

Rust Design Patterns

Doelgroep Cursus Rust Design Patterns

De cursus Rust Design Patterns is bestemd voor Rust developers en software architecten die Rust idioms en Design Patterns willen toepassen bij het ontwerpen van Rust applicaties

Voorkennis Cursus Rust Design Patterns

Kennis van en ervaring met Rust is vereist. Ervaring met object georiënteerde analyse en design met UML wordt aanbevolen.

Uitvoering Training Rust Design Patterns

De concepten worden behandeld aan de hand van presentatie slides. De theorie wordt geïllustreerd met demo's van patterns in Rust. Er zijn oefeningen in design problemen waarbij Rust patterns worden toegepast.

Certificaat Rust Design Patterns

De deelnemers krijgen na het goed doorlopen van de cursus een certificaat Rust Design Patterns.

Duur: 3 dagen

Prijs: € 2250

[Open Rooster](#)



Rust Design Patterns



Inhoud Cursus Rust Design Patterns

In de cursus Rust Design Patterns leren de deelnemers over Design Patterns en idiomen die specifiek zijn voor de Rust language. Design Patterns zijn herhaalbare oplossingen voor veel voorkomende problemen bij software development. Design Patterns kunnen worden beschouwd als templates voor het oplossen van bepaalde problemen.

Rust Recap

De cursus Rust Design Patterns begint met het bespreken van de belangrijke kenmerken van de Rust language, waaronder Ownership, Moves, Shadowing, Guards, Crates, Closures en Traits.

Rust Idioms

Vervolgens wordt aandacht besteed aan de idiomen en conventies die specifiek zijn voor Rust. Dit betreft idiomen als Borrowed Type Arguments, Collections as Smart Pointers, Finalization in Destructors and On-Stack Dynamic Dispatch.

Behavioral Patterns

Vervolgens worden Design Patterns besproken die gerelateerd zijn aan behavior en communicatie tussen objecten. De specifieke Rust implementatie van bekende patterns als het Observer Pattern, het Command Pattern en het Iterator Pattern wordt besproken.

Structural Patterns

In de cursus Rust Design Patterns wordt ook aandacht besteed aan Structural Patterns die te maken hebben met de structuur van objecten en classes. Hierbij is er aandacht voor de Rust specifieke implementatie van patterns als het Adapter Patterns, het Façade pattern en het Composite pattern.

Functional Programming

Tot slot staan functionele patterns en hoe deze in Rust kunnen worden toegepast op het programma van de cursus Rust Design Patterns. Dit betreft patterns als Object Based API's, Type Consolidation en Wrapping Iterators.

Modules Cursus Rust Design Patterns

Module 1 : Rust Recap	Module 2 : Rust Idioms	Module 3 : Behavioral Patterns
Rust Data Types Ownership and Moves Type Anonymity Shadowing Guards Crates Closures Traits Designators Lifetimes Dynamic Dispatch	Borrowed Type Arguments Strings with format! Constructor Default Trait Collections as Smart Pointers Finalization in Destructors On-Stack Dynamic Dispatch Iterating over Option Pass Variables to Closure Privacy For Extensibility Foreign Function Interface	Command Interpreter Newtype RAII Guards Strategy Visitor Chain of Responsibility Mediator Observer Iterator Strategy
Module 4 : Structural Patterns	Module 5 : Functional Programming	
Adapter Composite Decorator Bridge Façade Builder Factory Method Compose Structs Prefer Small Crates Small modules	Object Based API's Type Consolidation Wrapping Iterators Programming Paradigms Generics as Type Classes Lenses and Prisms Profunctor Optics Anti Pattern Unneeded Clone Defer Polymorphism	