

Eidl Praktikum 5: Zeichenketten

19. Dezember 2023

Hinweise

- Ihr Programm vom letzten Blatt soll erweitert werden. Das bedeutet, dass alle bisherigen Anwendungsfälle, Anforderungen und Eigenschaften erhalten bleiben und weiterhin funktionieren sollen. Testen Sie daher ihr gesamtes Programm bevor Sie in das Abtestat gehen.
- (**Informatiker**) Denken Sie auch an fehlerhafte Eingaben und typische Probleme beim Umgang mit Dateien. Ihr Programm darf nicht abstürzen.
- Achten Sie bei der Benennung von Funktionen, Parametern und Variablen auf aussagekräftige Namen.
- Ihre Funktionen sollen mit doxygen dokumentiert sein.
- Sie können Ihren Code bei Bedarf auf mehrere Dateien aufteilen und auf geeignete Weise einbinden.
- Sie demonstrieren Ihre Lösungen der Aufgaben. Bereiten Sie dafür geeignete Testfälle vor, die die Funktionalität Ihres Programms aufzeigen. Sie sollten außerdem in der Lage sein, Ihren Code zu verändern, zu compilieren und zu erklären.
- Achten Sie darauf, dass die Vorstellung der Aufgaben gleichmäßig innerhalb Ihrer Gruppe verteilt ist.
- Buch: „C von A bis Z“ Kapitel 11.13 & 21

Aufgabe 1: Zeichenketten

Themen/Kompetenzen

- Analysieren und bewerten
- Speichermanagement

a) Für das Kopieren von Zeichenketten können die Funktionen `char* strcpy(char *dest, const char *source)` oder `char* strncpy(char *dest, const char *source, size_t num)` verwendet werden.

Warum gilt `strncpy` als *sicher* und sollte bevorzugt verwendet werden? Erklären Sie anhand dieses Beispiels typische Problemstellungen beim Umgang mit Zeichenketten und die Bedeutung des Begriffs *sicher* in diesem Zusammenhang.

b) (Informatiker) Erstellen Sie ein Programm, dass die „Unsicherheit“ von `strcpy` demonstriert.

Aufgabe 2: verkettete Liste

Themen/Kompetenzen

- Datenstrukturen
- strukturierte Datentypen
- Modellieren

a) Modellieren Sie einen strukturierten Datentyp für eine Person, der mindestens die Felder Vor- und Nachname besitzt.

b) Implementieren Sie eine *einfach* oder *doppelt verkettete Liste*, die als Datentyp eine Person aufnehmen kann. Stellen Sie die Datenstruktur in einer separaten c-Datei zur Verfügung, die über einen geeigneten Header eingebunden werden kann.

Hinweis: Orientieren Sie sich an den Beispielen aus der Vorlesung (G3-01 Liste & G3-02 Doppelliste).

c) Erweitern Sie Ihre Studienplan-App um die *Anwendungsfälle*:

7. Hinzufügen beliebig vieler Personen zu einem Termin. Verwenden Sie Ihre Implementation der Liste aus 2b.
8. Ausgabe aller Termine einer vom Anwender ausgewählten Person.
9. (**Informatiker**) Entfernen einer Person aus einem Termin.

Anforderungen:

- Die Namen der Personen sollen *unique* (=einzigartig) innerhalb eines Termins sein, d.h. es darf keine zwei Personen mit demselben Vor- und Nachnamen in einem Termin geben.
- Bei der Ausgabe aller Termine (Anwendungsfall 3, Praktikum 4) sollen die Namen der Personen angezeigt werden.
- (**Informatiker**) Die Personen eines Termins sollen bei der Ausgabe (vorherige Anforderung) alphabetisch nach dem Nachnamen sortiert sein.

Hinweis

- Sie dürfen Ihr Hauptmenü beliebig erweitern, zum Beispiel um den Punkt „Termin bearbeiten“.

d)(Informatiker) Zeigen Sie, dass Ihre Implementation *robust* ist, indem Sie ungültige Funktionen in Ihrer Demonstration aufrufen (z.B. Löschen eines nicht vorhandenen Elements). Ein *robustes* Programm stürzt in diesem Fall nicht ab und lässt sich weiter benutzen.

Aufgabe 3: Arbeiten mit Dateien

Themen/Kompetenzen

- Serialisierung
- Dateizugriff
- Datenströme

Erweitern Sie Ihre Stundenplan-App um folgende Anwendungsfälle:

10. (**Ingenieure**) Exportieren der Termine ohne Personen in eine Textdatei.
11. (**Informatiker**) Exportieren der Termine inkl. Personen in eine Textdatei.

Hinweis: Das Format des Dateiinhalts darf frei gewählt werden, muss aber *men-
schenlesbar* sein.