1 Einleitung

Die meisten Entwickler verbinden objektorientierte Programmierung mit den üblichen an Hochschulen gelehrten Sprachen wie C++ und Java, die auf Klassen beruhen. Bevor Sie in diesen Sprachen irgendetwas tun können, müssen Sie zunächst eine Klasse erstellen, selbst wenn Sie nur ein einfaches Befehlszeilenprogramm schreiben wollen. Auch die üblichen Entwurfsmuster in der Branche stützen das Prinzip der Klassen. In JavaScript aber gibt es keine Klassen, und das ist einer der Gründe für die Verwirrung, die sich bei Personen breitmacht, die diese Sprache nach C++ oder Java zu lernen versuchen.

Objektorientierte Sprachen zeichnen sich durch folgende Charakteristika aus:

- **Kapselung**: Daten können zusammen mit der Funktionalität gruppiert werden, die mit diesen Daten arbeitet. Das ist, einfach ausgedrückt, die Definition eines Objekts.
- **Aggregation**: Ein Objekt kann auf ein anderes verweisen.
- **Vererbung**: Ein neu erstelltes Objekt erhält die gleichen Merkmale wie ein anderes Objekt, ohne dass dessen Funktionalität ausdrücklich dupliziert werden muss.
- **Polymorphismus**: Eine Schnittstelle kann von mehreren Objekten implementiert werden.

Auch JavaScript weist alle diese Merkmale auf, doch da die Sprache keine Klassen kennt, sind sie anders umgesetzt, als Sie es vielleicht erwarten. Auf den ersten Blick kann ein JavaScript-Programm sogar wie ein prozedurales Programm aussehen, wie Sie es in C schreiben würden. Wenn Sie eine Funktion schreiben und ihr einige Variablen übergeben, dann haben Sie ein funktionierendes Skript, in dem es scheinbar keine Objekte gibt. Ein genauerer Blick auf die Sprache enthüllt jedoch das Vorhandensein von Objekten, da die Punktschreibweise verwendet wird.

In vielen objektorientierten Sprachen wird die Punktschreibweise eingesetzt, um auf die Eigenschaften und Methoden von Objekten zuzugreifen, und JavaScript geht syntaktisch ebenso vor. Allerdings schreiben Sie in JavaScript keine Klassendefinitionen, importieren keine Pakete und schließen keine Headerdateien ein. Sie fangen einfach an, mit den gewünschten Datentypen Code zu schreiben. Später können Sie diese Typen beliebig gruppieren. Sicherlich ist es möglich, JavaScript auf prozedurale Weise zu schreiben, aber die wirklichen Vorteile zeigen sich erst, wenn Sie die objektorientierten Aspekte ausnutzen. Darum geht es in diesem Buch.

Damit wir uns nicht falsch verstehen: Viele der Prinzipien, die Sie in den herkömmlichen objektorientierten Programmiersprachen kennengelernt haben, müssen nicht unbedingt auch für JavaScript gelten. Das irritiert Anfänger meistens, doch bei der Lektüre dieses Buchs werden Sie schnell feststellen, dass Sie dank der schwach typisierten Natur von JavaScript weniger Code schreiben müssen als in anderen Sprachen, um dieselben Aufgaben zu erledigen. Sie können einfach mit dem Schreiben beginnen, ohne erst die erforderlichen Klassen zu planen. Brauchen Sie ein Objekt mit besonderen Feldern? Erstellen Sie es ad hoc, wann immer Sie wollen! Haben Sie vergessen, einem Objekt eine Methode hinzuzufügen? Kein Problem – ergänzen Sie sie später.

