Füllt die/der StudentIn aus

Matrikelnummer		Vorname		Nachname
Beispiel 1	Beispiel 2	Gesamtsumme	Nachfrist O ja O nein	TutorIn

Füllt der/die TutorIn aus

2. Abgabe

WS00/01

	Beispiel 1		Beispiel 2	
	70 Punkte		**30 Punkte**	
Programmverständnis				
Konkret				
Hintergrundwissen				
Runtime Quality				
Normalfall				
Sonderfälle				
Source Code Quality				
Stil & Lesbarkeit				
Modularisierung				
Variablenscope				
Dokumentation				
Kommentare				
Pseudocode				
Übersicht				
Summe (pro Beispiel)				

Füllt der/die TutorIn aus

Allgemeine Hinweise

Dateien einlesen

Verwenden Sie zum Einlesen der Dateien die im Skriptum beschriebene Klasse ReadFile. Sie steht auf der EProg-Homepage zur Verfügung.

Testdaten

Die Testdaten, die im Folgenden erwähnt werden, stehen der EProg-Homepage.

Modularisierung

Bei dieser zweiten Abgabe sollten Sie bereits das Programm mittels mehrerer eigener Klassen modularisieren.

Beispiel 1: Banner 70 Punkte

Schreiben Sie ein Programm, das Textzeilen aus einem File am Bildschirm in Blockbuchstaben als Grafik zeichnet.

Das File `banner.txt` (enthalten im unten genannten WinZip-Archiv) enthält in jeder Zeile zuerst eine ganze Zahl, die den Vergrößerungsfaktor für diese Zeile angibt und dann (nach einem Leerzeichen) den Text in Blockbuchstaben, der ausgegeben werden soll.

Im WinZip-Archiv steht für jeden Buchstaben ein File mit der Extension `def`, also z.B. `A.def` für den Buchstaben A. Jedes dieser Files enthält die Beschreibung des Buchstabens in mehreren Zeilen, von denen jede eines der beiden folgenden Formate hat:

`Strich` <*x1*> <*y1*> <*x2*> <*y2*>

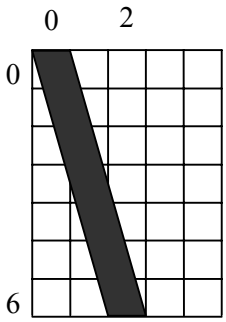
`Kurve` <*x*> <*y*> <*Außenradius*> <*Anfangswinkel*> <*Bogenwinkel*>

Alle Buchstaben sind auf einem 5x7-Raster spezifiziert. Bei der Ausgabe müssen sie um den Vergrößerungsfaktor der jeweiligen Zeile vergrößert werden. Zwischen den Buchstaben bliebt ein Kästchen breit Abstand.

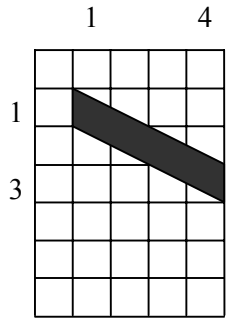
Strich meint einen (breiten) Strich in Form eines Parallelogramms. Dabei ist zu unterscheiden, ob es sich um einen Strich nach rechts oder einen Strich nach unten handelt. Wenn die Differenz von x1 und x2 größer ist als die von y1 und y2, dann handelt es sich um einen Strich nach rechts. Striche, die im Winkel von 45° nach rechts unten weisen, werden daher zu den Strichen nach unten gerechnet.

Die Zahlen <*x1*> und <*y1*> bezeichnen die X und Y-Koordinaten des ersten Kästchens, *in* dem der Strich beginnt, die Zahlen <x2> und <y2> die des Kästchens, *in* dem der Strich endet. Die Strichbreite ist ein Kästchen. Kästchen (0/0) ist links oben, entsprechend dem Koordinatensystem in Java.

