

Cash-Flow-Reminder

Problemdefinition

1 Problembeschreibung:

Viele Studenten verfügen über ein eher geringes monatliches Budget, mit welchem sie aber alle Lebensunterhaltskosten decken sollten. Deshalb ist oft am Ende des Geldes noch sehr viel Monat übrig. Außerdem wundert man sich oft, wo das Geld geblieben ist, da man gefühlsmäßig gar nicht all zu viel ausgegeben hat...

2 Projektziel:

Dieses Programm soll jederzeit das restliche monatliche Guthaben anzeigen. Außerdem werden die Verwendungszwecke, Zeitpunkte bzw. Beträge der getätigten Ausgaben regelmäßig eingegeben, welche dann ebenfalls jederzeit abrufbar sind.

Dabei können Kategorien gebildet werden, wie zB:

- Miete
- Handyrechnung
- Ausgaben fürs Studium (Bücher, Skripten etc.)
- Lebensmittel (Einkäufe bei Spar, Hofer etc.)
- Hygieneartikel (Einkäufe bei DM, Bipa etc.)
- Fortgehen (...)
- Ausflüge/Reisen
- Etc.

Am Ende des Monats kann eine Liste aller Ausgaben erstellt werden, bzw. werden die Summen aller Kategorien berechnet.

Somit weiß man am Ende, wo das Geld geblieben ist bzw. wo in Zukunft Einsparungen gemacht werden sollten.

3 Aufgaben des Benutzers

Beim Start des Programmes wird dem Benutzer ein Menü angezeigt, in welchem er eine Auswahl an Befehlen hat.

Barbara Dellanoi 0250678

Zu Beginn sollte er ein neues Konto erstellen, und dort sein monatliches Guthaben eingeben. Anschließend sollte er die bereits getätigten Ausgaben eingeben. Außerdem kann er seinen Kontostand abrufen sowie die bereits verbuchten Ausgaben.

Analyse und Modellierung

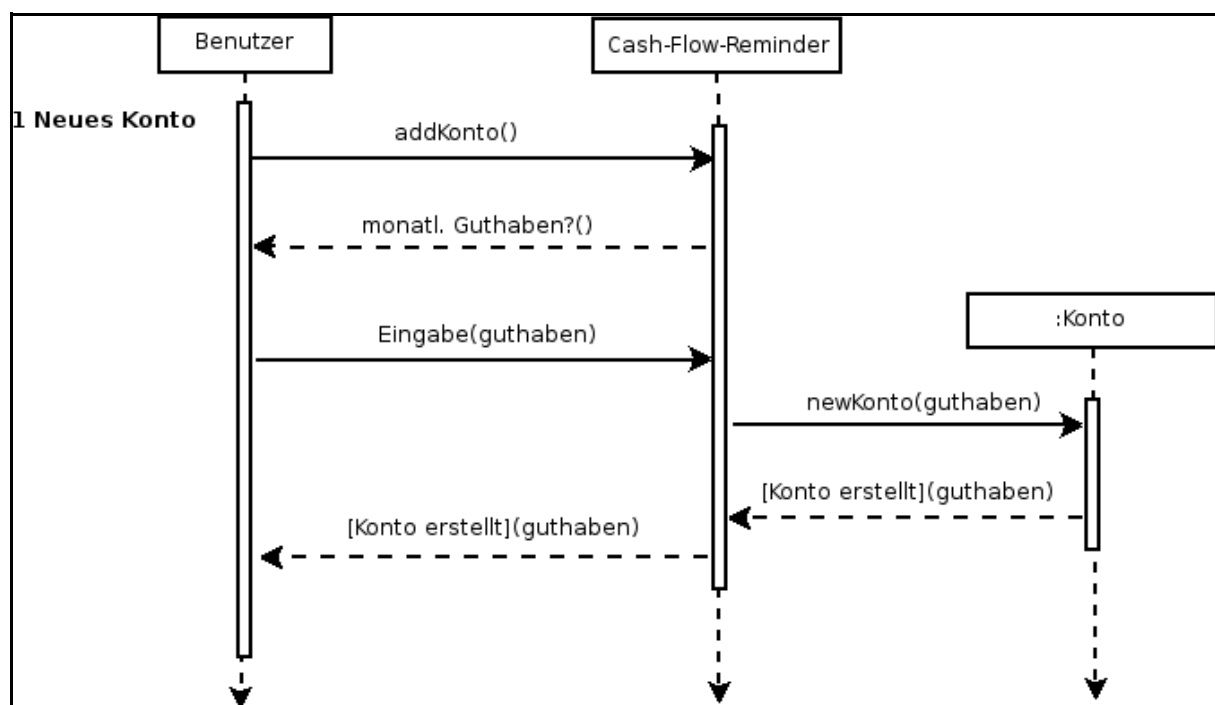
Beim Programmstart wird dem Benutzer folgendes Menü agezeigt:

