

3. Praktikum zum Programmieren II

Ablauf:

- a) Bearbeiten Sie die Aufgaben in Teams zu zwei Personen. Sie erzeugen aber eine Lösung pro Team.
- b) Achten Sie bei der Teamzusammenstellung darauf, dass **Beide** an der Lösung der Aufgabe tatsächlich arbeiten und sich gegenseitig kontrollieren und ergänzen. Nicht gewünscht ist es, dass eine Person die Aufgabe(n) erledigt und die zweite Person wohlwollend aber passiv danebensitzt.
- c) Wenn Sie die Aufgaben erledigt haben, geben Sie dem Praktikumsleiter Nachricht. Die Arbeit wird dann **im Termin** besprochen und abgenommen.
- d) Lassen Sie sich mit der Fertigstellung nicht bis zum Ende des Praktikumstermins Zeit, da Sie ggf. noch Nacharbeiten an den Aufgaben durchführen müssen. Sie können das Praktikum erst verlassen, wenn die Aufgaben abgenommen sind und gegen Ende des Termins wird es erfahrungsgemäß zeitlich eng werden.
- e) Bevor Sie den Praktikumsleiter rufen, machen Sie Qualitätstests anhand folgender Checkliste:
 - i) Jedes Programm ist kommentiert.
 - ii) Jede Datei enthält im Kopf einen Kommentar. Als Autor des Programmes führen Sie alle Team-Mitglieder auf.
 - iii) Der Programmcode lässt sich übersetzen und zwar fehlerfrei und **ohne Warnings**. Prüfen sie das mit dem Befehl „Projektmappe neu erstellen“.
 - iv) Sie haben das Programm ausprobiert und auch einmal im Debugger Schritt-für-Schritt verfolgt?
 - v) Sie halten die Codierungsrichtlinien ein?

Einstimmung ins Thema

In diesem Praktikum soll geübt werden:

1. Umfang mit File I/O
2. Dynamische Speicherverwaltung und Umgang mit Listen
3. Suchen innerhalb von Strings

Aufgabe 3.1 – Filmverwaltung

Tom und Jerry lieben Filme und haben die Übersicht über die gesammelten DVDs und Blu-Rays verloren. Deswegen benötigen sie jetzt ein Verwaltungsprogramm und Sie sollen dieses Programm schreiben. Die beiden möchten bei jedem Film die folgenden Informationen gespeichert haben:

- Den Titel des Filmes,
- den Namen des Regisseurs
- das Erscheinungsdatum (Monat und Jahr) und
- die Länge in Minuten.
- die Art des Mediums (DVD, Blu-Ray und 3D-Blu-Rays).

```
... enum
{
    unbekannt=0,
    DVD,
    BluRay,
    BluRay3d
} TFilmTyp;

typedef struct
{
    char name[60];
    char regie[40];
    int laenge;
    char monat;
    short jahr;
    TFilmTyp typ;
} TFilm;
```

- Definieren Sie den oben angedeuteten enum für den Filmtyp und für den Film.
- Legen Sie ein ausreichend großes Array an, welches alle Filme aufnehmen kann. Die Verwendung von dynamisch allokiertem Speicher ist für diese Aufgabe möglich aber nicht vorgeschrieben, Sie können das lösen wie Sie wollen.
- Laden Sie alle Filme aus der bereitgestellten-Datei „FilmDatenbank.dat“.
- Fragen Sie den Benutzer danach, ob er noch weitere Filme eingeben möchte und fügen Sie diese ggf. dem Bestand hinzu. Ein Rückspeichern/Erweitern der Datei „FilmDatenbank.dat“ ist nicht gefordert
- Erstellen Sie eine Funktion zur Ausgabe des Filmbestandes auf dem Bildschirm.

Was möchten Sie tun?

- 1: Alle Filme ausgeben.
- 2: Einen weiteren Film eingeben.
- 3: Nach einem bestimmten Film suchen.

- (f) Schreiben Sie Funktionen, mit denen sie beiden nach Filmtiteln oder einer bestimmten Art des Mediums suchen können.

