accountId som tillhör kunden pNr, presentation av kontots

saldo ska genereras.

Milestones

Mina mål med detta program är att kunna lära sig med hjälp av java att bygga en fiktiv bankapp. Följande mål som jag vill uppnå är:

1. Implementera metoder
2. String ToString metoden
3. Skapa en kunden med följande namn, efternamn, ålder, account number
4. Kunna ta ut och sätt in värde i följande valörer.
5. Skapa en fungerande applikation.

Tasks

Day 1 and 2

Skapa klasser samt struktur i koden. Lägg till kommentarer där du vill lägga till koden för att lättare kunna hålla koll på vad du gör samt ska göra.

Day 3-4 skapa metoder för get/set account number, get/set name, get/set customer och add Customer samt att kunna skriva ut pNr.

Day 5-6

Lägga till deposit and withdrawal, ändra namn på kund samtidigt behålla samma pNr, kunna tabor account och lägga till funktion som visar true or false.

Day 7

Idag blir det final touches på koden och så att man kan testa programmet. Med hjälp av en test menu på main så kan vi få ut det i konsolen som en lista.

Day 8 Redovisning för Rafael.