Bitte einen Befehl durch die entsprechende Zahl auswaehlen, und mit return bestaetigen:

1. Neues Konto erstellen
2. Naechster Monat
3. Fixe Abbuchung definieren
4. Variable Abbuchung eingeben
5. Kontostand abrufen
6. Alle bisherigen Abbuchungen ausgeben
7. Summe Ausgaben der Kategorien
8. Menue anzeigen
9. Programm beenden

1 Beschreibung Befehl 1: Neues Konto erstellen

Sequenzdiagramm:



Bsp: Benutzer möchte ein Konto mit 700 Euro erstellen:

Eingabe Benutzer: **1**

Ausgabe C.F.R.: *Monatliches Guthaben eingeben:*

Eingabe Benutzer: **700**

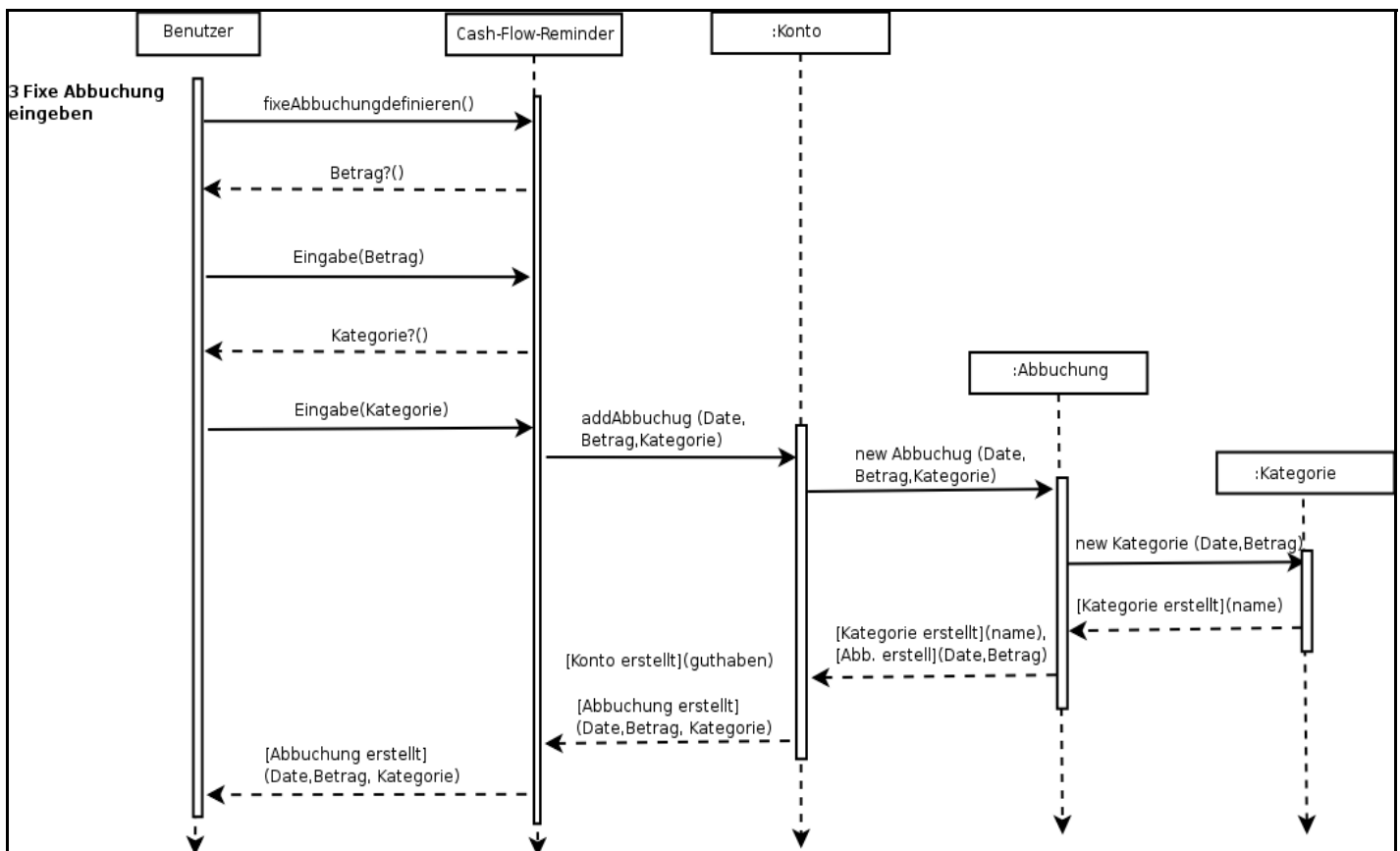
Ausgabe C.F.R.: *Fuer diesen Monat stehen Dir 700.0 Euro zur
Verfuegung. Aber gib nicht alles auf einmal aus!!!*

