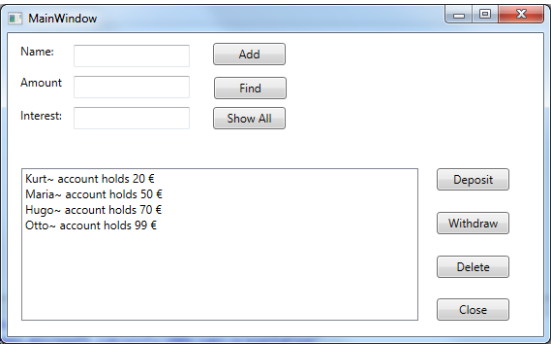


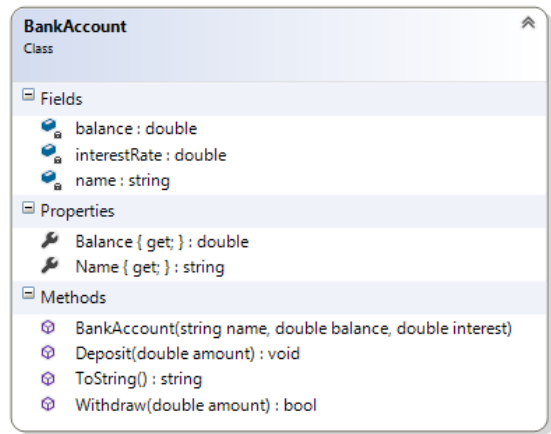
09_Bank

Schreibe eine Applikation, mit der man Bankkonten verwalten kann. Die Applikation soll folgendes GUI haben:



Folgende Klassen seien gegeben für die Implementierung:

Klasse BankAccount



Felder:

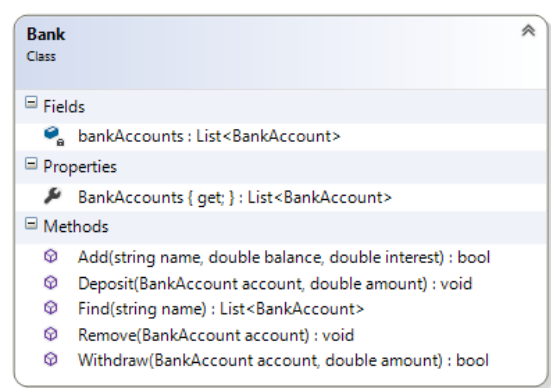
Den aktuellen Kontostand speichert das Feld *balance*, den Zinssatz *interestRate* und den Kontoinhaber *name*.

Methoden:

- Der Konstruktor initialisiert alle Felder.
- Deposit – erhöht den Kontostand um den gewünschten Betrag (**amount**).
- Withdraw - verringert den Kontostand um den gewünschten Betrag (**amount**), falls möglich ist. Achte darauf, dass der Kontostand nicht negativ sein darf. Wenn der Betrag (**amount**) zu hoch ist liefert die Methode false, ansonsten true.

- ToString – liefert die string-Darstellung der Objekte dieser Klasse.

Klasse Bank



Felder:

- bank - ist eine Liste von BancAccount-Objekten.

Property:

- BanAccounts – liefert die Liste aller Bankkonten.

Methoden:

- Add – fügt ein BankAccount-Object in die Collection **bank** ein. Dabei soll geprüft werden, ob **balance** und **interest** positive Zahlen sind. Bei fehlerhaften Werten soll kein Objekt eingefügt und **false** zurückgegeben werden.

- Find – sucht nach den BankAccounts mit einem bestimmten Namen und liefert diese Accounts in einer Liste.
- Remove – löscht ein BankAccount (**account**).
- Deposit – ermöglicht das Einzahlen eines Betrages (**amount**) auf ein Konto (**account**). Achte darauf, dass du keinen negativen Wert (**amount**) addierst!
- Withdraw - ermöglicht das Abheben eines Betrages (**amount**) von einem Konto (**account**). Der Rückgabewert informiert, ob dies möglich war. Achte darauf, dass du keinen negativen Wert (**amount**) abheben lässt!

Gutes Gelingen!