Fernkurs Google Go Programmierung (golang)

Dieser Google Go Fernkurs bietet eine kompakte und professionelle Einführung in die Programmierung mit Google Go (golang).

Information

Kurscode: FGOGDies ist ein Fernkurs ohne zeitliche und örtliche Präsenz..

Ablauf Fernkurs

Dieser Fernkurs deckt die gleichen Themen wie der Standard Google Go Programmierung Kurs ab. Sie erhalten die gleichen Unterlagen elektronisch und arbeiten den Kurs im Selbststudium nach eigenem Tempo und ermessen durch. Für die Übungen erhalten Sie von uns ein Ubuntu VM Ware Image. Für Fragen stehen wir per EMail zur Verfügung und erhalten damit direkten professionellen Support.Der publizierte Termin dient rein organisatorischen Zwecken.

Einleitung

Die Programmiersprache Google Go hat sich seit der offiziellen Freigabe im Jahr 2009 zur Programmiersprache des Jahres entwickelt. Immer mehr wird Google Go auch in grösseren Projekten eingesetzt.Go ist eine pragmatische Programmiersprache und verbindet Konzepte von C,Java und Python. Go Programme sind zudem sehr schnell, da solche native kompiliert werden und damit wird kein zusätzliches Environment benötigt. Go ist zudem portabel und läuft auf allen gängigen Plattformen.Von der Sprache her ist Go streng typisiert, bietet aber keine Klassen sondern Strukturen mit Methoden unterscheidet sich damit von klassischen objektorientierten Sprachen wie z.B. Java. So ist z.B. auch die Vererbung mit Go nicht möglich und dies ganz bewusst, denn es bietet die Komposition von Typen und auch Interfaces. Go Programme sind damit zwingend schlanker aufgebaut. Go arbeitet analog C mit Pointern, aber ohne deren Arithmetik.Mit Google Go lassen sich parallele Abläufe mit Goroutines schnell und einfach programmieren. Go bietet reiche Concurrency Features und dies ist eines der wertvollsten Elemente dieser Sprache.Go wird zur Zeit vor allem serverseitig eingesetzt und enthält einen integrierten Web Server (Http). Damit lassen sich auch Web Applikationen effizient und einfach programmieren.Durch den einfachen Aufbau der Sprache bietet Go zu allen Aufgaben eine ähnliche Lösung und ist damit weniger vielfältig, dies ist ein enormer Vorteil für die Arbeit in Teams. Man kann den Programmcode leichter verstehen.Mit diesem Kurs steigen Sie in den Google Go Zug ein. Wir arbeiten neben der Theorie vor allem mit Codebeispielen und Übungen. Wir referenzieren immer wieder auch die Quellen im Internet, so dass Sie umfassend informiert sind.Mit dem Jumpstart beginnen wir zudem gleich mit dem Schreiben erster Unit Tests. Testing und Benchmarks sind direkt in Google Go integriert und lightweight.Dieser Google Go Kurs vermittelt die Konzepte anhand von gezielt gewählten Beispielen und Anwendungsfällen. Damit folgt der Kursaufbau den pragmatischen Zielen von Go. Die Theorie wird direkt im Beispielcode oder als Notizen vermittelt. So verfügen Sie nach dem Kurs über umfassende Use Cases von Go.

Ihr Nutzen

- Sie kennen die Ziele und Architektur von Google Go
- Sie kennen die zentralen Sprachkonzepte von Google Go
- Sie können Google Go Programme schreiben, builden und ausführen, oder einfach gesagt: "Sie können mit Go programmieren".
- Sie wissen wie man grössere Google Go Programme strukturiert und erstellt.
- Sie kennen die wichtigsten Quellen im Internet.

Verwandte Kurse

- Google Go Concurrent Programming (GOCO)
- Google Go für C# Programmierer (golang for c# programmers) (GOCS)
- Google Go für C++ Programmierer (golang for c++ programmers) (GOCP)
- Google Go für Java Programmierer (golang for java programmers) (GOJA)
- Google Go für PHP Programmierer (golang for php programmers) (GOPH)

Voraussetzungen

Praxiserfahrung einer Programmiersprache wie Javascript, Visual Basic oder ähnlich.

Teilnehmerkreis

Programmierer, Entwickler, Projektleiter

Unterlagen

- Tutorial
- Code Walks
- Internet / Intranet

Folgekurse

- Google Go Concurrent Programming (GOCO)

Inhalt

- Einführung
- Installation und IDE
- Go Workspace
- Pakete und Imports
- Go Modules
- Jumpstart
  - Functional
  - Object
- Basistypen, Typumwandlungen, Typinferenz, Nullwerte, Konstanten, Auto Increment (iota)
- Pointers
- Strings, Buffer, Runes
- Variablen und Konstanten
- Scope
- Operatoren (Assignment, Arithmetic, Comparison, Logical, Bit)
- Type Cast
- Integer Overflow
- Entscheidungen mit if und switch
- For-Schleife
- Pointer und Strukturen
- Arrays und Slices
- Slice Tricks
- Maps
- make und new
- Funktionen (func) und Closures
- Strukturen (struct)
- Verzögerte Ausführungen mit defer
- Naming Conventions, Exportables
- Panic und Recover
- Methoden und Strukturen
- Interfaces
  - Slice of Interfaces
  - Implementation
  - Extending Struct
  - Type Assertions
  - Type Switches
- Stringer Interface
- Objektorientierte Programmierung mit Go und Interfaces
  - Embedding als Ersatz für Vererbung
  - New Method
  - Encapsulation
  - Polymorphismus

- Fehlerbehandlung
  - errors Package
  - Custom Errors
  - Error struct
- Reflection
  - TypeOf, ValueOf
  - StructOf, StructField
  - MakeSlice, MakeMap, MakeOf
  - MakeFunc
- Das io-Packet (Reader und Writer)
- Goroutinen (schlanke Threads) und Kanäle
  - Goroutines vs Threads
  - Goroutine Creation
  - Channels Deadlock
  - Channels Buffered/Unbuffered
  - Waiting for Channels (WaitGroup)
  - Channels select
  - Dynamic Channels
- Go Generics
- Synchronisieren mit sync.Mutex
  - Lock/Unlock
  - Atomic Sync
- Logging
- Package Testing, Unit Testing
- Benchmarks
- Subtests, Sub-benchmarks
- Diverses

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/theorie