

Hardwarenahe Programmierung

Gruppe 6 (Stefan)

In dieser Übung arbeiten Sie mit komplexen Datentypen in C-Programmen, die über structs realisiert werden.

Sie werden mit einigen Ihrer Lösungen in den nächsten Tagen weiterarbeiten! Sie sollten diese also aufbewahren und so schreiben, dass Sie damit gut weiterarbeiten können.

Aufgabe 1 Datenstrukturen

- (a) Erstellen Sie eine Headerdatei, in welcher Sie eine Datenstruktur (**struct**) definieren, die ein Datum (z.B. 10.02.2020) speichern kann. Nutzen Sie Ihre Datenstruktur, um einen Datentypen mit Namen **date** zu definieren.
- (b) Fügen Sie Ihrer Headerdatei einen Prototypen für eine Funktion hinzu, die ein Datum des Typs **date** als Eingabe erhalten und im Format **tt.mm.jjjj** auf der Konsole ausgeben soll. Implementieren Sie diese Funktion in einer passenden zweiten Datei.
- (c) Ergänzen Sie Ihre Dateien geeignet um eine Funktion **int is_before(date a, date b)**, die folgendes zurückgibt:
 - 0, wenn beide eingegebenen Daten gleich sind;
 - -1, wenn **a** zeitlich vor **b** liegt;
 - 1, wenn **a** zeitlich nach **b** liegt.

Testen Sie Ihre Funktion mit den vorgegebenen Unit-Tests! Ergänzen Sie dazu die markierten Stellen, um Ihre Headerdatei und Ihren Datentyp passend einzubinden.

- (d) Schreiben Sie eine main-Funktion (in einer separaten Datei *main.c*), welche die Verwendung Ihres Datentyps und beider Funktionen demonstriert.

Hinweis: Sie brauchen in der gesamten Aufgabe keine semantisch falschen Daten abzufangen (wie zum Beispiel 40.13.2020).

Aufgabe 2 *Tier-Daten*

Ein Tierhalter möchte zu seinen Tieren folgende Daten speichern: Rasse, Name und Geburtsdatum und Art des Tiers.

- Rasse,
- Name,
- Geburtsdatum (Tag Monat Jahr),
- und Tiersorte (Hund oder Katze).

Wir haben Ihnen in *tiere.c* einen Prototypen zur Verfügung gestellt, in welchem eine Funktion `read_tiere` existiert, die ein übergebenes Array mit Daten von der Standardeingabe füllt. Diese Daten können auch von einer Datei aus in das Programm umgeleitet werden. Beispieldaten dazu finden Sie in *hunde_und_katzen.txt*, diese sehen z.B. folgendermaßen aus:

```
Collie Lassie 17 12 1938 Hund
Hauskatze Simon 4 4 2008 Katze
```

Die Funktion `print_tiere` gibt Tierdaten in menschenleslicher Form auf der Konsole aus, eine Beispiel-Ausgabe finden Sie in *ausgabe_tiere.txt*.

- Implementieren Sie einen Datentypen `tier`, der die obigen Daten speichern kann. Verwenden Sie für das Datum Ihre Implementierung aus der vorherigen Aufgabe. Legen Sie für die Speicherung der Tiersorte einen geeigneten `enum`-Typen an. Implementieren Sie die Funktionen `print_tierart` und `tierart_einlesen`, die für das Einlesen und Ausgeben benötigt werden, passend dazu.
- Schreiben Sie eine `main`-Funktion, in der Sie ein Array anlegen, in welchem 10 Tiere gespeichert werden können. Füllen Sie das Array mittels `read_tiere` und geben Sie den Inhalt mit `print_tiere` aus.
- Schreiben Sie eine Funktion `aeltestes_tier`, die eine Liste von Tierdaten übergeben bekommt und das älteste Tier davon auf der Konsole ausgibt. Fügen Sie einen Aufruf der Funktion `aeltestes_tier` in Ihre `main`-Funktion ein und demonstrieren Sie die Funktion damit.

Aufgabe 3 *Brüche*

Ergänzen Sie die gegebenen Dateien *bruch.h* und *bruch.c* so, dass sie die gegebenen Unittests erfüllen und *main.c* funktioniert. Die Dateien sollen:

- eine Datenstruktur `bruch` zur Speicherung eines Bruchs enthalten,
- eine Funktion `product` enthalten, die zwei Brüche multipliziert und das Produkt zurückgibt,
- eine Funktion `kuerze` enthalten, die einen Bruch soweit wie möglich kürzt,
- und eine Funktion `print` enthalten, die einen Bruch als Gleitkommazahl auf der Konsole ausgibt.

Aufgabe 4 *Platzsparende Datenstrukturen*

Gesucht ist eine Datenstruktur, in der Antworten eines Feedbackfragebogens zu einer Übung gespeichert werden können. Implementieren Sie diese Datenstruktur und verwenden Sie dabei Enums und Bitfelder, um möglichst wenig Speicher zu verwenden. *Bonus*: Verwenden Sie zusätzlich dynamische Speicherverwaltung, um möglichst wenig Platz für die Speicherung von Strings zu belegen.

Der Fragebogen beinhaltet die folgenden Fragen (Antwortmöglichkeiten in Klammern):

1. “Welches Fach studieren Sie?” (Informatik/Mathematik/Physik)
2. “In welchem Fachsemester studieren Sie?” (1-12)
3. “Bitte benoten Sie Ihre Übung in Schulnoten.” (1-6)
4. “Haben Sie weitere Anmerkungen?”
(Freitext, maximal 120 Zeichen)

Betten Sie Ihre Datenstruktur in ein Programm ein, welches einen solchen Fragebogen vom Benutzer abfragt, die Antworten speichert und wieder auf der Konsole ausgibt. Wir haben Ihnen einen Prototypen bereitgestellt, welcher die Eingabe schon vorbereitet.