2 Beschreibung Befehl 2: Nächster Monat

Da es sich bei dem Programm um einen Prototypen handelt, kann dieser Befehl noch nicht ausgeführt werden. Geplant ist, dass beim Abruf dieses Befehls das Guthaben vom vorherigen Monat angezeigt wird, bzw. die fixen Abbuchungen automatisch abgezogen werden.

3 Beschreibung Befehl 3: Fixe Abbuchung definieren

Sequenzdiagramm:

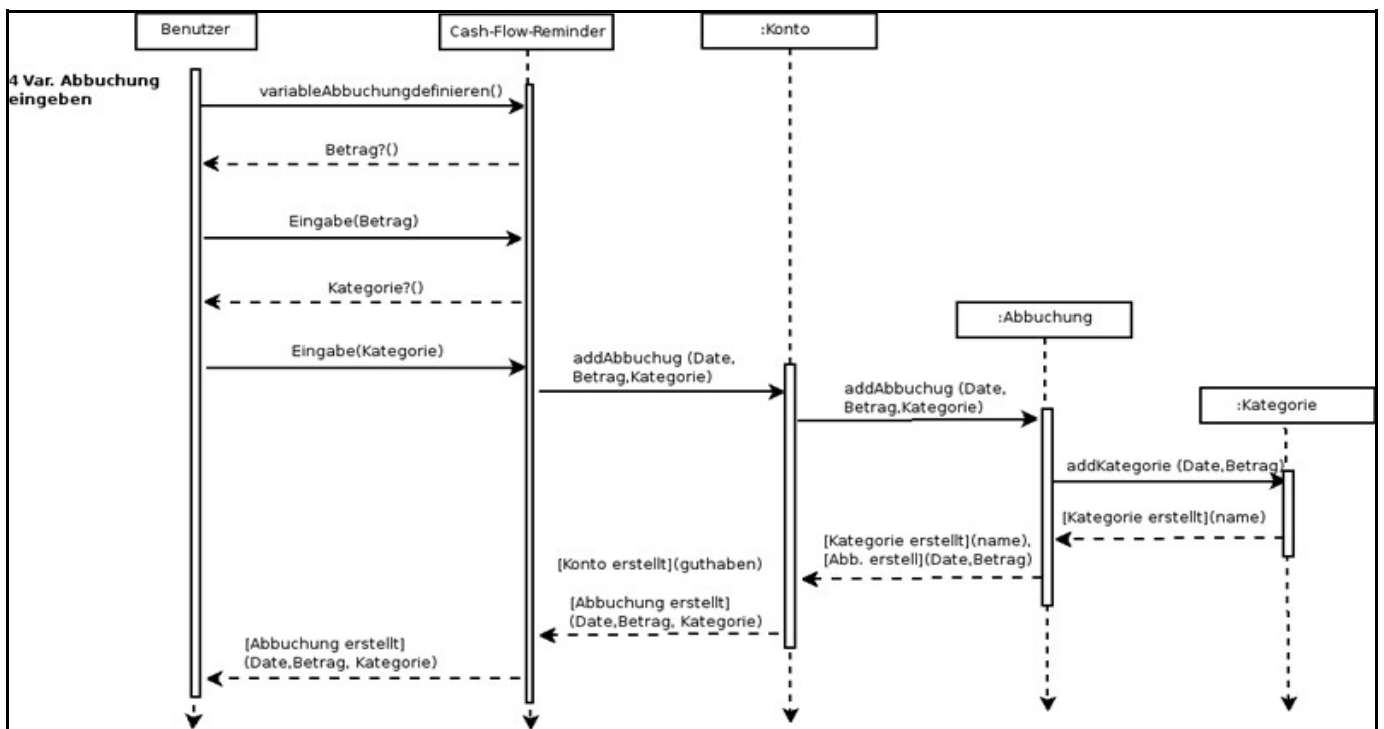


Bsp: Benutzer möchte eine fixe Abbuchung von 300,- für „Miete“ definieren.:

