Warum sollte ich Go lernen Blog

Go die etwas andere Programmiersprache

Go oder Golang ist eine Open-Source-Programmiersprache, die bei Google von Robert Griesemer, Rob Pike und Ken Thompson im Jahr 2009 erstellt wurde.Go wird kompiliert und statisch wie C/C++, aber mit Garbage Collection und streng typisiert ist.Es wird von einigen Produktionssystemen von Google verwendet und auch die bekannten Open Source Projekte Docker und Kubernetes sind in Go programmiert.Go ist keine Weiterentwicklung bekannter Programmiersprachen wie Java, C# oder C/C++, obwohl es zu dieser Kategorie gehört.Denn Go ist anders, Go ist nicht rein objektorientiert aber auch nicht rein funktional, es kommt aber den C Konzepten am nächsten.

Die Ziele von Go

Go versucht nicht einen Preis im Design von Programmiersprachen zu gewinnen, denn Go folgt pragmatischen Zielen:EinfachheitPrägnanzLesbarkeitConcurrency (gleichzeitige Abläufe)PerformanceDas Hauptziel dabei ist, einfachen und sauberen Programmcode (simple and clean) zu erstellen, der effizient ausgeführt wird.

Go Programme sind native portabel

Analog wie C/C++ Programme sind Go Programme native kompiliert. Go Programme sind in der Regel statisch kompiliert und enthalten die Go run-time.Go Programme sind im ersten Moment eher gross, so umfasst ein einfaches Hello World mehr als 1MB:Die Go runtime ist keine virtuelle Maschine und dennoch sind Go Programme native portabel. Diese Portabilität wird aber über die Kompiliation mit dem Operating System (GOOS) und der Hardware Architektur (GOARCH) hergestellt. Der gleiche Programmcode kann für viele verschiedene Zielsysteme wie z.B. Linux, Darwin (iOS) und Windows kompiliert werden.

Warum Go

Warum sollte ich nun eine neue Programmiersprache wie Go lernen, es gibt ja schon viele andere gute Sprachen wie C++, C# oder Java und die lassen praktisch keine Wünsche offen.Zudem ist ja jede Sprache einfach, wenn man solche beherrscht.Aus der Sicht von Java oder C# bietet Go doch etliche Nachteile:Go kennt kein Function OverloadingGo kennt keine Klassen sondern nur StrukturenGo kennt keine KonstruktorenGo kennt keine VererbungGo kennt nur die vereinfachte Kapselung, es fehlt das protected weil ja die Vererbung fehlt.Go entspricht nicht den klassischen OO Sprachen und ist damit nicht einfacher lernbar, man muss die Konzepte grundlegend verstehen.Go kennt keine GenericsGo kennt kein Exception Handling...Go liegt technisch zwischen Java, C und Python : Was sind denn nun die echten Vorteile von Go ?Go Code ist cleanerGo Code muss strengen Richtlinien zwingend folgen (z.B Blockklammern)Vereinfachte Sprachmittel, nur eine Schleife (for)Begrenztes Set an Collections (Arrays, Slices und Maps)Go Methoden und Funktionen können mehrere return Werte liefernGo Code ist schnell kompiliertGo hat einen Garbage CollectorGo kennt Interfaces, die durch alle Strukturen über Methoden frei implementierbar sindGo kennt PolymorphismusGo kennt die Komposition aber keine HierarchieGo unterstützt einfaches Unit Testing und BenchmarksGo bietet built-in Concurrency (goroutines) Featuresgoroutines sind nicht ThreadsGo Channels dienen der Kommunikation zwischen Concurrent Tasks und deren SynchronisierungGo löst das 10k Problem (siehe)Einfaches Build System, kein Bedarf an Makefiles und komplexen Build Konfigurationen (einer aus meiner Sicht grössten Vorteile)...

Anwendungsbereiche von Go

Go eignet sich zur Zeit vor allem für die Programmierung von Server Anwendungen. Hier sticht Go vor allem bei der Webentwicklung hervor. Die Entwicklung von Web Anwendungen basiert auf der HTML Template Language von Go. Die Sprache wird aber auch für die Integration via Web Services und API's eingesetzt.Go eignet sich nicht für die Entwicklung klassischer Desktop Anwendungen.Wir setzen Go für die Entwicklung von Tools, für die Integration und im Web Development ein. Diese Site wird mit dem von uns entwickelten stack.ch Server gehostet.Bekannte Firmen wie Uber, Google, Netflix, Dropbox, Dell, eBay, Yahoo, Zelando, Zynga. Docker, MongoDB programmieren mit Go.

Demos

Anhand der folgenden Demos zeigen wir Ihnen Go anhand von echten Beispielen:Goroutine creationGoroutine atomic not sync / syncWebserver ChannelsHTML TemplatesDie Beispiele finden Sie unter https://github.com/stackch/demo

Links

Wichtige und interessante Golang Links:golang.orgThe Go BlogGo TalksGo Git RepoRob Bike about Go

Unsere Erfahrungen

Der Einsatz von Go hat sich für uns gelohnt und wir konnten die folgenden Erfahrungen sammeln:Ein geübter C/++, C# oder Java Programmierer lernt Go in ca. 1 Monat.Mit Go können wir Probleme einfacher und auch schneller lösen.Der einfache Build Prozess bringt am meisten Vorteile.Die Umstellung von der klasischen Objektorientierung zu Go Interfaces und Strukturen war nicht ganz einfach.Einen wesentlichen Anteil an Programmcode konnten wir von Java her portieren (z.B. die generische map).Man findet in der Community zu praktisch allen Problemen eine Lösung. Die Online Doku ist sehr gut.Einige Dinge wie z.B. das XML Parsen oder Rendering sind gewöhnungsbedürftig.JSON Support und HTML Templates sind sehr effizient.Wir setzen wo immer auch möglich Go anstelle von C/C++, C# oder Java ein.

Feedback

War dieser Blog für Sie wertvoll. Wir danken für jede Anregung und Feedback

Go Bildungsweg

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

https://www.simtech-ag.ch/klasischen