1. **Projekt Bankverwaltung**
   1. **Konto Verwaltung Basis**

### Es soll eine Kontoverarbeitung für Girokonten, Sparkonten und Darlehenskonten aufgebaut werden.

**Außerdem sollen für die Konten Buchungen durchgeführt werden können.**

Zunächst sollen die Daten für die Konten aus der Datei Bankdaten.txt eingelesen werden.

NeuesGiroKonto;1;Hugo Scheckerich;1000;10.0  
NeuesSparkonto;2;Hugo Scheckerich;0;01.01.2017;5.0  
NeuesDarlehenskonto;3;Hugo Scheckerich;300  
buchung;1;-100  
buchung;1;-1000  
buchung;2;-100  
buchung;2;100  
buchung;3;100  
buchung;3;300

Die Ausgabe soll zunächst auf der Konsole erfolgen:

Buchung in Höhe -1000.0 für das Konto 1 ( GiroKonto ) konnte nicht durchgeführt  
werden !  
----> Dispo limit ist überschritten

Buchung in Höhe -100.0 für das Konto 2 ( SparKonto ) konnte nicht durchgeführt werden !  
----> Konto ist nicht gedeckt

Buchung in Höhe 100.0 für das Konto 3 ( DarlehensKonto ) konnte nicht durchgeführt werden !  
----> Buchungsbetrag muss der Rate entsprechen

Kontonr : 1  
Art : GiroKonto  
Inhaber : Hugo Scheckerich  
Saldo : -100.00  
-----Zugehörige Buchungen----------------------

01.02.2023 -100.00

Kontonr : 2  
Art : SparKonto  
Inhaber : Hugo Scheckerich  
Saldo : 100.00  
-----Zugehörige Buchungen----------------------

01.02.2023 100.00

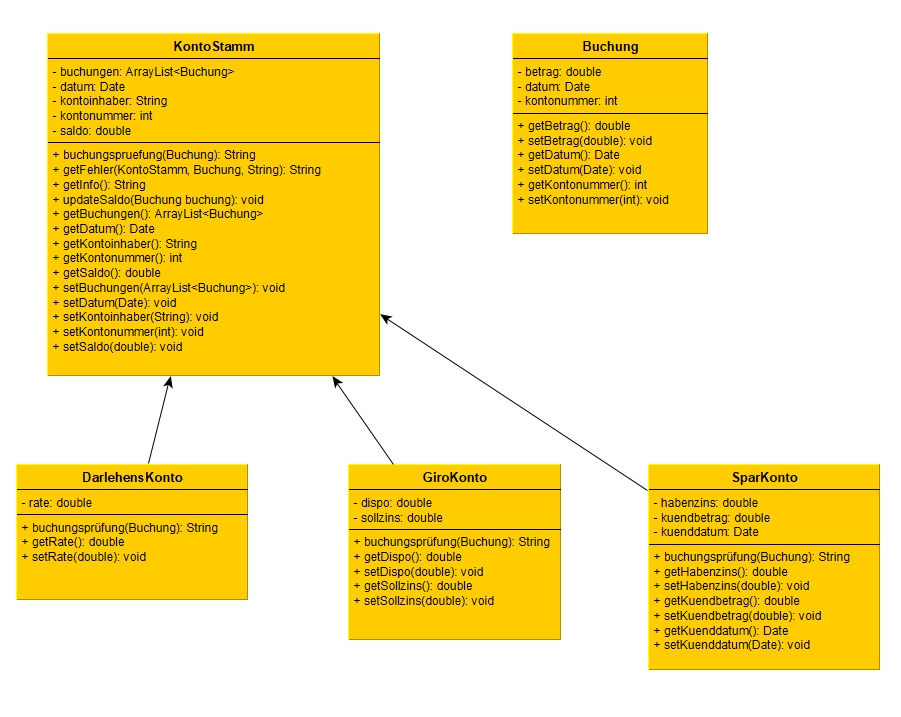
Kontonr : 3  
Art : DarlehensKonto  
Inhaber : Hugo Scheckerich  
Saldo : 300.00  
-----Zugehörige Buchungen----------------------  
  
01.02.2023 300.00

Im weiteren Verlauf des Kurses soll die Eingabe aus der Datenbank erfolgen, die Ausgabe soll über eine JavaFX GUI erfolgen.

### Gehen Sie zur Lösung der Aufgabe wie folgt vor :

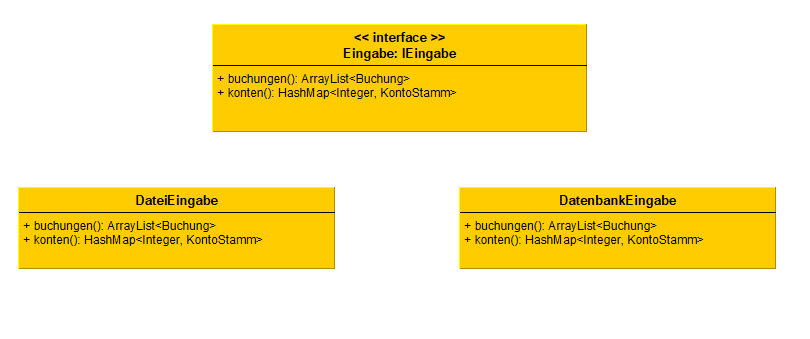
1. Erstellen Sie die Klassen , Attribute und Getter/Setter Methoden nach dem beigefügten Klassendiagramm in einem Package bank.klassen. Erstellen Sie zuächst nur die Klassen, Attribute und die Getter /Setter Methoden. Die anderen Methoden werden in den nachfolgenden Punkten erstellt.