Diese Suchfunktion soll:

- Wenn es keinen Treffer gibt, eine passende Meldung ausgeben.
- Wenn es genau einen Treffer gibt, diesen Film mit Titel, Regie ... und all seinen Eigenschaften geordnet ausgeben.
- Wenn es mehrere Treffer gibt, die Liste der möglichen Filmtiteln durchnummeriert ausgeben, z.B. so:

```
Welchen Film möchten Sie sehen? Der
```

```
Die Auswahl "Der" ist nicht eindeutig, was möchten Sie sehen?
```

```
4: "Der Pate" von "Francis Ford Coppola"
```

```
6: "Der Schwarze Falke" von "John Ford"
```

```
13: "Die Passion der Jungfrau von Orléans" von "Carl Theodor Dreyer"
```

```
16: "Lichter der Großstadt" von "Charles Chaplin"
```

```
17: "Das süße Leben" von "Federico Fellini"
```

```
18: "Manche mögen's heiß" von "Billy Wilder"
```

```
23: "Der dritte Mann" von "Carol Reed"
```

```
24: "Moderne Zeiten" von "Charlie Chaplin"
```

```
25: "Dr. Seltsam oder: Wie ich lernte, die Bombe zu lieben" von  
"Stanley Kubrick",
```

```
28: "Das Lied der Straße" von "Federico Fellini"
```

```
31: "Der Mann, der Liberty Valance erschoss" von "John Ford"
```

```
32: "Der Stadtneurotiker" von "Woody Allen"
```

```
33: "Der weiße Hai" von "Steven Spielberg"
```

```
39: "Der letzte Kaiser" von "Bernardo Bertolucci"
```

- (g) Verwenden Sie keine globalen Variablen und übergeben Sie die notwendigen Arrays durch Pointer.

Aufgabe 3.2 To-Do-Liste

Wir alle ersticken ja ständig in Arbeit. Um den Überblick zu behalten, verwenden viele eine "to-do-list".

Erstellen Sie ein Programm, das es dem User ermöglicht, eine solche Liste zu verwalten. Zu einem

Eintrag in dieser Liste gehören mindestens die folgenden Informationen:

- Die Bezeichnung der Aufgabe, d.h. was muss gemacht werden,
- die Deadline, d.h. bis wann muss die Aufgabe spätestens erledigt sein,
- wo muss die Aufgabe erledigt werden

- Schreiben Sie Funktionen, mit denen der User neue Aufgaben in die Liste eintragen kann. Verwenden Sie zur Speicherung auf jeden Fall eine Listenstruktur (einfach verkettete Liste).
- Schreiben Sie Funktionen, um die eingegebenen Information abzuspeichern und beim folgenden Programmstart wieder von der Festplatte einzulesen.
- Mit einer gut organisierten Todo-Liste schafft man was. Ermöglichen Sie also dem User, Aufgaben zu entfernen, weil sie erledigt sind oder sich anderweitig erledigt haben.

```
Sie haben aktuell 123 Einträge in Ihrer TO-DO Liste.  
Was möchten Sie tun?
```

```
1: Alle TO-DOs ausgeben und zwar die mit der nächsten Deadline  
zuerst.
```

```
2: Weiteres TO-DO eingeben.
```

```
3: Bestehendes TO-DO aus der Liste löschen.
```

```
0: aktuelle TO-DO Liste in Datei "MeineToDoListe.dat" speichern und  
dann Programm beenden.
```

Tipp 1: In dieser Aufgabe geht es nicht darum, ein möglichst effizientes Sortierverfahren zu implementieren, Sie können ein beliebiges nehmen.

Versuchen Sie zunächst, selbst ein Verfahren zu schreiben. Bleiben Sie dran, beim ersten Mal geht es naturgemäß etwas schwieriger.

Wenn es gar nicht geht, können Sie die beiliegende Programmskizze als Anhaltspunkt verwenden.

Widerstehen Sie der Versuchung so lange Sie können!