

Übungen zu „Software-Entwicklung“

Objektgrundlagen und einfache Datenobjekte

Profs. Dres. Rathke & Thies

Mit großer Freude ist Ihre schnelle Implementierung der neuen Kontenverwaltung von Ihrem Kunden, der Spar Nix Bank AG, aufgenommen worden. Ihr Ansprechpartner Bodo Bank hat daraufhin gleich versucht, die neuen Klassen, die sie ihm übermittelt haben, zu testen, bevor diese in die Produktivumgebung übernommen werden, ist dabei jedoch auf einige Probleme gestoßen. Bei dem letzten Telefonat hat er Sie gebeten, ob Sie sich mal seine Testklasse `BankDemo` anschauen könnten, und ihm bei der Fehlersuche und Fehlerbeseitigung helfen könnten.

Damit Sie Bodo Bank helfen können, laden Sie die Datei 'IT1_Banking_08_1.zip' und importieren Sie das Projekt in Eclipse.

- 1) Schauen Sie sich in der Klasse `BankDemo` (im Paket `de.hdm.bankProject`) den Code zur Aufgabe 1 (Uebung 08 Aufgabe 1) an. Wo liegt hier der Fehler? Korrigieren Sie diesen!

Der Code erzeugt kein `Account`-Objekt. In diesem einfachen Fall erzeugt der Compiler eine Warnung. Zur Laufzeit wird eine `Null-Pointer-Exception` geworfen. Korrektur:

```
Account myFirstAccount = new Account();
```

- 2) Schauen Sie sich in der Klasse `BankDemo` den Code zur Aufgabe 2 (Uebung 08 Aufgabe 2) an. Dieser Code erzeugt ein `Bank`-Objekt und ein `Account`-Objekt. Wie viele Referenzen zu diesen Objekten existieren, nachdem der Code dieser Aufgabe ausgeführt ist? Ist eines der Objekte reif für den Garbage Collector?

Es existiert eine Referenz für das `Bank`-Objekt und eine für das `Account`-Objekt. Keines dieser Objekte ist reif für den Garbage Collector.

- 3) Wie löscht ein Programm ein Objekt, das es erzeugt hat?

Ein Programm löscht ein Objekt nicht explizit. Man könnte alle Referenzen auf ein Objekt entfernen, in dem man sie auf `null` setzt, so dass das Objekt reif für den Garbage Collector wird.

- 4) Schauen Sie sich in der Klasse `BankDemo` den Code zur Aufgabe 4 (Uebung 08 Aufgabe 4) an. Welche Fehler enthält der Code?

Zeile 2: `„myThirdAccount.updateBalance = 5678.90;“` ist falsch, da `updateBalance` kein Attribut sondern eine Methode ist. Methoden können keine Werte zugewiesen werden!

Zeile 3: `„double myBalance = myThirdAccount.getBalance;“` ist falsch! Es muss heißen: `„double myBalance = myThirdAccount.getBalance();“` Es fehlen die runden Klammern beim Methodenaufruf!

- 5) Schauen Sie sich in der Klasse `BankDemo` den Code zur Aufgabe 5 (Uebung 08 Aufgabe 5) an. Welchen Wert ergibt der Ausdruck `„myFourthAccount == myFifthAccount“`? Begründen Sie Ihre Aussage!

Der Ausdruck ergibt `false`, da er die Identität der beiden Objekte testet und nicht deren inhaltliche Gleichheit.

- 6) Erstellen Sie in der Klasse `BankDemo` zur Aufgabe 6 (Uebung 08 Aufgabe 6) Code, der folgendes ausführt:

- 1) Erstellen Sie eine Instanz der Klasse `Bank` mit dem Namen „Test Bank e.V.“
- 2) Erstellen Sie eine Instanz der Klasse `Account` und weisen Sie diesem Konto einen Eigentümer und eine Kontonummer zu.
- 3) Fügen Sie das `Account` Objekt zu Ihrer Instanz der Klasse `Bank` hinzu.
- 4) Buchen Sie auf dieses Konto eine Gehaltszahlung in Höhe von €2000.-
- 5) Lassen Sie sich von diesem Konto den Kontoauszug ausgeben.
- 6) Lassen Sie sich eine Übersicht aller Konten der Bank ausgeben.

Lösung:

```
Bank mySecondBank = new Bank("Test Bank e.V.");
Account mySixthAccount = new Account(new Customer("Franz", "Freund"), 2221);
mySecondBank.addAccount(mySixthAccount);
mySixthAccount.book(2000, "Gehalt 05/09");
mySixthAccount.printAccountStatement();
mySecondBank.printAccounts();
```