Eingabe Benutzer: **3**
Ausgabe C.F.R: *Wievil fällt fix weg?*
Eingabe Benutzer: **300**
Ausgabe C.F.R: *Und fuer was?!?!?*
Eingabe Benutzer: **Miete**
Ausgabe C.F.R: *300.0 Euro wurden Dir fuer Miete als Fixkosten abgezogen.*

4 Beschreibung Befehl 4: Variable Abbuchung definieren

Sequenzdiagramm:



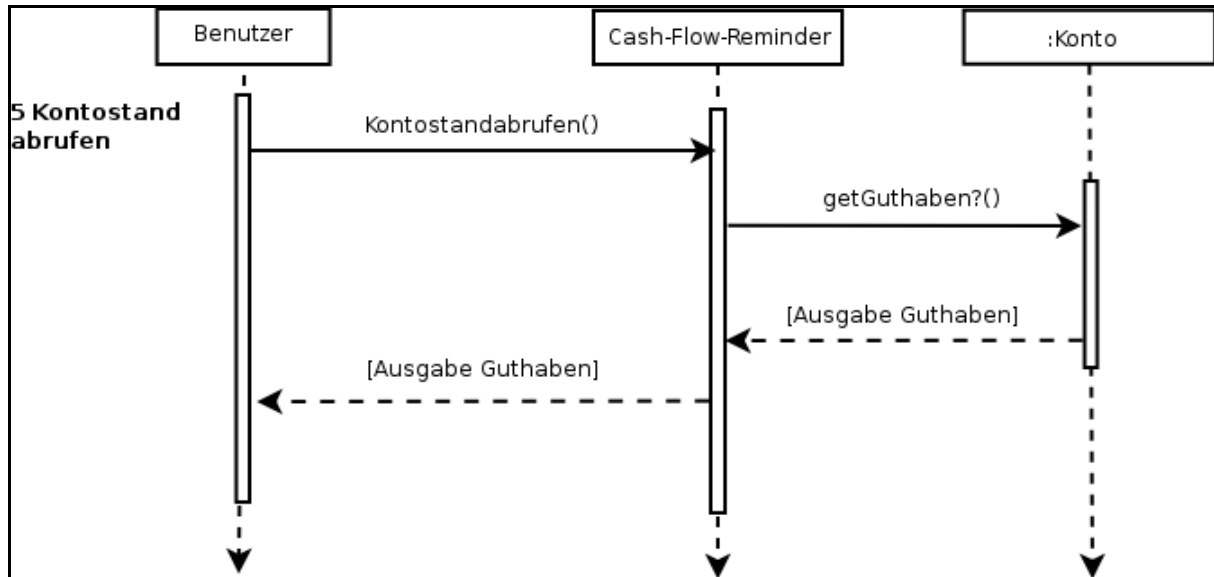
Bsp: Benutzer möchte eine variable Abbuchung von 25,- für „Lebensmittel“ definieren.:

