

Aufgabe 3

Präsentieren Sie 5 wichtige, lehrreiche Hinweise.

1. Man sollte die Struktur hinter der API verstecken. Das heißt Objekte sollten über Getter und Setter steuerbar sein, jedoch nicht direkt über ihre Variablen. Diese sollten also als private ausgezeichnet sein.
2. Wenn es de-facto standardisierte Namen für Funktionen gibt, die allgemein bekannt sind, sollte man diese auch verwenden. Das heißt, nicht das Rad neu erfinden, sondern bekannte Namen für Funktionen, aber auch Variablen usw. verwenden.
3. Man sollte wichtige Dinge, die sich z.B. in einer neuen Version einer Library, eines Programms, usw. geändert haben, auch dokumentieren und veröffentlichen. Also hinzugefügte/entfernte Funktionalität, veränderte API usw. Dadurch wird gewährleistet dass man als Entwickler vorher überprüfen kann ob das eigene Programm mit einer neuen Version einer verwendeten Library noch kompatibel ist.
4. Immer mit `#ifndef`, `#define` und `#endif` überprüfen ob eine Headerdatei bereits von einem anderen Teil eines Programms eingefügt wurde.
5. Man sollte außerdem immer Error-Codes zur Verfügung stellen (zurückgeben) und nicht einfach die Abarbeitung z.B. mit `exit()` oder `abort()` abbrechen. Programme, die eine Library verwenden sollen also nicht einfach „abgeschossen“ werden, sondern ihnen soll ein Fehlercode übermittelt werden, mit dem sie weitere Schritte einleiten können.

Was halten Sie davon?

Ist der Artikel für die eigene Arbeit relevant?

Als zukünftiger Web-Entwickler sehe ich mich zwar nicht wirklich in der Situation wieder, dass ich eines Tages eine C-Library schreiben werde. Ich denke aber, dass allgemein gültige Dinge (wie z.B. standardisierte Namen für Funktionen) hier verständlich erklärt werden um sich einen Überblick zu verschaffen. Will man jedoch detailliertere Informationen, muss man sich weitere Quellen suchen.

Welche Punkte sind Ihnen unklar?

- autotools
- GPL, LGPL
- Zero global state
- „hidden“ `fork()/exec()`
- symbol versioning