Klassendiagramm für die Bankklassen



1. Die Eingabe der Daten soll zunächst über die Datei Bankdaten.txt erfolgen. Da zu einem späteren Zeitpunkt die Eingabe über eine Datenbank erfolgen soll, soll eine Schnitstelle erstellt werden, die für beide Funktionen Gültigkeit hat. Erstellen Sie bitte die Schnittstelle IEingabe und die Klasse DateiEingabe nach folgendem Klassendiagramm.

Klassendiagramm für die Eingabe



Zunächst erstellen Sie bitte in der Klasse DateiEingabe die Methode konten.

Hier sollen die Klassen Girokonto, Sparkonto und Darlehenskonto abhängig von dem ersten Feld in der Datei *Bankdaten.txt* (Beispiel NeuesGirokonto..) erstellt werden

### Hinweis :

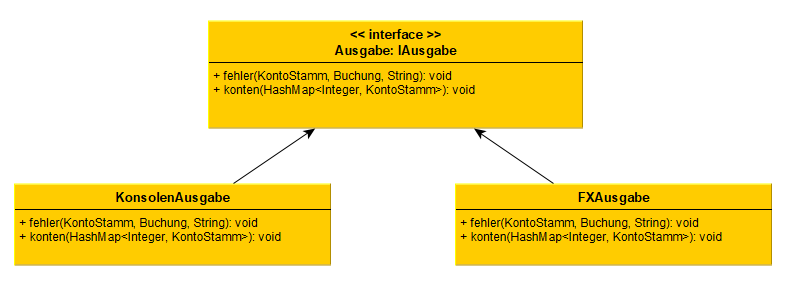
Die Eingabe sind jeweils in einer Zeile getrennt durch ; entahlten. Der erste Eintrag enthält die Verarbeitungsart. :

Bei den Konten kommt danach Kontonummer, Kontoinhaber anschließend die für die jeweiligen Konten speziellen Angaben.

Die erstellten Klassen sollen in der HashMap (Key ist die Kontonummer) gespeichert werden. Diese soll zurückgeben werden.

1. Erstellen Sie eine ausführbare Klasse AppBankDatei , die die Klasse DateiEingabe über die Schnittstelle instantiert und die Methode konten aufruft. Die erstellte Hashmap soll in der Klasse AppBankDatei gespeichert werden. Geben Sie ggf. zum Testen einige Daten auf der Konsole aus.
2. Die Ausgabe soll ähnlich wie die Eingabe verlaufen, hier soll zunächst die Ausgabe über die Konsole erfolgen, im Laufe des Kurses soll die Ausgabe auch über eine GUI erfolgen können. Daher erstellen Sie bitte die Schnittstelle *IAusgabe* und die Klasse *Konsolausgabe* nach folgendem Klassendiagramm.

Klassendiagramm für die Ausgabe



Kodieren Sie in der Klasse *KontoStamm* die Methode *getInfo* (siehe Ausgabe) und rufen Sie diese in der Methode *konten* der Klasse *Konsolausgabe* für jedes Konto auf und geben Sie den String, der in der Methode getInfo erzeugt wird auf der Konsole aus.

1. Lesen Sie die Buchungen aus der Datei *Bankdaten.txt* ein. Verwenden Sie hierzu die Methode *buchungen* der Klasse *DateiEingabe* ( siehe auch Seite 5). In der Klasse *AppBankDatei* rufen Sie die Methode auf und speichern die zurüggegebene ArrayList.
2. Erstellen Sie bitte in den Klasse *KontoStamm* die Methode *buchungspruefung* . Diese Methode soll in den abgeleiteten Klassen überschrieben werden.

Hierbei beachten Sie bitte folgende Vorgaben :

Bei einem Girokonto darf der Saldo nach der Buchung nicht kleiner als das Dispolimit sein.

Bei einem Sparkonto darf der Saldo nach der Buchung nicht negativ sein.

Bei einem Darlehenskonto darf der Buchungsbetrag nur der Rate entsprechen.