Eingabe Benutzer: **4**
Ausgabe C.F.R: *Wieviel hast Du verpulvert?*
Eingabe Benutzer: **25**
Ausgabe C.F.R: *Und fuer was?!?!?*
Eingabe Benutzer: **Lebensmittel**

Ausgabe C.F.R: *25.0 Euro wurden Dir abgezogen fuer: Lebensmittel*

5 Beschreibung Befehl 5: Kontostand abrufen

Sequenzdiagramm:



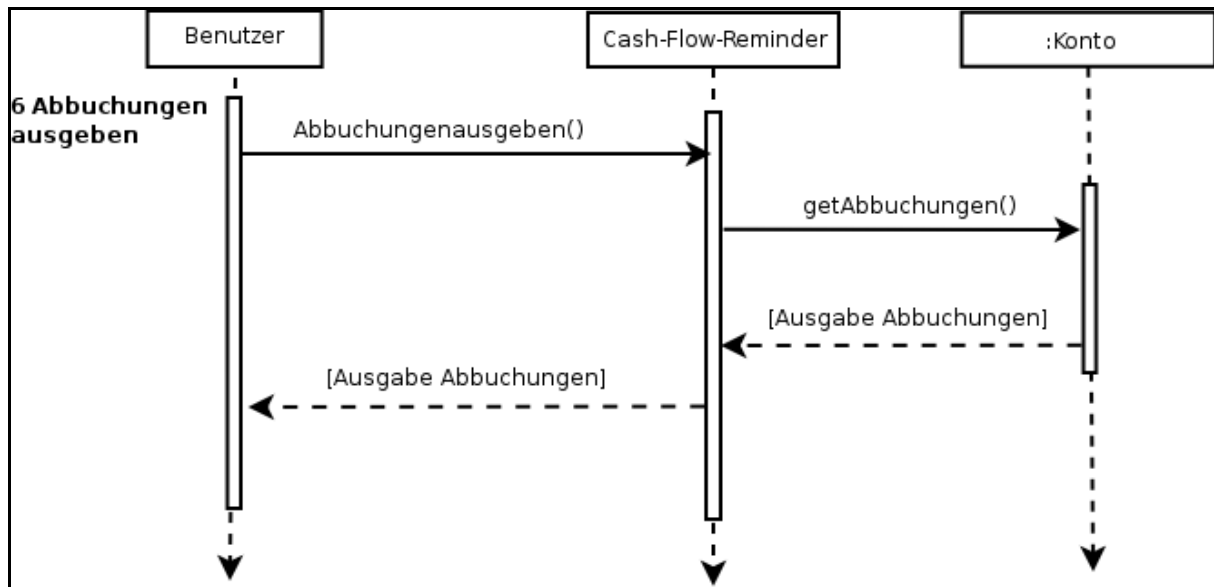
Bsp: Benutzer möchte seinen Kontostand abrufen:

Eingabe Benutzer: **5**

Ausgabe CFR: *Fuer diesen Monat hast Du nur noch 375.0 Kroeten.*

6 Beschreibung Befehl 6: Alle bisherigen Abbuchungen ausgeben

Sequenzdiagramm:



Bsp: Benutzer möchte seine bisherigen Abbuchungen ansehen:

Eingabe Benutzer: **6**

Ausgabe CFR: *So schnell geht's dahin...*

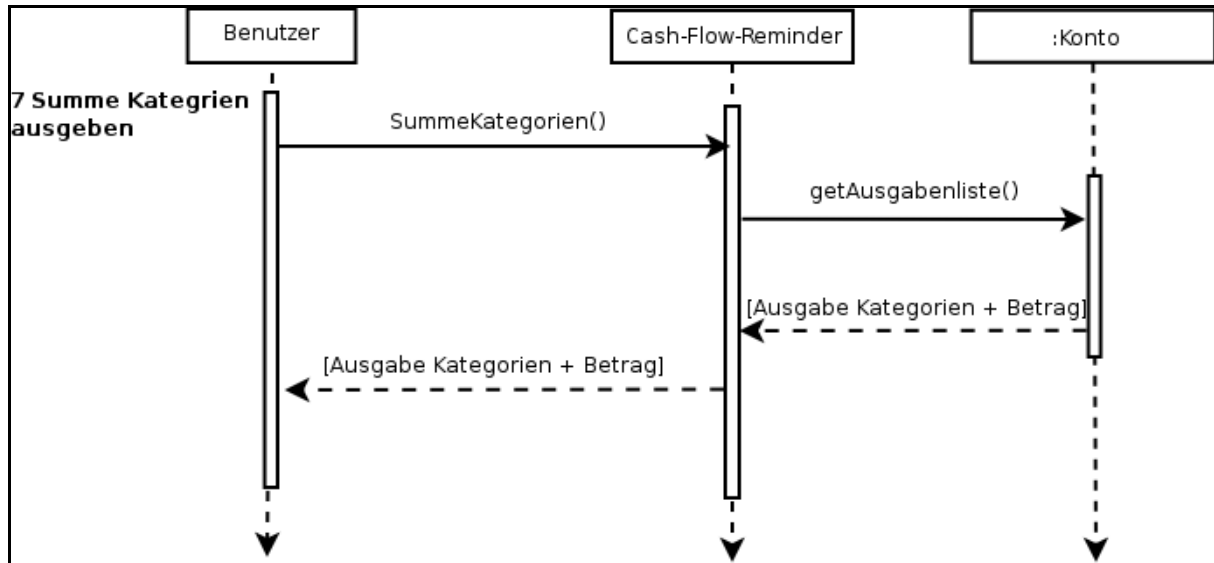
Diesen Monat hattest Du folgende Ausgaben:

Fri Nov 25 14:06:41 CET 2005 Miete 300.0 Euro

Fri Nov 25 14:07:00 CET 2005 Lebensmittel 25.0 Euro

7 Beschreibung Befehl 7: Summe Ausgaben der Kategorien

Sequenzdiagramm:



Bsp: Benutzer möchte die Kategorien mit bisherigen Ausgaben ansehen:

Eingabe Benutzer: **7**

Ausgabe CFR: *Jetzt siehst Du mal, wohin die Kohle geflossen ist:*

Miete: 300.0 Euro

Lebensmittel: 25.0 Euro

8 Beschreibung Befehl 8: Menü anzeigen

Bsp: Benutzer möchte die einzelnen Menüpunkte nochmal angezeigt bekommen:

Eingabe Benutzer: **8**

Ausgabe CFR: *Bitte einen Befehl durch die entsprechende Zahl auswaehlen, und mit return bestätigen:*

1. Neues Konto erstellen
2. Naechster Monat
3. Fixe Abbuchung definieren
4. Variable Abbuchung eingeben
5. Kontostand abrufen
6. Alle bisherigen Abbuchungen ausgeben
7. Summe Ausgaben der Kategorien
8. Menue anzeigen
9. Programm beenden

