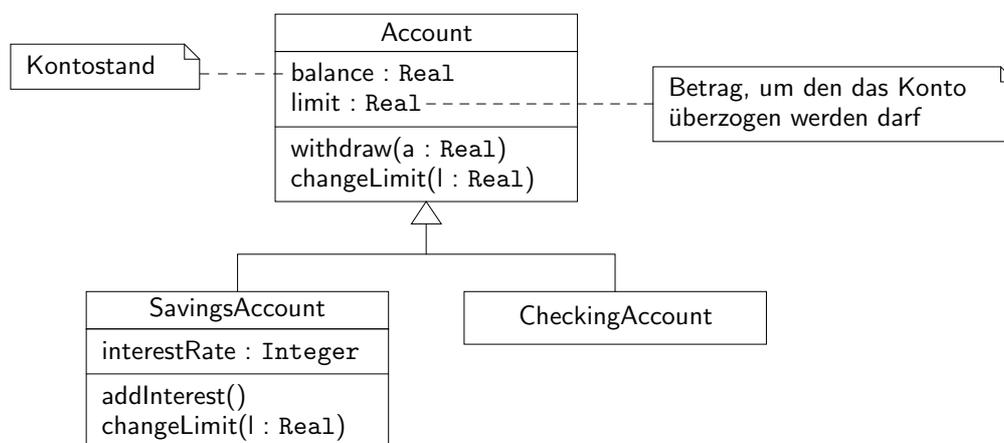


Kurzprüfung zu Formale objektorientierte Software-Entwicklung

Es sei das folgende Klassendiagramm BankAccounts als Modell für die Konten einer Bank gegeben:



Weiter sei I eine (leere) Query-Interpretation über BankAccounts.

Aufgabe 1

(5 Punkte)

Für die Operation `addInterest` der Klasse `SavingsAccount` in `BankAccounts` sei die folgende Operationsspezifikation *AddInterest* gegeben:

```

context SavingsAccount::addInterest()
pre: interestRate >= 0 and balance >= 0
post: balance = balance@pre*(1+interestRate/100)
    
```

1. Berechnen Sie $PreSat_I^{BankAccounts}(AddInterest)$.
2. Berechnen Sie $dom(\llbracket AddInterest \rrbracket_I^{BankAccounts})$.

Es darf vorausgesetzt werden, dass *AddInterest* wohlgeformt ist.

Aufgabe 2

(5 Punkte)

Geben Sie für die Operation `withdraw` der Klasse `Account` in `BankAccounts` eine bezüglich I wohlgeformte Operationsspezifikation *AccountWithdraw* an, so dass folgende Bedingungen erfüllt sind:

1. Vom Kontostand `balance` wird mindestens der als Parameter übergebene Wert a abgezogen.
2. Der Kontostand ist vor und nach Ausführung der Operation `withdraw` größer oder gleich dem Negativen des Werts von `limit`.

Begründen Sie kurz, dass die Operationsspezifikation *AccountWithdraw* bezüglich I wohlgeformt ist.

Aufgabe 3

(6 Punkte)

Es seien die folgenden beiden Operationsspezifikationen über `BankAccounts` gegeben:

```
context Account::changeLimit(1 : Real)
pre: 0 <= 1 and -1 < balance
post: limit = 1

context SavingsAccount::changeLimit(1 : Real)
pre: limit = 0
post: limit = limit@pre
```

1. Geben Sie die Normalform für die Menge der beiden Operationsspezifikationen im Kontext der Klasse `SavingsAccount` an.
2. Entscheiden Sie unter Angabe einer kurzen Begründung, ob die Menge der beiden Operationsspezifikationen bezüglich I wohlgeformt ist.