

### 5.3.A Datenstrukturen II/Arrays von Records – Arbeitsblatt

#### Aufgabe 1: Regalbretter-Tabelle

Für das Holzlager soll nun eine Tabelle aller fertig zugeschnittenen Bretter angelegt und (später) verwaltet werden (`regalbrett3.c`).

- a) Erstellen Sie das Jackson-Diagramm für die Tabelle!
- b) Erstellen Sie die passende Variable `holztab` im Hauptprogramm! Sie soll maximal 50 Bretter aufnehmen können.
- c) Bestimmte Bretter sollen standardmäßig vorhanden sein. Sorgen Sie dafür, dass die Variable `holztab` zum Programmstart wie folgt initialisiert wird:
  - 1)  $800 \times 200 \times 19$ , Kiefer
  - 2)  $600 \times 300 \times 21$ , Fichte
  - 3)  $1200 \times 250 \times 24$ , Spanplatte
  - 4) weitere Einträge leer
- d) Geben Sie *die ersten drei* Einträge der Tabelle aus!

#### Aufgabe 2: Film-Tabelle

Für den Film-Katalog soll ebenfalls eine Tabelle der vorhandenen Filme angelegt und (später) verwaltet werden (`filmkatalog3.c`).

- a) Erstellen Sie das Jackson-Diagramm für die Tabelle!
- b) Erstellen Sie die passende Variable `filmtab` im Hauptprogramm! Sie soll maximal 2000 Filme aufnehmen können.
- c) Bestimmte Filmtitel sollen als Testdaten bereits mithilfe der Initialisierung zum Programmstart in die Filmtabelle geschrieben werden:
  - 1) Der letzte Mann, 1924, 86 min
  - 2) Metropolis, 1927, 117 min
  - 3) Avatar: Der Weg des Wassers, 2022, 193 min
  - 4) alle weiteren Einträge leer
- d) Geben Sie *alle nicht leeren* Einträge der Tabelle aus!

#### Aufgabe 3: Tabelle mit Zufallswerten füllen

Zu Testzwecken soll die Buchtabelle (`buchtyp4.c`) mit Zufallswerten vollständig gefüllt werden. Die Zufallsergebnisse sollen dann angezeigt werden.

- a) Der Autor soll einen von zwanzig zufälligen Vornamen und einen von zwanzig zufälligen Nachnamen haben.
- b) Auch der Titel soll aus zwei zufällig gewählten Teilen bestehen, von denen der erste ein Subjekt (z. B. *Die Langeweile* oder *Im kalten Schwimmbad*) und das zweite ein Attribut (z. B. *des Grauens* oder *für alle*) ist.
- c) Die Auflage soll zwischen 1 und 11 liegen.
- d) Geben Sie alle Einträge der Tabelle aus!

## Aufgabe 4: Bußgeldrechner

```

1 #include <stdio.h>
2 #include <limits.h>
3 struct bussgeldstufe_t
4 {
5     int innerorts;      /* 1=innerorts, 0=ausserorts, -1=Ende */
6     int kmhzuviel;     /* maximale Ueberschreitung um x km/h */
7     int bussgeld_eur;  /* das kostet es */
8     int punkte;        /* so viele Punkte in Flensburg gibt es */
9     int fahrverbot_mon; /* und so lange darf man nicht fahren */
10 };
11 int teststufe(int kmhzuviel, int innerorts, struct bussgeldstufe_t bgs);
12 int main(void)
13 {
14     struct bussgeldstufe_t bg[]={
15         {1, 0, 0,0,0}, {1,10, 15,0,0}, {1,15, 25,0,0}, {1,20, 35,0,0},
16         {1,25, 80,1,0}, {1,30,100,3,0}, {1,40,160,3,1}, {1,50,200,4,1},
17         {1,60,280,4,2}, {1,70,480,4,3}, {1,INT_MAX,680,4,3},
18         {0, 0, 0,0,0}, {0,10, 10,0,0}, {0,15, 20,0,0}, {0,20, 30,0,0},
19         {0,25, 70,1,0}, {0,30, 80,3,0}, {0,40,120,3,0}, {0,50,160,3,1},
20         {0,60,240,4,1}, {0,70,440,4,2}, {0,INT_MAX,600,4,3}, {-1,0,0,0,0}};
21     int geschw_erlaubt, geschw_real;
22     int geschlossene_ortschaft;
23     int ende, lauf;
24     do{
25         printf("Bussgeldrechner\n"
26             "Wie_schnell_sind_Sie_gefahren_(in_km/h)"
27             " .....: ");
28         scanf("%i", &geschw_real); while(getchar()!='\n'){
29             printf("Wie_schnell_haetten_Sie_eigentlich_nur"
30                 "_fahren_duerfen_(in_km/h)...: ");
31             scanf("%i", &geschw_erlaubt); while(getchar()!='\n'){
32                 printf("Befanden_Sie_sich_in_einer_geschlossenen"
33                     "_Ortschaft_(0=nein, _1=ja): ");
34                 scanf("%i", &geschlossene_ortschaft); while(getchar()!='\n'){
35                     for(lauf=0; bg[lauf].innerorts!=-1; ++lauf)
36                     {
37                         if(teststufe(geschw_real-geschw_erlaubt,
38                             geschlossene_ortschaft, bg[lauf]))
39                         {
40                             printf("Bussgeld ...: %d_EUR\n", bg[lauf].bussgeld_eur);
41                             printf("Punkte ....: %d\n",      bg[lauf].punkte);
42                             printf("Fahrverbot: %d\n",      bg[lauf].fahrverbot_mon);
43                             break;
44                         }
45                     }
46                 printf("Programm_beenden_(0=nein, _1=ja)"
47                     " .....: ");
48                 scanf("%i", &ende); while(getchar()!='\n'){
49                     }while(ende==0);
50                 return 0;
51             }

```

Das oben stehende Programm stellt einen Entwurf für einen Bußgeldrechner dar. Nach Abfrage von gefahrener und erlaubter Geschwindigkeit sowie der Tatsache, ob der Vorfall in einer geschlossenen Ortschaft stattfand, werden Bussgeld, Punktzahl und Dauer eines Fahrverbots ausgegeben (Stand: 2014!<sup>1</sup>).

- a) Erstellen Sie zum Hauptprogramm das zugehörige Struktogramm!
- b) In jedem Record des Typs `struct bussgeldstufe_t` werden die Daten für eine Gebührenstufe gespeichert: Unterscheidung innerorts/außerorts, maximale Geschwindigkeitsüberschreitung, Höhe von Bußgeld, Punkteintrag und Fahrverbot. Stellen Sie dieses Record in einem Jackson-Diagramm dar!
- c) In der Variablen `bg` sollen sämtliche Gebührenstufen gespeichert werden. Stellen Sie die Variable in einem Jackson-Diagramm dar!
- d) In der Funktion `teststufe()` soll festgestellt werden, ob ein Vorfall zu einer bestimmten Gebührenstufe passt. Falls ja, soll eine Eins zurückgegeben werden, sonst eine Null. Erstellen Sie zu dieser Funktion ein Struktogramm; schreiben Sie die Funktion in C!

---

<sup>1</sup>Quelle: <http://www.bmvi.de/SharedDocs/DE/Artikel/LA/bussgeldkatalog.html>, abgerufen am 7. April 2014