## TypeScript

Die Programmiersprache TypeScript zeichnet sich durch ein mächtiges statisches Typsystem aus, das hilft, Programmierfehler zu vermeiden. Dank verschiedener Operatoren und Konstrukte bietet das JavaScript-Superset aber auch die Flexibilität, Verknüpfungen und Abhängigkeiten zwischen Typen zu erstellen. TypeScript lässt sich nahtlos in JavaScript-Projekte einbinden und ist mittlerweile aus der Welt der Frontend-Frameworks nicht mehr wegzudenken.

ab Seite 7

## Kotlin

Kotlin verbindet funktionale Konzepte mit objektorientierter Programmierung auf der JVM. Als Alternative zu Java verspricht Kotlin höhere Effizienz: Die Programmiersprache punktet mit klarer Struktur und guter Lesbarkeit. Kotlin lässt sich auch jenseits der JVM nutzen: Das SDK Kotlin Multiplatform Mobile eröffnet die Möglichkeit, native Anwendungen plattformübergreifend zu entwickeln und dabei einmal erstellte Businesslogik wiederzuverwenden.

ab Seite 45

## Rust

Mit dem Ownership-Konzept zielt Rust auf mehr Speichersicherheit ab und verzichtet dabei auf den Overhead eines Garbage Collector, legt die Hürde für Neueinsteiger aber recht hoch. Das gilt gleichermaßen für die asynchrone Programmierung, für die eine eigene Laufzeitumgebung wie Tokio notwendig ist. Ein mächtiges Werkzeug in Rust sind Makros, die weit über die aus C/C++ bekannten einfachen Textersetzungen hinaus gehen. Sie sind nicht nur weniger fehleranfällig, sondern eröffnen auch die Möglichkeit, Domain-specific Languages in den Rust-Code zu integrieren.

ab Seite 73

## Go

Zu den Stärken von Go zählt Concurrency. In Gestalt der Goroutinen ist nebenläufige Programmierung in der Sprache elegant und effizient umgesetzt. Sie vermeiden die gefürchteten Data Races, daher eignet sich Go besonders für verteilte Anwendungen. Den Interfaces verdankt Go zudem die Flexibilität dynamisch typisierter Sprachen und mittels der seit Go 1.18 endlich verfügbaren Generics lassen sich nun auch generische Funktionen, Strukturen sowie Channels erstellen. Mit Blick auf die Sicherheit der Software Supply Chain bietet Go eine Reihe von Konzepten, die helfen, Angriffe zu vermeiden.

ab Seite 93



## Sprachenvielfalt

Für jede Aufgabenstellung findet sich eine geeignete Programmiersprache. In der Quantenprogrammierung lässt sich Microsofts Q# sowohl eigenständig als auch im Zusammenspiel mit Python und .NET-Sprachen verwenden. Etablierte Platzhirsche wie Java und C++ investieren kontinuierlich in neue Funktionen und Konzepte, um sich gegen die Konkurrenz zu behaupten. Rasante Build-Tools wie Vite verleihen TypeScript im JavaScript-Ökosystem noch mehr Fahrt und das Gespann aus Flutter und Dart empfiehlt sich als universelles Cross-Plattform-Werkzeug.

ab Seite 125