Aufgabe 4h: Modulare Programmierung

1 Lernziele

Anwenden der modularen Programmierung sowie Vertiefen der Kenntnisse über die Gültigkeit und Sichtbarkeit von Variablen. Sie können erste Erfahrungen sammeln im gemeinsamen Erstellen von Software: Man einigt sich auf eine Schnittstelle und jeder Beteiligte löst unabhängig vom anderen einen Teil der Aufgabe.

Die Aufteilung, wer aus der Zweiergruppe welches Modul implementiert, bleibt Ihnen überlassen. Am Ende des Praktikums müssen beide Module zusammen kompilierbar sein und das Programm die gewünschte Funktionalität aufweisen.

2 Aufgabe

Erstellen Sie ein Programm, das eine Zuordnung von Bauteilen auf Schaltungen vornimmt: Jedes Bauteil ist in verschiedenen Schaltungen eingebaut, jede Schaltung besteht aus verschiedenen Bauteilen.

Teil 1: Schnittstelle `component`

```
component_t *createComponent(char *id, char *description);
void getID(component_t *comp, char *id, int size);
void addCircuit(component_t *comp, circuit_t **circuit);
void getComponentData(component_t *comp, char *data, int size);
void getCircuitsToComp(component_t *comp, char *circuits, int size);
void removeComponent(component_t *comp);
```

Schreiben Sie ein Modul `component.c`, das die Schnittstelle `component` implementiert. Stellen Sie sicher, das auf die Komponenten der Struktur nur innerhalb des Moduls zugegriffen werden kann. Überlegen Sie sich, welche Komponenten die Struktur `component_t` haben muss und definieren Sie mittels `typedef` einen Datentyp.

Teil 2: Schnittstelle `circuit`

```
circuit_t *createCircuit(char *id, char *creator, char *description);
void getID(circuit_t *circuit, char *id, int size);
void addComponent(circuit_t *circuit, component_t **comp);
void getCircuitData(circuit_t *circuit, char *data, int size);
void getCompsOfCircuit(circuit_t *circuit, char *comps, int size);
void removeCircuit(circuit_t *circuit);
```

Schreiben Sie ein Modul `circuit.c`, das die Schnittstelle `circuit` implementiert. Stellen Sie sicher, das auf die Komponenten der Struktur nur innerhalb des Moduls zugegriffen werden kann. Überlegen Sie sich, welche Komponenten die Struktur `circuit` haben muss und definieren Sie mittels `typedef` einen Datentyp.

Das Modul `main.c`, das eine Benutzeroberfläche bereitstellt, erhalten Sie von uns, es muss also nicht von Ihnen erstellt werden.

Zu einem vollständigen Programm gehört eine Fehlerbehandlung!

3 Testat

Voraussetzung ist jeweils ein fehlerfreies, korrekt formatiertes Programm. Der korrekte Programmlauf muss anhand einer Beispieleingabe nachgewiesen werden. Sie müssen in der Lage sein, Ihr Programm zu erklären.