

Concepts of programming languages

Rust

Tobias van Driessel, Tom Freijsen, Ruben Schenkhuizen, Floris Schild, Mats Veldhuizen





Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Presentation Schedule

- ▶ Background
- ▶ Design principles
- ▶ What problems does Rust solve and how?
- ▶ Practical details



Background

- ▶ Personal project of Mozilla employee
- ▶ Sponsored by Mozilla Research
- ▶ First stable release (1.0) in 2015
- ▶ Used in Firefox and by Dropbox
- ▶ Most loved programming language - SO Developer Survey



Quick Overview

- ▶ Statically typed
- ▶ Functional **and** Imperative paradigms
- ▶ Strict language



Beautiful Quotes

Mozilla Research:

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

“It fuses the expressive and intuitive syntax of high-level languages with the control and performance of a low-level language.”



Features



Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Features

- ▶ zero-cost abstractions



Features

- ▶ zero-cost abstractions
- ▶ move semantics



Features

- ▶ zero-cost abstractions
- ▶ move semantics
- ▶ guaranteed memory safety



Features

- ▶ zero-cost abstractions
- ▶ move semantics
- ▶ guaranteed memory safety
- ▶ threads without data races



Features

- ▶ zero-cost abstractions
- ▶ move semantics
- ▶ guaranteed memory safety
- ▶ threads without data races
- ▶ trait-based generics



Features

- ▶ zero-cost abstractions
- ▶ move semantics
- ▶ guaranteed memory safety
- ▶ threads without data races
- ▶ trait-based generics
- ▶ pattern matching



Features

- ▶ zero-cost abstractions
- ▶ move semantics
- ▶ guaranteed memory safety
- ▶ threads without data races
- ▶ trait-based generics
- ▶ pattern matching
- ▶ type inference



Features

- ▶ zero-cost abstractions
- ▶ move semantics
- ▶ guaranteed memory safety
- ▶ threads without data races
- ▶ trait-based generics
- ▶ pattern matching
- ▶ type inference
- ▶ minimal runtime



Features

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

