

Seminar: The Rust Programming Language

Ivana Zuzic

Technical University of Munich

January 31, 2021

Rust - Code Example

```
fn main() {
    let program = "+ + * - /";
    let mut accumulator = 0;
    for token in program.chars() {
        match token {
            '+' => accumulator += 1,
            '-' => accumulator -= 1,
            // ...
            _ => { /* ignore everything else */ }
        }
    }
    println!("The program \"{}\" calculates the value
    ↪ {}\"", program, accumulator);
}
```

"Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety."

- zero-cost abstractions
- move semantics
- guaranteed memory safety
- threads without data races
- trait-based generics
- pattern matching
- minimal runtime
- efficient C bindings

General:

- Block course intended for 14 attendees
- Latex paper (around 10 pages) is required
- Language (for the talks and the paper) is English

Talks:

- 30 minutes presentation + 15 minutes discussion for each topic
- Talks have to demonstrate a concept using real code
- Working code examples should be presented
- **Presence at all the talks is obligatory in order to pass the seminar!**

- Exact dates to be published by the end of the week on the course page
- Contact me about choosing a topic (from the list or your own) as soon as you enroll
- Schedule weekly 15 min meetings with me to track your progress
- **Start of May** - present me the **first draft of the paper**
- **Start of June** - present me the **first draft of the slides**
- **Talks around end of June**

- The course page: <https://www.in.tum.de/i02/lehre/sommersemester-21/seminare/rust/>
- The Moodle page
- Weekly meetings
- My email: ivana.zuzic@tum.de

Topic Suggestions

- 1 Comparison of Rust Basic Concepts with C++ and Java
- 2 IDEs, Tools and Module System for Rust
- 3 Ownership and Lifetimes
- 4 Reference Counting and Garbage Collection
- 5 Polymorphism, Traits, Generic Types
- 6 Collections (Vec, LinkedList, HashMap...) and Iterators in Rust
- 7 Functional Programming Concepts in Rust
- 8 Macros in Rust
- 9 Concurrency in Rust
- 10 Rust for Network Servers
- 11 Rust and WebAssembly for Web Applications
- 12 Rust for Embedded Programming
- 13 The Future of Rust
- 14 Unsafe Rust

Questions? :)