

Go Programmeren

Doelgroep Cursus Go Programmeren

De cursus Go Programmeren is bedoeld voor developers die willen leren programmeren in de taal Go en de mogelijkheden ervan willen leren kennen.

Voorkennis training Go Programmeren

Om aan deze cursus deel te nemen is voorafgaande kennis van en ervaring met programmeren in een moderne programmeertaal zoals Java, C# of Python noodzakelijk.

Uitvoering cursus Go Programmeren

De theorie wordt behandeld aan de hand van presentaties. Illustratieve demo's worden gebruikt om de behandelde concepten te verduidelijken. Er is voldoende gelegenheid om te oefenen en afwisseling van theorie en praktijk. De cursustijden zijn van 9.30 tot 16.30.

Certificering cursus Go Programmeren

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat Go Programmeren.

Duur: 3 dagen

Prijs: € 1999

Open Rooster



Go Programming



Inhoud Cursus Go Programmeren

In de cursus Go Programmeren leren de deelnemers applicaties te ontwikkelen met de beknopte en efficiënte programmeertaal Go. Go is syntactisch vergelijkbaar met C, maar met de extra voordelen van memory safety, garbage collection en structural typing. De concurrency mechanismen in Go zijn gebaseerd op Communicating Sequential Processes (CSP) en Go programma's kunnen zo het beste halen uit multicore machines. Go is een snelle statisch getypeerde en gecompileerde taal maar voelt aan als een dynamisch getypeerde en geïnterpreteerde taal.

Go Intro

De cursus gaat van start met een bespreking van de syntax van Go met data types, type inference, arrays, control flow en operators. Hierbij komt ook het verschil tussen rvalues en lvalues aan de orde en wordt aandacht besteed aan immutable data.

Functions en Closures

Vervolgens wordt ingegaan op functions en parameter passing in Go. Het onderscheid tussen call by value en call by reference wordt uitgelegd. En ook variadic functions, recursion en closures komen aan de orde.

Pointers

Dan is het tijd om aandacht te besteden aan het gebruik van pointers in Go. Het verschil met pointers in C wordt besproken evenals pointer arithmetic, nil pointers en pointers tot pointers.

Classes

Ook classes in Go zijn onderdeel van het programma. De cursus gaat in op de twee manieren waarop Go een alternatief biedt voor traditionele inheritance. De eerste is embedding en kan worden beschouwd als een geautomatiseerde vorm van composition of delegation. De tweede is het gebruik van Go interfaces, die runtime polymorphism bieden.

Error Handling

En ook de specifieke manier van error afhandeling in Go met het error en panic interface komt aan de orde. Hierbij wordt tevens aandacht besteed aan het recover interface en de analyse van stack traces.

Concurrency

Tenslotte wordt ingegaan op concurrency in Go met Go routines, sending and receiving met channels en worker pools en synchronisatie mechanismen.

Modules Cursus Go Programmeren

Module 1 : Intro Go	Module 2 : Language Syntax	Module 3 : Operators and Control Flow
Origins of Go Features of Go Compilation Model Type Inference Concurrency Support Go Routines Native Binaries Intentionally Exclusions Inheritance Operator Overloading Installing Go Comments Go Programs	Identifiers Data Types Integers and Floats Strings and Booleans Derived Types Variable Declaration Static Type Declaration Type Inference lvalues and rvalues Constants String Literals UTF-8 Sequences Immutability	Operator Types Miscellaneous Operators Operator Precedence if and else Nested if switch Statement select Statement for Loop Nested Loops Infinite Loops range Keyword break and continue goto Statement
Module 4 : Functions	Module 5 : Arrays	Module 6 : Pointers
Function Definition Function Declaration Calling Functions Local and Global Variables Parameters Return Values Call by Value and by Reference Functions as Values Function Closures Methods Variadic Functions Recursion	Array Types Declaring Arrays Initializing Arrays Accessing Array Elements Multi Dimensional Arrays Passing Arrays Variables and Adresses Slices Nil Slice Subslicing len() and cap() Functions append() and copy() Functions	Address Operator Pointer Type Accessing Pointers Pointer Arithmetic Comparison C Pointers Uage of Pointers Dereferencing Pointers Nil Pointers Array of Pointers Pointer to Pointer Pointers as Parameters Type Casting
Module 7 : Data Structures	Module 8 : Error Handling	Module 9 : Concurrency
User Defined Data Types type Statement struct Keyword Structure Definition Accessing Structure Members Passing Structures Pointers to Structures Object Oriented Programming Structs versus Classes Composition over Inheritance Polymorphism Maps, Keys and Values make() and delete() Function Interfaces	Error is Type Error Interface Panic Interface Reasons for Panic Comparison to Exceptions Using Multiple Return Values Unrecoverable Error Programmer Error Defer Execution Recover Interface Runtime panics Goroutines Stack Trace Analyzing Stack Traces	Concurrency versus Parallelism Goroutines versus Threads Multiplexing Channels Race Conditions and Deadlock Multiple Goroutines Declaring Channels Sending and Receiving with Channels Blocking by Default Unidirectional Channels Buffered Channels Worker Pools WaitGroup Select and Mutex