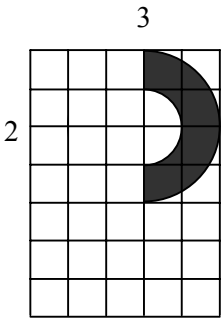
Kurve meint einen Bogen, der so breit ist, wie ein Kästchen. <*x*> und <*y*> bezeichnen den Mittelpunkt des Bogens. Er liegt immer am Kreuzungspunkt von Trennlinien zwischen den Kästchen und nicht im Mittelpunkt eines Kästchens. <*Außenradius*> bezeichnet den Außenradius des Bogens; <*Anfangswinkel*> dessen Anfangswinkel und <*Bogenwinkel*> den Winkel, über den sich der Bogen erstrecken soll. Beide Winkelangaben entsprechen dem 5. und 6. Argument der fillArc-Methode (s.u.), sodaß keine Umrechnungen notwendig sind.



`Strich 0 0 2 6`



`Strich 1 1 4 3`



`Kurve 3 2 2 270 180`

Hinweise:

- Beachten Sie bitte die unterschiedliche Interpretation der Koordinaten: Bei Strichen beziehen sie sich auf Kästchen(-mittelpunkte), bei Kurven auf die Trennlinien zwischen den Kästchen.
- Verwenden Sie zum Zeichnen der Striche die Methode `fillPolygon` und zum Zeichnen der Kurven die Methode `fillArc`, beide in der Klasse `Graphics`.
- Um einen Bogen am Bildschirm darzustellen, zeichnen Sie zuerst ein Kreissegment in der Zeichenfarbe (schwarz), dessen Radius dem Außenradius des Bogens entspricht, und dann in der Hintergrundfarbe (weiß) ein Kreissegment, dessen Radius dem Innenradius (ein Kästchen weniger als der Außenradius) entspricht.

Beispiel 2: Rechnungserstellung 30 Punkte

Aufgabe:

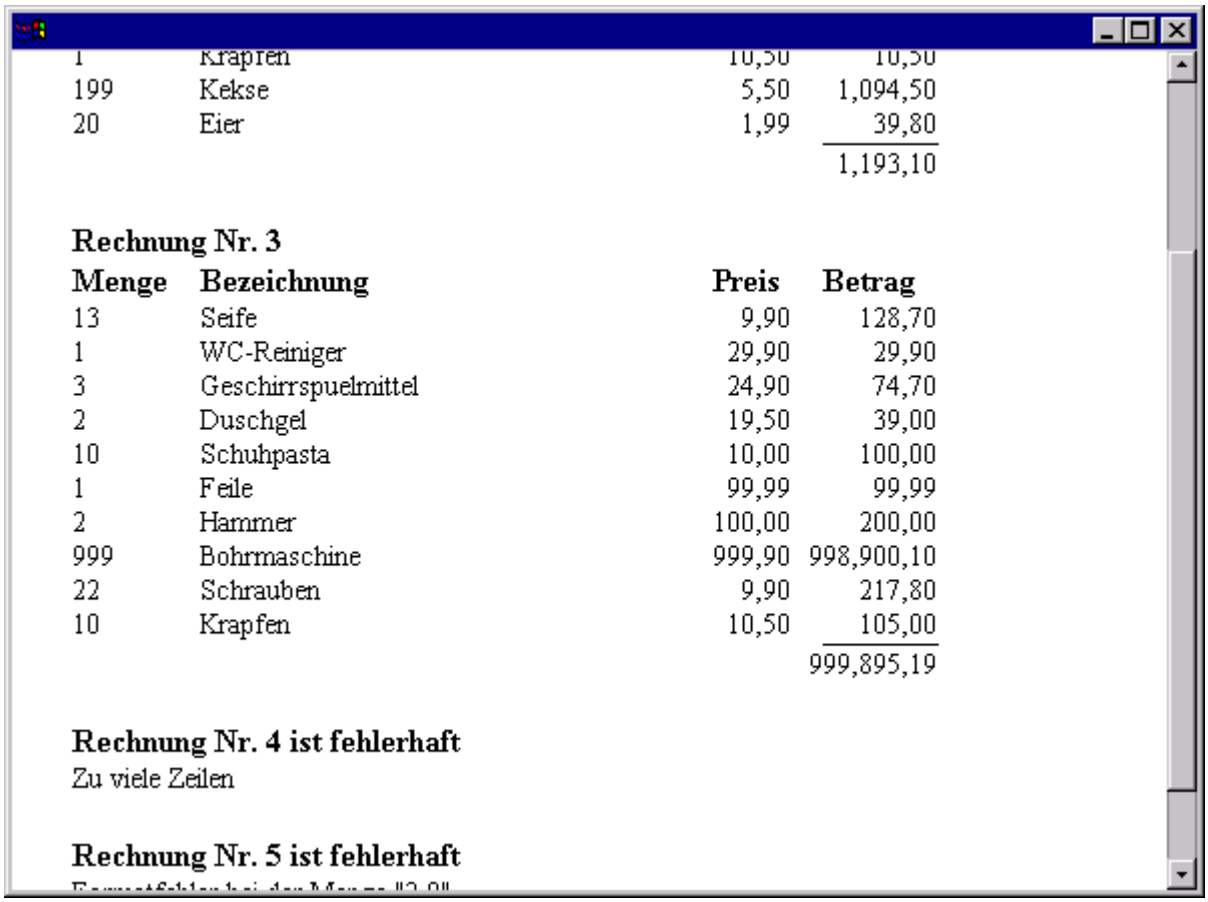
Ändern Sie das Programm, das Sie für die letzte Abgabe geschrieben haben, so daß die Ausgabe mittels Methoden der Klasse `Graphics` am Bildschirm erscheint (in einem `Frame`).

