

Beispiel 1: Viewer für Textdateien Version 2 (24 Punkte)

Die vorliegende Übung ist eine Neuauflage der Übung 8. Der wesentliche Unterschied liegt in der Vorgabe, daß die Verwaltung von Objekten mittels der Template-Klasse `list` auf standardisierten Containerklassen aufgebaut werden muß.

Wenn Sie die Übung 10 abgeben, aber die Übung 8 nicht gemacht haben (oder weniger als 12 Punkte erreicht haben), wird die Übung 10 als zwei abgegebene Übungen (max. 48 Punkte) gerechnet.

Schreiben Sie ein Programm, das eine Textdatei mit Text im ASCII Format lesen und am Bildschirm darstellen können soll. Die Textdatei soll aus Zeilen beliebiger Länge bestehen können. Jede Zeile wird durch ein Zeichenpaar CR LF begrenzt ($\text{ORD}(\text{CR}) = 13$, $\text{ORD}(\text{LF}) = 10$).

Die Datei soll als Kommandozeilenparameter angegeben werden. Ihr Programm soll die Datei seitenweise am Bildschirm ausgeben. Nach jeder Seite soll der Benutzer wählen können, ob er die nächste Seite oder die vorhergehende Seite sehen oder das Programm beenden möchte.

Die Ausgabe am Bildschirm soll sich nicht nach dem Zeilenumbruch in der Datei richten. Die Ausgabe am Bildschirm soll Wörter am Zeilenende nicht trennen, aber die Zeilen so gut wie möglich ausnützen. Nur nach jedem Absatz soll eine neue Zeile begonnen werden.

Ein optionaler Kommandozeilenparameter soll es ermöglichen, die Ausgabe so zu gestalten, daß nach jedem Satz eine neue Zeile begonnen wird.

Sie sollen die gesamte Datei bei Programmstart in den Hauptspeicher einlesen. Verwenden Sie Klassen, um verschiedene Elemente der Datei zu modellieren: Datei, Absätze, Sätze, Wörter. Die Objekte sollen Ausgabefunktionen besitzen, die in der Lage sind, sich und ihre Elemente entsprechend des gewählten Ausgabeformats auszugeben. Manche Ihrer Objekte werden Referenzen auf andere Objekte enthalten. Nach dem Einlesen der Datei soll die Datei im Hauptspeicher als eine Datenstruktur dargestellt werden, die aus einer Menge von miteinander verknüpften Exemplaren Ihrer Objekte besteht. Ihre schriftliche Abgabe soll diese Datenstruktur verbal und graphisch beschreiben.

Beim Lesen der Datei gelten folgende lexikalischen Regeln: Wörter werden durch „“, „“, „“, „“, „“, „:“ oder ein Satzende begrenzt. Ein Satzende wird durch „?““, „!““, „.““ angezeigt. Ein Absatz besteht entweder aus Text gefolgt von zwei Zeilenumbrüchen oder zwei Zeilenumbrüchen.

Ihr Programm soll durch externe Ereignisse nicht zum Absturz zu bringen sein. Dazu gehören Dateien, auf die nicht zugegriffen werden kann, oder Dateien, die sich nicht an die Spielregeln halten (z.B. .EXE Dateien), oder solche, die zu groß für den Hauptspeicher sind. Brechen Sie statt dessen die Programmausführung mit einer sinnvollen Fehlermeldung ab.

Bauen Sie die Verwaltung von Listen von Objekten ausschließlich auf der Template-Klasse `list` auf. (Header file „list“ ohne „h“)

Achten Sie bitte darauf, bei Ihrer schriftlichen Abgabe im Header Ihre Account-Nummer anzugeben, damit wir Ihre elektronische Abgabe finden können.

Abgabe: bis 21.6. 23:59 Uhr elektronisch (Verzeichnis uebung10 einrichten!) und am 21.6. in der Sprechstunde des Instituts oder am 24.6. im HS10 vor der Klausur auf Papier (Listing der Quelldateien und Dokumentation).