

6.7.A Extras/Typdefinitionen mit typedef – Arbeitsblatt

Aufgabe 1: typedef für elementare Datentypen, Zeiger, Arrays und Records

Erstellen Sie mit typedef die geeigneten Datentypen!

- uint16_t soll ein Datentyp für 16 Bit lange vorzeichenlose Ganzzahlen sein; auf PCs ist es meistens gleichbedeutend mit unsigned short.
- l_double_ptr_t soll ein Datentyp sein für einen Zeiger auf long double.
- screen_bitmap_t soll ein Typ für zweidimensionales Array (600 Zeilen, 800 Spalten, Elemente vom Typ int) sein.
- persontyp soll ein neuer Name für den bereits bestehenden Typ struct person_t sein.

Aufgabe 2: typedef für Prototypen

Erstellen Sie mit typedef die geeigneten Datentypen für Prototypen!

- c_type_funktion_t soll ein Typ für die Prototypen der Bibliotheksfunktionen sein, die in c_type.h deklariert werden (wie z.B. tolower()).

Aufgabe 3: typedef für Funktionszeiger

Erstellen Sie mit typedef die geeigneten Datentypen für Funktionszeiger!

- atexit_handler_t soll ein Typ für den Funktionszeiger sein, der als Parameter der Bibliotheksfunktion atexit() dient, so dass atexit() auch so deklariert werden kann:

```
int atexit(atexit_handler_t);
```

Aufgabe 4: Suche nach typedef

In den Header-Dateien der C-Standardbibliothek tauchen Typdefinitionen mit typedef wiederholt auf.

- In welchem Verzeichnis liegen in Ihrem System die genannten Headerdateien? — Tip: Suchen Sie mal nach stdio.h!
- Welche Typen sind durch typedef definiert, welche durch #define, welche mit Hilfe von struct?