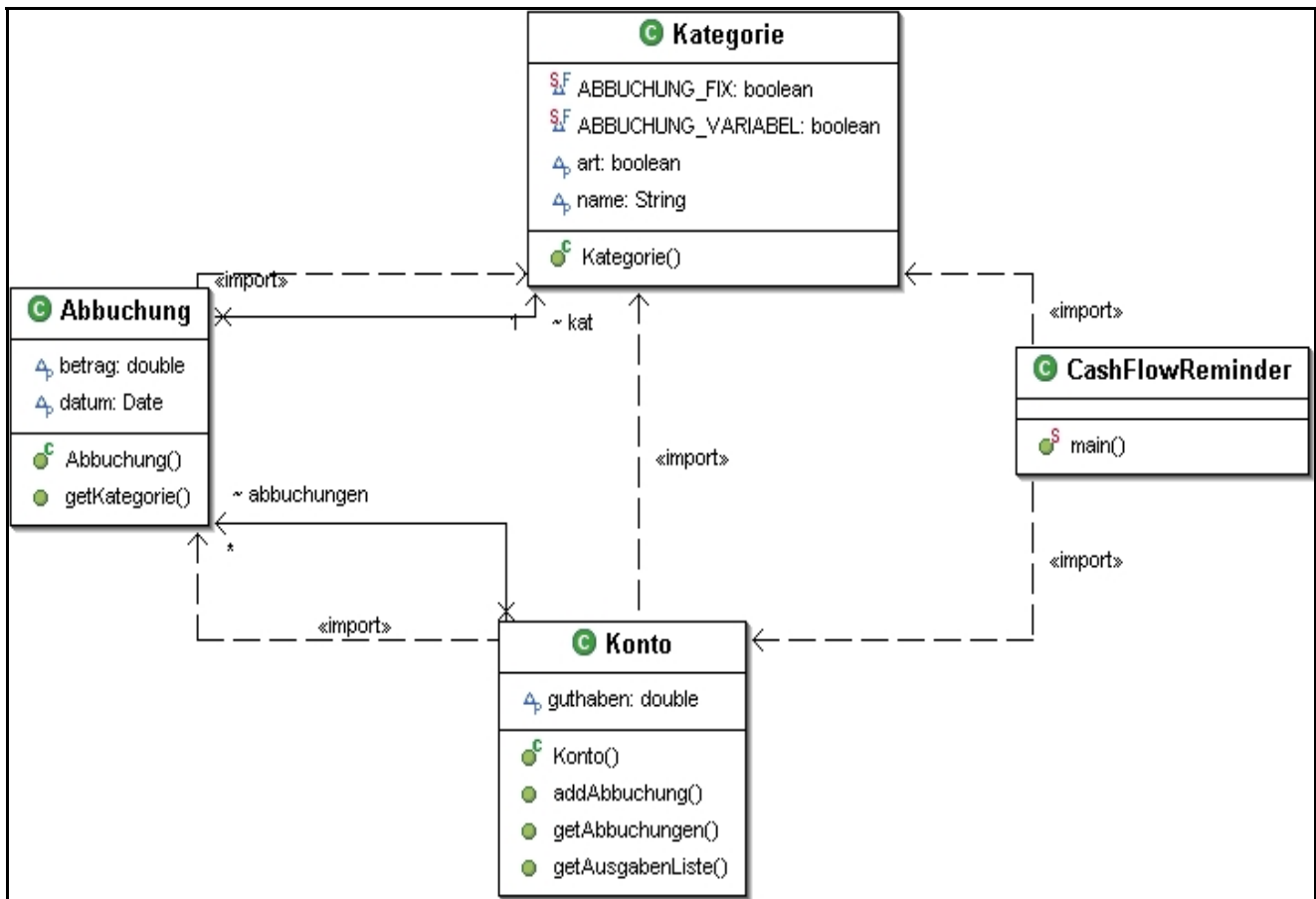
9 Beschreibung Befehl 9: Programm beenden

Bsp: Benutzer möchte den Cash-Flow-Reminder beenden:

Eingabe Benutzer: **9**

=> Programm wird beendet.

