

## **Rust Design Patterns**

#### Audience Course Rust Design Patterns

The course Rust Design Patterns is intended for Rust developers and software architects who want to apply Rust Idioms and Design Patterns when designing Rust applications.

#### **Prerequisites Course Rust Design Patterns**

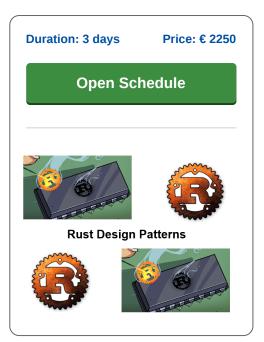
Knowledge of and experience with Rust is required. Experience with object oriented analysis and design with UML is recommended.

#### **Realization Training Rust Design Patterns**

The concepts are covered on the basis of presentation slides. The theory is illustrated with demos of patterns in Rust. There are exercises in design problems where Rust patterns are applied.

#### **Certificate Course Rust Design Patterns**

After successful completion of the course the participants receive a certificate Rust Design Patterns.



### **Content Course Rust Design Patterns**

In the course Rust Design Patterns the participants learn about design patterns and idioms that specific for the Rust language. Design patterns form a general repeatable solution to a commonly occurring problem in software design. Design patterns can be considered templates for how to solve a particular problem.

#### **Rust Recap**

The course Rust Design Patterns starts by reviewing the important features of the Rust language, including Ownership, Moves, Shadowing, Guards, Crates, Closures and Traits.

#### **Rust Idioms**

Next attention is paid to the idioms and conventions that are specific to Rust programming. This will include idioms like Borrowed Type Arguments, Collections as Smart Pointers, Finalization in Destructors and On-Stack Dynamic Dispatch.

#### **Behavioral Patterns**

Subsequently Behavioral Patterns that are related to object behavior and communication, are explored. The specific Rust implementation of familiar patterns like the Observer pattern, the Command pattern and the Iterator pattern, are discussed.

#### **Structural Patterns**

The course Rust Design Patterns also pays attention to structural design patterns that are concerned with the structure of objects and classes. This includes the Rust specific implementation of patterns like the Adapter pattern, the Façade pattern and the Composite pattern.

#### **Functional Programming**

Finally functional programming patterns and how they can be applied in Rust, are on the schedule of the course Rust Design Patterns. This will include topics such as Object Based API's, Type Consolidation and Wrapping Iterators.

info@spiraltrain.nl www.spiraltrain.nl Tel.: +31 (0) 30 – 737 0661 Locations Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online



# **Modules Course Rust Design Patterns**

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

Locations Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online