

Programmierung 2 - Sommersemester 2023

Prof. Dr.-Ing. Martin Burger

Übungsblatt Nr. 22 Abgabe KW 28

Allgemeiner Hinweis: Implementieren Sie für alle Ihre Lösungen einen Dialog-Test, auch wenn dies bei einzelnen Aufgaben nicht explizit in der Aufgabenstellung genannt ist. Der Test sollte so gestaltet sein, dass die Funktionsweise Ihres Programms bei der Abnahme gut nachvollzogen werden kann.

1. Aufgabe

- (a) Implementieren Sie zwei Funktions-Bibliotheken zur Berechnung einfacher physikalischer Formeln. Die Bibliotheken sollen auf einfache Art und Weise wiederverwendbar sein. Teilen Sie dazu Deklaration und Implementierung in unterschiedliche Dateien auf. Definieren Sie Ihre eigenen Typen für die verwendeten physikalischen Größen. Zur Wiederverwendbarkeit sollen diese in einer eigenen Header-Datei definiert sein.

Die erste Funktions-Bibliothek soll Funktionen zur Berechnung folgender mechanischer Größen besitzen:

- i Geschwindigkeit: $v := \frac{\Delta s}{\Delta t}$
- ii Beschleunigung: $a := \frac{\Delta v}{\Delta t}$
- iii Impuls: $p := m \cdot v$

Die zweite Bibliothek soll Funktionen zur Berechnung geradliniger Bewegungen beinhalten:

- i Weg einer gleichförmigen geradlinigen Bewegung: $s = v \cdot (t - t_0)$
- ii Weg einer gleichmäßig beschleunigten geradlinigen Bewegung: $s = \frac{v^2 - v_0^2}{2a}$
- iii Geschwindigkeit einer gleichmäßig beschleunigten geradlinigen Bewegung:
 $v = a \cdot (t - t_0) + v_0$

Mit: v := Geschwindigkeit, a := Beschleunigung, p := Impuls, s := Weg, t := Zeit, m = Masse

- (b) Implementieren Sie ein interaktives Testprogramm, welches Ihre Bibliotheken einbindet und testet.

2. Aufgabe

Es soll ein Programm zur Verwaltung des Autobestands in einem Autohaus entwickelt werden. Implementieren Sie dazu folgendes:

- (a) Definieren Sie einen Typ Auto mit folgenden Eigenschaften:
- i Marke (der Markenname ist auf 20 Zeichen beschränkt)
 - ii Höchstgeschwindigkeit
 - iii Türenanzahl
 - iv ABS (ja oder nein)

- v Sonderausstattung (eine Liste von Ausstattungsmerkmalen als Strings; ein Auto hat maximal 10 Ausstattungsmerkmale, jedes Ausstattungsmerkmal hat maximal 100 Zeichen)
 - vi Motor Ein Motor soll durch einen eigenen Typ mit folgenden Eigenschaften dargestellt werden:
 - PS
 - Zylinder
 - Hubraum
- (b) Erstellen Sie eine Funktion, welche die Eigenschaften eines Autos als Parameter entgegennimmt und ein Auto zurückliefert.
- (c) Schreiben Sie eine Funktion, welche den Wert eines Autos folgendermaßen schätzt:
- Maximalgeschwindigkeit multipliziert mit 50
 - multipliziert mit der Türenanzahl
 - plus 5000 EUR, wenn das Auto ABS hat
 - plus 3000 EUR für jede Sonderausstattung
 - ist die Marke Porsche, multipliziert mit 2
- (d) Implementieren Sie eine Print-Funktion, welche ein Auto mit allen seinen Eigenschaften ausgibt.
- (e) Verwalten Sie die Menge der Autos eines Autohauses in einem Array. Implementieren Sie dazu Funktionen, um Autos hinzuzufügen und zu entfernen.
- (f) Implementieren Sie ein interaktives Testprogramm.

3. Aufgabe (Zusatzaufgabe)

Implementieren Sie ein C-Programm welches die folgende Ausgabe erzeugt:

```
*-----*
-*-----*
--*-----*
---*-----*
---*-----*
--*-----*
-*-----*
*-----*
```

Lesen Sie die Anzahl der Zeilen und Spalten vom Nutzer ein.

4. Aufgabe (Zusatzaufgabe)

Ermitteln Sie den Speicherbedarf eines `Auto`-Structs aus Aufgabe 2 und ermitteln Sie die Speicheradressen sowie den Offset der einzelnen Komponenten (inkl. der Komponenten des `Motor`-Structs). Was stellen Sie fest und wie begründen Sie Ihre Feststellung?

5. Aufgabe (Zusatzaufgabe)

Implementieren Sie eine Bibliothek zur Verarbeitung von Strings. Die Bibliothek soll folgende Funktionen beinhalten:

- `strcmp`: Nimmt zwei Strings entgegen und vergleicht diese. Liefert `<0` wenn der erste String kleiner ist, `0` wenn beide Strings gleich sind und sonst `>0`.

- `strspn`: Nimmt zwei Strings entgegen und liefert die Anzahl der Zeichen am Anfang des ersten Strings, die sämtlich im zweiten String vorkommen.
- `strlen`: Liefert die Länge eines übergebenen Strings, ohne `'\0'`.
- `indexOf`: Nimmt einen String und ein einzelnes Zeichen entgegen und liefert den Index des ersten Vorkommens des Zeichens. Falls das Zeichen nicht vorkommt, wird -1 zurückgegeben.
- `lastIndexOf`: Nimmt einen String und ein einzelnes Zeichen entgegen und liefert den Index des letzten Vorkommens des Zeichens. Falls das Zeichen nicht vorkommt, wird -1 zurückgegeben.
- `replace`: Nimmt einen String und zwei einzelne Zeichen entgegen. Ersetzt alle Vorkommen des ersten Zeichens durch das Zweite.
- `toUpperCase`: Nimmt einen String entgegen und wandelt alle Buchstaben in Großbuchstaben um.
- `swap`: Nimmt zwei Strings entgegen und vertauscht diese.

Implementieren Sie auch ein interaktives Testprogramm für Ihre String-Bibliothek.