10 Klassendiagramm:



Erklärung der Klassen:

1 Klasse Cash-Flow-Reminder

In dieser Klasse wird die main-Methode aufgerufen bzw. automatisch das Hauptmenü ausgegeben. Der Benutzer wird aufgefordert, durch Eingabe einer Zahl von 1-9 einen Menüpunkt auszuwählen. Eingabe wird durch `readLine()` gelesen und über die 9 cases Methoden aufgerufen, die im Anschluss erklärt werden:

Methoden:

Mit *addKonto()* kann der Benutzer ein neues Konto erstellen bzw. sein monatliches Guthaben bestimmen. Eingabe wird durch `readLine()` gelesen, bei richtiger Eingabe erscheint die Ausgabe des Betrages (siehe 2.1). Wenn keine Zahl eingegeben wird, wird eine Exception aufgerufen, es kommt eine Fehlermeldung und es muss wieder ein Hauptmenübefehl eingegeben werden (eine Zahl von 1-9).

Mit *FixeAbbuchungdefinieren()* bzw. *VariableAbbuchungdefinieren()* werden die Höhe der Abbuchung sowie die Kategorie abgefragt. Eingaben werden durch `readLine()` gelesen, bei richtiger Eingabe erscheint eine Ausgabe mit Betrag bzw. Kategorie (siehe 2.3). Bei falscher Eingabe wird eine Exception aufgerufen, es kommt eine Fehlermeldung und es muss wieder ein Hauptmenübefehl eingegeben werden (eine Zahl von 1-9).

Mit *Kontostandabrufen()* kann der Kontostand abgerufen werden. Es wird die Methode `getGuthaben()` (Klasse Konto) abgerufen.

Mit *Abbuchungenausgeben()* können alle bisher getätigten Ausgaben ausgegeben werden (Datum, Betrag + Kategorie). Es wird die Methode `getAbbuchungen()` (Klasse Konto) abgerufen.

Mit *SummeKategorienausgeben()* werden die Kategorien mit der jeweiligen Summe ausgegeben. Es wird die Methode `getAusgabenListe()` (Klasse Konto) abgerufen.

Mit *menueanzeigen ()* wird das Hauptmenü angezeigt. Es werden alle Menüpunkte mit Erklärung ausgegeben.

2 Klasse Kategorie

Variablen:

String name (Name der Kategorie)

boolean art (fix oder variabel)

Methoden:

getName() => Name wird abgerufen (je nach Eingabe des Benutzers)

getArt() => Art wird abgerufen (fix oder variabel)

3 Klasse Abbuchung

Variablen:

Date datum

double Betrag

Kategorie kat

Methoden:

getDatum() => Datum wird abgerufen

getBetrag() => Betrag wird abgerufen

getKategorie() => Kategorie wird abgerufen

4 Klasse Konto

Variablen:

double guthaben;

double startguthaben;

Vector abbuchungen

Methoden:

addAbbuchung() => Es wird dem Vektor ein Element hinzugefügt bzw. eine Abbuchung gebildet. Die Abbuchung beinhaltet Betrag, Datum u Kategorie bzw wird der Betrag vom Guthaben abgezogen.

GetAbbuchungen() => liefert String-Liste aller Abbuchungen zurück

getAusgabenListe() => Im Vektor werden alle Ausgaben überprüft. Bei jeder neuen Kategorie wird eine Kategorie gebildet und ein Element hinzugefügt, falls eine Kategorie mehrmals vorhanden ist, werden die Beträge aufsummiert.

NegativerKontostand() => Boolean-Methode => gibt true zurück wenn der Kontostand negativ ist. Sollte mehr abgebucht werden, als Guthaben vorhanden so ertönt ein beep-signal und ein Warnhinweis.

Installation und Wartung

Im Dos-Fenster kann das Programm bereits ausgeführt werden.

Da es sich erst um einen Prototypen handelt und er noch keine Datenbank-Anbindung beinhaltet, kann sich das Programm getätigte Ausgaben noch nicht „merken“. Alle Schritte können jedoch innerhalb einer Sitzung durchgeführt werden.