Übungsblatt 7

Ralph Gauges Ursula Rost Katja Wegner

05.12.2007

Aufgabe 1:

Schreiben Sie einen Ausgabeoperator für die Klasse für Komplexe Zahlen aus Aufgabe 3 vom 28.11.2008.

Aufgabe 2:

Schreiben Sie einen Ausgabeoperator für die Klasse *Bruch* aus Aufgabe 2 vom 07.11.2007.

Aufgabe 3:

Erweitern Sie die Klasse *Image* aus Aufgabe 1 vom 28.11.2007 um eine Methode, welche das Bild als TGA Datei speichert.
TGA ist eines der einfachsten Formate, um Bitmap Grafiken zu speichern. Das TGA Format gibt es in mehreren unterschiedlichen Varianten. Wir verwenden hier die einfachste Variante, die lediglich aus einem sogenannten Header besteht, welcher Informationen wie die Höhe und Breite des Bildes und den eigentlichen Bilddaten in unkomprimierter Form enthält.
Der Header kommt als erstes und hat die in Tabelle 1 angegebene Struktur. Was die einzelnen Zahlen bedeuten, soll uns hier nicht weiter interessieren. Die einzelnen Zahlen sind als Byte-Werte (auf den meisten Plattformen **char** Datentyp) aufzufassen, nur die Höhe und die Breite werden als 2 Byte Werte (Word) abgespeichert. Da das TGA Format vor allem für Rechner mit Intel Prozessoren entwickelt wurde, muss man eine kleine Besonderheit beim

Byte Nr.	Wert
0-1	0
2	2
3-11	0
12-13	Breite in Pixeln
14-15	Höhe in Pixeln
16	24
17	32

Tabelle 1: TGA Header

Abspeichern der Höhe und der Breite berücksichtigen. Word-Werte werden auf dieser Plattform in zwei Byte Werte aufgeteilt und diese werden in umgekehrter Reihenfolge geschrieben.
Angenommen das Bild hätte eine Breite von 1025 Pixeln, dies entspräche der Hexadezimalzahl 0x0401. Teilt man diese in zwei Byte Werte, so erhält man 0x04 und 0x01. Will man nun diese Zahl in die Datei schreiben, so muss man zuerst die 0x01 schreiben und dann die 0x04. Dies ist mit den Operatoren die heute behandelt wurden leicht möglich.
Im Anschluss an diesem Header folgen die Bilddaten und zwar jeweils als Rot-, Grün- und Blauwert, wobei jeder Wert ein Byte groß sein muss. D.h. Die Farbe Schwarz hätte die Werte 0 0 0 und die Farbe weiß hätte die Werte 255 255 255. Da dies auch das Format ist, das wir in der Image Klasse verwendet haben, können wir die Daten aus der Image Klasse einfach Byte für Byte in die Datei schreiben. Die Bilddaten werden dabei Zeilenweise von Oben nach Unten geschrieben.