Medieninformatik(B): ATS5 - Funktionale und Reaktive Programmierung, Prof. Knabe, 2014-11-03

Aufgabe Konto: Scala-Klassen, Beerbung, Begleitobjekt, JUnit

Gegeben ist der JUnit-kompatible Testtreiber `KontoTest.scala`. Studieren Sie ihn.
Schreiben Sie nun folgende Klassen zur Modellierung einer Bank und testen Sie diese mit dem Testtreiber, ohne ihn zu verändern.

Klasse Person mit Konstruktorparameter name: String

Basisklasse Konto mit nicht änderbarem Attribut inhaber: Person und nur geschütztem saldoCt: Long

Die Klasse soll folgende Methoden anbieten:

/**Den aktuellen Saldo liefern*/
saldoCt: Long

/**Den Saldo um den Betrag >0 erhöhen*/
einzahlen(betragCt: Long)

/**Den Betrag >0 vom Saldo abziehen und beim Zielkonto addieren*/
ueberweisen(zielKonto: Konto, betragCt: Long)

/**Die Guthaben/Kreditzinsen für die Anzahl Tage aufsaldieren/abziehen und liefern (Jahr=360Tage)*/
verzinsen(anzahlTage: Int): Long

Unterklasse SparKonto mit Konstruktorparameter inhaber: Person und zinssatzPA: Double.
Bei dieser Kontoart sollen die Methoden ueberweisen und verzinsen so überschrieben werden, dass die folgenden Regeln für Sparkonten eingehalten werden:
* Der Saldo darf nicht negativ werden!
* Guthaben werden mit dem im Konstruktor angegebenen Jahres-Zinssatz verzinst. Dieser ist als Faktor auzufassen, z.B. 0.12 für 12%.

Unterklasse GiroKonto mit Konstruktorparameter inhaber: Person und kreditrahmenCt: Long.
Bei dieser Kontoart sollen die Methoden ueberweisen und verzinsen so überschrieben werden, dass die folgenden Regeln für Girokonten eingehalten werden:
* Der Saldo darf durch ueberweisen nicht den Kreditrahmen überschreiten!
  Achtung: kreditrahmenCt ist eine positive Zahl, saldoCt darf nicht unter -kreditrahmenCt sinken!
* Guthaben werden nicht verzinst.
* Überziehungen werden mit eingestellten Kreditzinssatz als Jahres-Zinssatz verzinst. Dieser ist als Faktor auzufassen, z.B. 0.12 für 12%.

Für die Basisklasse Konto ist ein Begleitobjekt zu erstellen, das 3 Factory-Methoden namens apply definiert:
Mit Angabe von nur dem inhaber wird ein einfaches Konto-Objekt geliefert.
Mit Angabe von inhaber und zinssatzPA wird ein entsrechendes SparKonto geliefert.
Mit Angabe von inhaber und kreditrahmenCt wird ein entsprechendes GiroKonto geliefert.
Der Vorteil von apply ist, dass man diesen Namen beim Auruf weglassen kann. Also reicht Konto(new Person("Meier"), 0.02) aus, wenn man Konto.apply(new Person("Meier"), 0.02) aufrufen will.
Außerdem soll Konto eine Variable kreditzinssatzPA: Double exportieren. Diese Variable ist als Jahres-Zinsfaktor (z.B. 0.2 = 20%) in der Methode verzinsen im GiroKonto zu benutzen.