

Projekt Bankkonten:

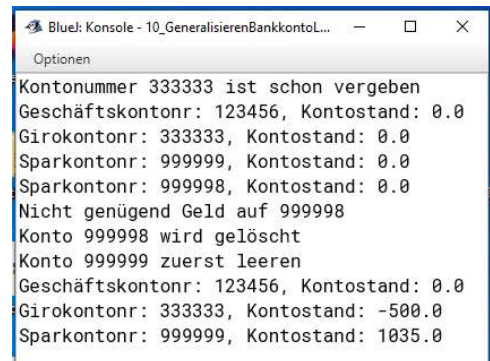
1.  Lies im Buch S.90, 1 durch.
 - a) Bearbeite Aufgabe a) in dein Heft.
 - b) Bearbeite Aufgabe b) mit `violetumeditor`.
2.  Erstelle nun ein neues Projekt `WHGBank` und lege die benötigten vier Klassen mit `KONTO` als Oberklasse an.
 - a) Jedes Konto hat zwei Attribute. Vereinbare diese und erstelle den Konstruktor der Klasse `KONTO`, wobei jedes neue Konto anfänglich leer sein soll. Die `kontonummer` soll vom Benutzer gewählt werden können.
 - b) Erstelle zwei Methoden `abheben` und `einzahlen`, die jeweils ohne Überprüfung den Kontostand um den übergebenen Wert erhöhen bzw. reduzieren sowie `kontostandGeben` und `kontonummerGeben`, die man bei jeder Kontoart benötigt und die den jeweiligen Wert zurückgeben.
3.  `SPARKONTO` ist eine Unterklasse von `KONTO`.
 - a) Du benötigst zusätzlich einen `zinssatz`, der auch an den Konstruktor übergeben wird. Dieser ruft den Konstruktor der Oberklasse auf und speichert den ebenfalls übergebenen Zinssatz ab.
 - b) Die `abheben`-Methode wird verändert, weil du zusätzlich prüfen musst, ob genügend Geld auf dem Konto vorhanden ist. Falls ja, rufst du anschließend `abheben` aus der Oberklasse auf.
 - c) Erstelle noch die Methoden `zinssatzGeben` sowie `verzinsen`, die den von dir berechneten Zins einzahlt.
4.  Auch `GIROKONTO` ist eine Unterklasse von `KONTO`. Der Konstruktor gibt seinen einzigen benötigten Wert direkt an den Oberklassenkonstruktor weiter. Beim `abheben` muss geprüft werden ob man innerhalb des Dispokredits von 2000 Euro bleibt.
5.  Beim Konstruktor von `GESCHÄFTSKONTO` wird der `dispo` mit abgefragt und festgelegt. Erstelle außerdem die passende `abheben`-Methode sowie `dispokreditSetzen` und `dispokreditGeben`.
6.  Erstelle nun die Klasse `VERWALTUNG`.
 - a) In einem Feld `kontenliste` werden mehrere Konten verwaltet.
 - b) Eine Methode `kontonummerPrüfen` gibt einen Wahrheitswert zurück, ob eine `kontonummer` noch frei ist (zwei Möglichkeiten für Wdh).

- c) Erstelle eine Methode `sparkontoEinrichten`, die mit der Methode aus b) prüft, ob die gewünschte Kontonummer noch frei ist. Falls nein, wird eine Fehlermeldung ausgegeben (siehe Screenshot), falls ja, wird ein Objekt des entsprechenden Kontentyps neu angelegt und in die `kontenliste` aufgenommen.
- d) Implementiere auf die gleiche Weise `girokontoEinrichten` sowie `geschäftskontoEinrichten`.
- e) Eine Methode `kontoLöschen` soll das Konto zu einer übergebenen Kontonummer löschen, falls sein Kontostand auf 0 ist, ansonsten Fehlermeldung (siehe Screenshot).
- f) Eine Methode `einzahlen` ermöglicht das Einzahlen auf das Konto mit der übergebenen Kontonummer.
- g) Eine Methode `abheben` führt das Abheben vom Konto mit der einzugebenden Kontonummer durch oder liefert eine Fehlermeldung (Screenshot).

nach Hefteintrag zu 2.7.:

- h) Eine Methode `verzinsen` verzinst alle Sparkonten und gibt mit einem Hinweis „Zinsen fällig!“ alle neuen Kontostände am Bildschirm aus.
- i) Eine Methode `kontodatenGeben` gibt die Daten (Kontoart, Kontonummer und Kontostand) aller Konten aus.

7.  Nun implementierst du in `VERWALTUNG` eine Methode `testen`. Entweder erstellst du sie komplett selbst, um folgendes Ergebnis zu erhalten, oder du kopierst ihren Inhalt aus R: und ergänzt sie anschließend noch um eigene Befehle.



```

Optionen
Kontonummer 333333 ist schon vergeben
Geschäftskontonr: 123456, Kontostand: 0.0
Girokontonr: 333333, Kontostand: 0.0
Sparkontonr: 999999, Kontostand: 0.0
Sparkontonr: 999998, Kontostand: 0.0
Nicht genügend Geld auf 999998
Konto 999998 wird gelöscht
Konto 999999 zuerst leeren
Geschäftskontonr: 123456, Kontostand: 0.0
Girokontonr: 333333, Kontostand: -500.0
Sparkontonr: 999999, Kontostand: 1035.0
  
```