Datentypen + Printf

Dienstag, 25. Oktober 2022 15:20

Gegeben seien folgende Datentypbeschreibungen (angelehnt an das JSON Schema)

```
{
  "title": "Student",
  "type": "object",
  "properties":
  {
    "Vorname": {
      "type": "string_10"
    },
    "Nachname": {
      "type": "string_10"
    },
    "Nationalität": {
      "type": "Aufzählung[
            deutsch|
            england|
            spanien|
            sonstiges]"
    },
    "Semester": {
      "type": "short"
    },
    "Fächer":
    {
      "type": "array[2]",
      "items":
       {
        "type": "object",
        "properties":
        {
          "Name": {
            "type": "string_10"
          },
          "Note": {
            "type": "FloatZahl"
          },
        },
      }
    },
  }
}
```

```
{
  "title": "Dozent",
  "type": "object",
  "properties":
  {
    "Titel": {
      "type": "string_10"
    },
    "Name": {
      "type": "string_10"
    },
    "Nationalität": {
      "type": "Aufzählung[
            deutsch|
            england|
            spanien|
            sonstiges]"
    },
    "Einstiegsjahr": {
      "type": "short"
    },
    "Fächer":
    {
      "type": "array[2]",
      "items":
       {
        "type": "object",
        "properties":
        {
          "Name": {
            "type": "string_10"
          },
          "Note": {
            "type": "FloatZahl"
          },
        },
      }
    },
  }
}
```

```
{
  "title": "Hochschule",
  "type": "object",
  "properties":
  {
    "Name": {
      "type": "string_10"
    },
    "Ort": {
      "type": "string_10"
    },
    "PLZ": {
      "type": "Ganzzahl"
    },
    "Strasse": {
      "type": "sring_10"
    },
    "Studenten": {
      "type": "long"
    },
  }
}
```

1) Realisieren sie für jedes dieser Datentypbeschreibungen einen C-Datentyp
   struct student {….};
   struct dozent {….};
   struct hochschule {….};

2) Realisieren sie einen weiteren Datentypen, welcher alle diese 3 Datentypen aufnehmen kann, wobei nur einer von diesen Aktiv sein kann (Stichwort Union). Zur Identifikation, welcher Datentyp in der Union gültig/aktiv ist, packen sie das Ganze in einer übergeordnete Struktur bestehend aus einem Aufzählungstyp zur Beschreibung des gültigen Eintrages und der Union
   struct element {
      //enum zu Identifikation des gültigen Unions
      //Union zur Aufnahme der 3 zuvor definierten Datentypen
      };

3) Mit dem Datentyp Specifier können einzelnen Strukturelementen zusätzliche Eigenschaften/Einschränkungen gegeben werden. Überlegen sie sich für jeden Specifier mindestens einen sinnvolles Strukturelement (sofern überhaupt möglich) und Begründen sie dessen Verwendung. Wenn es keine Verwendung für einen Specifier gibt, so bitte auch dieses Begründen
- static
- extern
- volatile
- const

4) Legen sie vom Datentyp 'struct element' ein Array mit 9 Elemente an und initialisieren sie dieses Array mit Musterdaten (je 3 von jedem Subdatentyp:
   struct element elemente[9]={….};

5) Schreiben sie eine Debug-Routine, welche das gesamte Array tabellarisch ausgibt. Die Überschrift der Tabelle lautet:

```
--------+----------+------------+----------+------------+------------+------------+------------
Vorname | Nachname | Nationali. | Semester | Fach0-Name | Fach0-Note | Fach1-Name | Fach1-Note
Titel   | Name     | Nationali. | EinsJahr | Fach0-Name | Fach0-Note | Fach1-Name | Fach1-Note
-       | Ort      | -          | Plz      | Straße     | Studenten  | -          | -
--------+----------+------------+----------+------------+------------+------------+------------
```

Hinweis:
- Alle Spalten sollen gleich breit sein, dürfen jedoch von dem Beispiel eine abweichende Breite haben
- Nutzen sie das '|'-Zeichen zum Trennen der Spalten

Hinweis:
- Zur Abgabe bitte folgendes Rahmenprogramm nutzen

```c
#include <stdio.h>
//CompilerSchalter: -fsanitize=address -Wall -Werror
//1) Realisieren sie für jedes dieser Datentypbeschreibungen einen C-Datentyp
struct student {
};
struct dozent  {
};
struct hochschule {
};
//2) Realisieren sie einen weiteren Datentypen, welcher alle diese 3 Datentypen
//   aufnehmen kann, wobei nur einer von diesen Aktiv sein kann (Stichwort Union).
struct element {
        //enum zu Identifikation des gültigen Unions
        //Union zur Aufnahme der 3 zuvor definierten Datentypen
};

//3) Mit dem Datentyp Specifier können einzelnen Strukturelementen zusätzliche
//   Eigenschaften/Einschränkungen gegeben werden. Überlegen sie sich für jeden
//   Specifier mindestens einen sinnvolles Strukturelement (sofern überhaupt möglich)
//   und Begründen sie dessen Verwendung. Wenn es keine Verwendung für einen Specifier
//   gibt, so bitte auch dieses Begründen
#if 0
  - static:  mögliches Strukturelement + Begründung
 - extern:   mögliches Strukturelement + Begründung
 - volatile: mögliches Strukturelement + Begründung
 - const:    mögliches Strukturelement + Begründung
#endif
//4) Legen sie vom Datentyp 'struct element' ein Array mit 9 Elemente an und
//   initialisieren sie dieses Array mit Musterdaten (je 3 von jedem Subdatentyp:
struct element elemente[9]={
/*[0]=*/{    },
/*[1]=*/{    },
/*[2]=*/{    },
/*[3]=*/{    },
/*[4]=*/{    },
/*[5]=*/{    },
/*[6]=*/{    },
/*[7]=*/{    },
/*[8]=*/{    },
};

//5) Schreiben sie eine Debug-Routine, welche das gesamte Array tabellarisch ausgibt.
void debug_element(void) {
        printf(
"--------+----------+------------+----------+------------+------------+------------+------------\n"
"Vorname | Nachname | Nationali. | Semester | Fach0-Name | Fach0-Note | Fach1-Name | Fach1-Note\n"
"Titel   | Name     | Nationali. | EinsJahr | Fach0-Name | Fach0-Note | Fach1-Name | Fach1-Note\n"
"-       | Ort      | -          | Plz      | Straße     | Studenten  | -          | -\n"
"--------+----------+------------+----------+------------+------------+------------+------------\n"
              );
    //Hinweis:
    //- Spaltenbreite der Überschrift darf angepasst werden!
    //- Die Spalten sollen im Ausdruck alle gleich breit sein
}
int main(int argc, char *argv[])
{
    (void) argc;
    (void) argv;
    debug_element();
    return 0;
}
```