

Übung 07: Klassen und Objekte

Abgabetermin: 09. 12. 2008

Name: _____

Matrikelnummer: _____

Gruppe: G1 (Prähofer)

G2 (Wolfinger)

G3 (Wolfinger)

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Pkte
Aufgabe 07.1	12	<input type="checkbox"/>	Java-Programm Testergebnisse	Java-Programm	<input type="checkbox"/>	
Aufgabe 07.2	12	<input type="checkbox"/>	Java-Programm Testplan Testergebnisse	Java-Programm	<input type="checkbox"/>	

Aufgabe 07.1: Schneiden von Rechtecken

Entwickeln Sie ein Programm, das es erlaubt, Rechtecke zu schneiden. Implementieren Sie eine Klasse `Rectangle` für Rechtecke. Implementieren Sie Methoden für das Einlesen eines Rechtecks, für die textuelle Ausgabe eines Rechtecks und für das Schneiden von zwei Rechtecken (soll als Ergebnis das Schnittrechteck liefern, wenn sich die Rechtecke schneiden, und `null`, wenn sich die Rechtecke nicht schneiden).

Schreiben Sie dann eine `main`-Methode, die von einer Eingabedatei eine Folge von Paaren von Rechtecken einliest, den Schnitt der beiden Rechtecke berechnet und die Rechtecke ausgibt.

Hinweis:

Auf der Homepage der LVA finden Sie das Programm `Window`, welches eine einfache grafische Ausgabe ermöglicht. Sie können diese Klasse verwenden, um die Rechtecke grafisch auszugeben:

- Öffnen Sie ein Fenster mit

```
void Window.open();
```

- Zeichnen Sie ein Rechteck mit

```
void Window.drawRectangle(x, y, width, height);
```

- Geben Sie ein gefülltes Rechteck (z.B. für den Schnitt) aus mit

```
void Window.fillRectangle(x, y, width, height, color);
```

wobei `color` vom Typ `java.awt.Color` ist und mit `Color.red`, `Color.blue`, `Color.yellow`, `Color.green`, etc. Konstanten für gängige Farben zur Verfügung stehen.

z.B.: `Rectangle r;`

...

```
Window.fillRectangle(r.x, r.y, r.width, r.height, Color.RED);
```

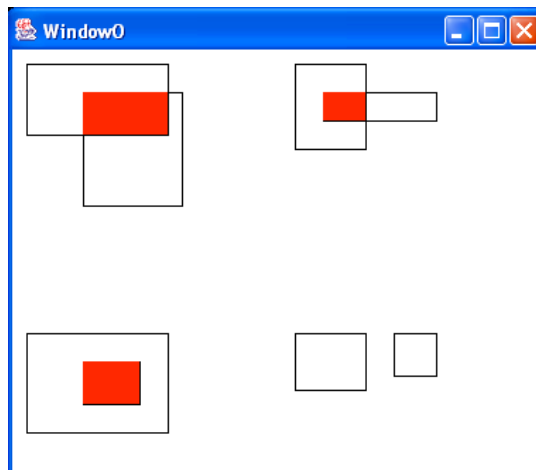
Testdaten:

```
10 10 100 50  
50 30 70 80
```

```
10 200 100 70  
50 220 40 30
```

```
200 10 50 60  
220 30 80 20
```

```
200 200 50 40  
270 200 30 30
```



Abzugeben ist:

- Java-Programm
- Testergebnisse

Aufgabe 07.2: Kontoverwaltung

Schreiben Sie für eine kleine Privatbank ein Java-Programm zur Verwaltung von Kundenkonten. Erstellen Sie Klassen für Kunde (`Customer`), Konto (`Account`) und für die Bank (`Bank`).