Das Programm soll alle Files, deren Name mit `rechnung` beginnt und mit `.txt` endet, ausgeben – unabhängig davon, wieviele Files mit einem solchen Namen im Directory stehen.

Als Argumente auf der Commandline können dem Programm ein oder zwei Integers übergeben werden. Ein Integer gibt die Größe des Fonts an, in dem die Ausgabe erfolgen soll. Bei zwei Integers gibt das erste Integer die Größe des Fonts für die Kopfzeile(n) an, das zweite die Größe des Fonts für die restlichen Zeilen einer Rechnung. Der Default für die Kopfzeile(n) ist 16 pt (Point), der Default für die restlichen Zeilen ist 14 pt. Die Kopfzeile(n) soll(en) immer fett (`bold`) gedruckt werden, die anderen Zeilen normal (`plain`).

Verwenden Sie die Methoden der Klassen `Font` und `Fontmetrics`, um die Zeilenhöhe und Spaltenbreiten zu berechnen, da im Programm festgeschriebene Konstanten bei wechselnder Fontgröße nicht zum Erfolg führen.

Zum Erreichen der vollen Punktezahl muß eine Möglichkeit vorgesehen werden, die Ausgabe hin- und herzuscrollen, wenn sie nicht in das Window paßt: Entweder mittels `Scrollbar` oder `Scrollpane`.



1	Krapfen	10,50	10,50
199	Kekse	5,50	1,094,50
20	Eier	1,99	39,80
			1,193,10

Rechnung Nr. 3

Menge	Bezeichnung	Preis	Betrag
13	Seife	9,90	128,70
1	WC-Reiniger	29,90	29,90
3	Geschirrspuelmittel	24,90	74,70
2	Duschgel	19,50	39,00
10	Schuhpasta	10,00	100,00
1	Feile	99,99	99,99
2	Hammer	100,00	200,00
999	Bohrmaschine	999,90	998,900,10
22	Schrauben	9,90	217,80
10	Krapfen	10,50	105,00
			999,895,19

Rechnung Nr. 4 ist fehlerhaft
Zu viele Zeilen

Rechnung Nr. 5 ist fehlerhaft