Auf den folgenden Seiten lernen Sie die besondere Art und Weise der objektorientierten Programmierung in JavaScript kennen. Werfen Sie Ihre Vorstellungen über Klassen und klassenbasierte Vererbung über Bord und lassen Sie sich in die prototypenbasierte Vererbung und die Konstruktorfunktionen einführen, die ein ähnliches Verhalten zeigen. Sie erfahren hier, wie Sie Objekte erstellen, wie Sie eigene Typen definieren, wie Sie die Vererbung einsetzen und wie Sie Objekte noch auf viele andere Weisen bearbeiten, um den größten Nutzen aus ihnen zu ziehen. Kurz gesagt, Sie lernen alles, was Sie für die professionelle JavaScript-Entwicklung wissen müssen. Viel Vergnügen!

1.1 Zielgruppe

Dieses Buch dient als Leitfaden für Personen, die sich bereits in objektorientierter Programmierung auskennen und genau wissen wollen, wie sie in JavaScript funktioniert. Wenn Sie mit Java, C# oder der

objektorientierten Programmierung in anderen Sprachen vertraut sind, ist das schon ein guter Anhaltspunkt dafür, dass dieses Buch das richtige für Sie ist. Insbesondere richtet sich dieses Buch an die drei folgenden Gruppen von Lesern:

- Entwickler, die mit den Prinzipien der objektorientierten Programmierung vertraut sind und sie auf JavaScript anwenden möchten.
- Entwickler von Webanwendungen und Node.js-Entwickler, die versuchen, ihren Code wirkungsvoller zu strukturieren.
- JavaScript-Neulinge, die ein tieferes Verständnis der Sprache gewinnen möchten.

Dieses Buch ist nicht für Anfänger gedacht, die noch nie in JavaScript programmiert haben. Um dem Text folgen zu können, brauchen Sie ein gutes Verständnis dafür, wie JavaScript-Code geschrieben und ausgeführt wird.

1.2 Überblick

Kapitel 2: Primitive Typen und Referenztypen gibt eine Einführung in die beiden verschiedenen Wertetypen in JavaScript, nämlich primitive Typen und Referenztypen. Sie erfahren hier, worin sie sich unterscheiden und warum die Kenntnis dieser Unterschiede wichtig ist, um JavaScript insgesamt zu verstehen.

Kapitel 3: Funktionen erklärt alle Einzelheiten von Funktionen in JavaScript. Vor allem sogenannte First-Class-Funktionen (Funktionen erster Klasse) machen JavaScript zu einer so interessanten Sprache.

Kapitel 4: Objekte stellt ausführlich den Aufbau von Objekten in JavaScript vor. JavaScript-Objekte verhalten sich anders als Objekte in anderen Sprachen, weshalb ein genaues Verständnis ihrer Funktionsweise unverzichtbar ist, um die Sprache zu beherrschen.

Kapitel 5: Konstruktoren und Prototypen erweitert die vorangegangene Erörterung von Funktionen durch eine genauere Darstellung von Konstruktoren. Alle Konstruktoren sind zwar Funktionen, aber sie werden auch ein kleines bisschen anders eingesetzt. In diesem Kapitel lernen Sie die Unterschiede kennen und erfahren, wie Sie Ihre eigenen Typen erstellen können.

Kapitel 6: Vererbung erklärt, wie die Vererbung in JavaScript bewerkstelligt wird. Auch wenn es in JavaScript keine Klassen gibt, heißt das nicht, dass eine Vererbung unmöglich wäre. In diesem Kapitel lernen Sie die prototypische Vererbung und ihre Unterschiede zur klassenbasierten Vererbung kennen.

Kapitel 7: Objektmuster führt gebräuchliche Objektmuster vor. In JavaScript gibt es viele verschiedene Möglichkeiten, um Objekte anzulegen und zusammenzustellen. In diesem Kapitel lernen Sie die am häufigsten dafür verwendeten Muster kennen.

1.3 Hilfe und Unterstützung

Wenn Sie Fragen haben, Kommentare schreiben oder in irgendeiner anderen Form Rückmeldung zu diesem Buch geben möchten, suchen Sie bitte die Mailingliste unter *http://groups.google.com/group/zakasbooks* (in englischer Sprache) auf.