Ein Kunde (Klasse `Customer`) hat

- Vorname
- Nachname
- Telefonnummer

Ein Konto (`Account`) besteht aus

- Kontonummer
- Kontostand
- Überziehungsrahmen
- Inhaber (der Kunde)

Eine Bank (`Bank`) besteht aus einer Liste von Konten. Da unsere Bank nur sehr exquisite Kunden betreut, ist es ausreichend, wenn bis zu 100 Konten verwaltet werden können. Verwenden Sie ein Array, um eine Menge von `Account`-Objekten zu speichern. Gehen Sie dabei wie beim Beispiel `PhoneBook` aus der Vorlesung vor: Legen Sie ein Array von `Account`-Objekten ausreichender Größe an und verwenden Sie eine `int`-Variable `cnt`, um sich die tatsächliche Anzahl von Objekten zu merken.

Schreiben Sie für eine Bank (`Bank`) folgende Methoden:

- `static int createAccount(Customer owner, double overdraftLimit)`
erstellt ein Konto und liefert die Kontonummer für das erstellte Konto. Die Kontonummer soll hierbei fortlaufend (mit 0 beginnend) vergeben werden. Falls bereits zu viele Konten angelegt werden, soll `-1` als Kontonummer zurückgegeben werden.
- `static boolean deposit(int accountNo, double amount)`
bucht den angegebenen Betrag auf das gegebene Konto auf und liefert `true` falls die Einzahlung erfolgreich durchgeführt wurde; ansonsten `false` (wenn zB das Konto nicht existiert).
- `static boolean withdraw(int accountNo, double amount)`
bucht den angegebenen Betrag vom gegebenen Konto ab und liefert `true` falls die Abhebung durchgeführt wurde; ansonsten `false` (wenn zB das Konto nicht existiert oder die Deckung ungenügend ist).
- `static boolean transfer(int fromAccountNo, int toAccountNo, double amount)`
bucht den angegebenen Betrag vom `from`-Konto ab und bucht ihn auf das `to`-Konto auf. Liefert `true` falls die Überweisung durchgeführt wurde; ansonsten `false` (wenn zB ein Konto nicht existiert oder die Deckung am `from`-Konto ungenügend ist).
- `static double getAccountBalance(int accountNo)`
liefert den aktuellen Kontostand für das gegebene Konto.
- `static double getBalance()`
liefert den Bilanzwert (=Summe aller Kontostände) der Bank.
- `static printAccounts()`
gibt ein Liste aller Konten aus. Die Ausgabe soll wie folgt aussehen:

```
----- Bankauszug -----  
Kontonummer: 0  
Kontoinhaber: Max Mustermann  
Telefonnummer: +43 1 555 666  
Kontostand: 100.00  
Überziehungsrahmen: 1200.00  
-----  
Kontonummer: 1  
Kontoinhaber: Maximilia Mustermann  
Telefonnummer: +43 1 555 666  
Kontostand: 0.00  
Überziehungsrahmen: 2400.00  
-----  
Bilanzsumme: 100.00  
-----
```

Hinweise:

- Definieren Sie selbst wo notwendig und sinnvoll weitere Methoden
- Testen Sie dem zur Verfügung gestellten Verwaltungsprogramm `BankApp.java` ausführlich Ihre Kontoverwaltung. Ein Beispielzenario könnte dabei sein:
 - Legen Sie 5 Kunden an und eröffnen Sie für jeden Kunden ein Konto bei der Bank. Führen Sie anschließend einige Transaktionen durch – zahlen Sie Geld auf die Konten ein, heben Sie Geld und überweisen Sie Beträge von einem Konto auf ein anderes. Geben Sie zwischendurch öfters einen Bankauszug – wie unten angeführt – aus.
- Führen Sie auch ungültige Transaktionen durch (z. B. das Beheben einer Summe, die den Überziehungsrahmen überschreitet, ungültige Parameterwerte wie negative Beträge) und zeigen Sie, dass ihre Implementierung diese nicht zulässt.

Abzugeben ist:

- Java-Programm
- Testergebnisse