

Übungsblatt 3

Aufgabe 1: Konstruktorfunktionen

Erstellen Sie einen Prototypen `Container`. Ein Container aggregiert verschiedene andere Objekte und verwaltet diese intern in einem Array.

Wichtige Features dieses Prototypen sind:

- Bei der Erstellung eines Container-Objekts kann der initiale Inhalt angegeben werden.
- Ein neuer Eintrag kann mit der Methode `create(element)` hinzugefügt werden.
- Die Methode `getAll()` liefert den Inhalt des Containers als Array zurück.
- Es muss eine Möglichkeit geben, auf alle bisher erstellten Container-Objekte (nicht: die Elemente in einem Container-Objekt) zuzugreifen.

Stellen Sie durch Ausprobieren fest, ob sich Ihre Implementierung an diese Spezifikation hält.

Aufgabe 2: Revealing Module-Entwurfsmuster

In der Vorlesung haben Sie das Entwurfsmuster Revealing Module unter Zuhilfenahme sog. IIFEs (*Immediate Invoked Function Expressions*) kennengelernt.

Modularisieren Sie Ihren in Aufgabe 1 erstellten Prototypen gemäß dieses Entwurfsmusters.

Verwenden Sie den gleichen Code wie in Aufgabe 1, um die Korrektheit Ihrer Implementierung zu testen.

Aufgabe 3: CommonJS

In der Vorlesung haben Sie das CommonJS-Format zur Modularisierung von JavaScript kennengelernt.

Wandeln Sie Ihren Prototypen in ein CommonJS-Modul um und verteilen Sie so den Modulcode und den Testcode auf zwei separate Dateien.

Aufgabe 4: Vererbung 1 (TypedContainer)

Realisieren Sie einen spezialisierten `TypedContainer`, der nur Objekte gleichen Typs aufnimmt. Der Typ soll als Konstruktorargument angegeben werden können. Wird versucht ein anderes Objekt einzufügen, soll eine Ausnahme geworfen werden.

Hinweis: Zum Loggen von Fehlern sollte die Methode `console.error()` verwendet werden, da diese unabhängig von normalen Log-Ausgaben in eine separate Datei weitergeleitet werden kann.

Aufgabe 5: Vererbung 2 (SecureContainer)

Serverseitig können sensible Daten vorliegen, die bei einer Anfrage nicht mitgesendet werden sollen. Realisieren Sie einen weiteren spezialisierten Container, dem Sie ein Funktions-Array als Konstruktorargument übergeben geben. Diese Funktionen soll beim Aufruf von `getAll()` auf die intern verwalteten Objekte angewendet werden, um so sensible Daten in der Rückgabe (und nur dort) zu entfernen.

Anbei finden Sie zum Testen einen Mock-Datensatz *MOCK_DATA.json* mit kritischen Daten der Nutzer (nämlich E-Mail-Adressen).