

Modul "classless Objects" – Übungsaufgaben

Aufgabe 1

Erstelle ein Objekt `person` mit den Eigenschaften `firstName` und `lastName`. Überprüfe, ob die Eigenschaft `age` existiert, und gib das Ergebnis in der Konsole aus.

Aufgabe 2

Erstelle ein Objekt `car` und füge die Eigenschaften `make`, `model` und `year` hinzu. Verwende die `delete`-Anweisung, um die Eigenschaft `year` zu entfernen, und überprüfe anschließend, ob sie noch vorhanden ist.

Aufgabe 3

Erstelle ein Objekt `book` und füge die Eigenschaft `title` hinzu. Erstelle eine Kopie des Objekts und ändere `title` in der Kopie. Überprüfe, ob sich das Original geändert hat.

Aufgabe 4

Erstelle ein Objekt `user` und füge eine Methode `greet` hinzu, die den Text „Hello!“ in der Konsole ausgibt. Verwende die Methode und überprüfe das Ergebnis.

Aufgabe 5

Erstelle ein Objekt `rectangle` mit den Eigenschaften `width` und `height`. Füge Getter- und Setter-Methoden für `area` hinzu, um die Fläche basierend auf `width` und `height` zu berechnen.

Aufgabe 6

Erstelle ein Objekt `config` und definiere die Eigenschaft `theme` als nicht überschreibbar und nicht konfigurierbar. Teste, ob du `theme` ändern oder löschen kannst.

Aufgabe 7

Erstelle eine Factory-Funktion `createPerson` mit den Parametern `firstName` und `lastName`. Verwende die Funktion, um ein neues Objekt zu erstellen und gib es in der Konsole aus.

Aufgabe 8

Erstelle eine Constructor-Funktion `Animal`, die die Eigenschaften `species` und `age` setzt. Erzeuge ein neues Objekt und gib es in der Konsole aus.

Aufgabe 9

Erstelle ein Objekt `vehicle` und verwende `Object.create()` um ein neues Objekt `bike` zu erstellen, das von `vehicle` erbt. Füge dem Prototyp `vehicle` eine Methode `move` hinzu und rufe diese über `bike` auf.

Aufgabe 10

Erstelle ein Objekt `account` und füge eine Eigenschaft `balance` hinzu. Erstelle eine flache Kopie und überprüfe, ob beide Objekte unabhängig voneinander sind, indem du `balance` in einem der beiden änderst.

Aufgabe 11

Erstelle ein verschachteltes Objekt `user` mit `address` als innerem Objekt. Führe eine tiefe Kopie durch und überprüfe, ob Änderungen im inneren Objekt des Klons das Original nicht beeinflussen.

Aufgabe 12

Definiere eine Methode `addItem` auf dem Prototyp eines `Store`-Objekts. Erstelle ein `store`-Objekt und füge ein neues Item hinzu. Überprüfe, ob das Item korrekt hinzugefügt wurde.

Aufgabe 13

Erstelle eine Methode `toUpperCase` im Prototyp von `String`, die den gesamten String in Großbuchstaben umwandelt. Verwende diese Methode an einem Beispielstring.

Aufgabe 14

Erstelle ein Objekt `product` mit einer nicht-änderbaren und nicht-überschreibbaren Eigenschaft `id`. Verwende `Object.defineProperty` und überprüfe, ob die Eigenschaft tatsächlich unveränderlich ist.

Aufgabe 15

Verwende `Object.getOwnPropertyDescriptor` auf einem Objekt, um die Konfigurationseinstellungen der Eigenschaft `price` anzuzeigen. Zeige die Einstellungen in der Konsole an.

Aufgabe 16

Erstelle eine Factory-Funktion `createRectangle` mit den Parametern `width` und `height`. Füge eine Methode `getArea` hinzu, die die Fläche berechnet. Überprüfe die Methode mit einem Beispielrechteck.

Aufgabe 17

Erstelle eine Constructor-Funktion `Person`, die Getters und Setters für die Eigenschaft `fullName` enthält. Teste die Getter- und Setter-Funktionalität.

Aufgabe 18

Füge `Array.prototype` eine Methode `first` hinzu, die das erste Element eines Arrays zurückgibt. Teste die Methode an einem Beispielarray.

Aufgabe 19

Erstelle ein `Employee`-Objekt mit einem nicht-überschreibbaren Prototyp. Versuche, den Prototyp zu ändern, und überprüfe das Ergebnis.

Aufgabe 20

Schreibe eine Funktion `checkProperty`, die überprüft, ob eine Eigenschaft im Prototyp eines Objekts existiert, und gib das Ergebnis zurück. Teste die Funktion an einem Objekt und seinem Prototyp.