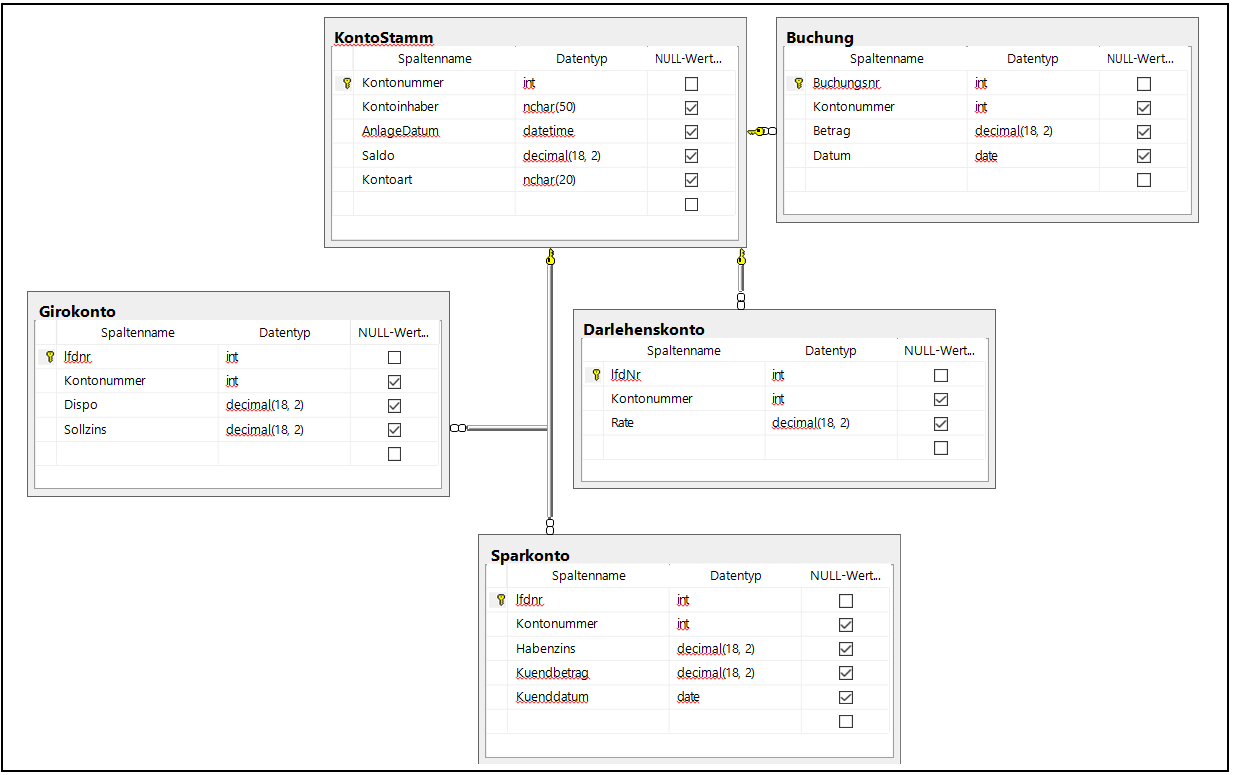
1. In der Klasse *AppBankDate*i soll pro Buchung das entsprechende Konto (aus der HashMap festgestellt werden und die Prüfung aufgerufen werden. Falls die Prüfung negativ ist, soll eine entsprechende Fehlermeldung über die Methode *fehlermeldung* der Klasse *Konsolausgabe* erstellt werden, anderenfalls soll die Methode *updateSaldo* der Klasse *KontoStamm* aufgerufen werden. Diese Methode soll die Buchung in der ArrayList *buchungen* der Klasse *KontoStamm* speichern und den Saldo updaten.
2. Am Schluss sollen die Methoede *getInfo* der Klasse *KontoStamm* erweitert werden, so dass die Buchungen ebenfalls auf der Konsole ausgeben werden.
   1. **Datenbanken(JDBC)**

Die Eingabe der Daten soll jetzt über die Datenbank erfolgen. Die Ausgabe der Daten soll wie bisher auf der Konsole erfolgen.

Führen Sie bitte folgende Schritte durch:

* + 1. Bitte legen Sie in SQL Server die Datenbank Bankverwaltung an.

Folgendes ERM ist zu erstellen.



* + 1. Erstellen Sie bitte eine Applikation, die die Daten aus der Eingabedatei *Bankdaten.txt* in die Datenbank schreibt. Verwenden Sie bitte hierzu für das Lesen der Daten bereits bestehende Klasse *DateiEingabe.*

### Hinweis:

Beachten Sie, dass für die einzelnen Kontoarten auch ein Datensatz in der Tabelle Kontostamm angelegt werden muss.  
  
Als Datum für die Buchungen verwenden Sie bitte das Tagesdatum

* + 1. Erstellen Sie bitte in dem Package *bank.eingabe* eine neue Klasse *DatenbankEingab*e, die die Schnittstelle *IEingabe* implementiert.

Lesen Sie die einzelnen Konten (GiroKonto, Sparkonto und Darlehenskonto) in der Methode konten ein und erstellen Sie die einzelnen Klassen, die in der HashMap *konten* gespeichert werden sollen.

In der Methode *buchungen* lesen Sie bitte alle Buchungen ein und speichern Sie in der ArrayList *buchung*