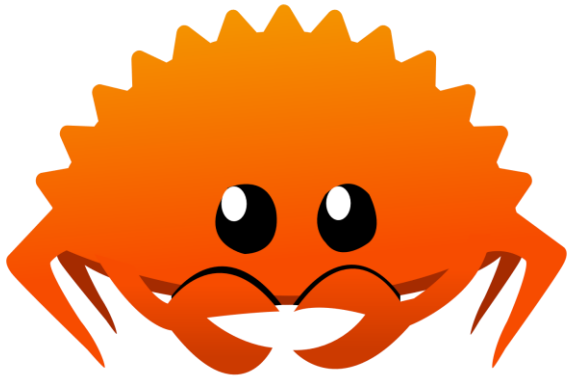


Rust

Simon Just TINF24B5

Was ist Rust

- Wahl einer Programmiersprache:

➢ Komfort und Sicherheit, aber langsamer

➢ Performance, aber komplex anzuwenden

- Anspruch von Rust: Geschwindigkeit, Sicherheit und Benutzerfreundlichkeit

  → Besserer Code bei wenigeren Kompromissen

Geschichte von Rust

- 2006 Hobbyprojekt von Graydon Hoares
- 2010 Präsentation seiner Arbeit auf einem Mozilla-Summit
- Förderung und Entwicklung des Projekts durch Mozilla
- 2017 Firefox mit Rust ausgeliefert
- Gründung der Rust Foundation
- 2021 Teile des Linux-Kernels in Rust geschrieben
- 2023 Rust im Kernel von Windows 11

Warum Rust

- Rust bietet die Geschwindigkeit von C++
- kombiniert mit der Sicherheit moderner Sprachen
- Es zwingt Entwickler, Speicher- und Besitzregeln klar zu definieren, was zu stabilerem und sichererem Code führt
- starkes Typsystem
- Es bietet Null-Cost-Abstraktionen, Pattern Matching, starke Generics und ein modernes Modul-System

Wofür wird es genutzt

- Systemprogrammierung
- Webentwicklung
- Eingebetteten Systemen
- Spiele- und Grafikentwicklung
- Cloud-Computing
- Sicherheit & Kryptographie

Speicher Management im Vergleich

	Pro	Kontra
Garbage Collection	- Fehlerfrei - Weniger Aufwand	- Keine Kontrolle - Langsamere runtime performance - Größere Programmgröße
Manuelles Management	- Volle Kontrolle - Schnellere runtime performance - Kleinere Programmgröße	- Fehleranfällig - Mehraufwand
Ownership Modell	- Volle Kontrolle - Schnellere runtime performance - Fehlerfrei - Kleinere Programmgröße	- Erheblicher Mehraufwand - Komplex - Steile Lernkurve

Ownership Modell

- Eigentümerschaft ist eine Reihe von Regeln, die bestimmen, wie ein Rust-Programm den Speicher verwaltet
- Wird eine Regel verletzt kann nicht kompiliert werden
- Regeln verlangsamen in der Laufzeit nicht das Programm

Stack vs Heap

- Beides Teil des Arbeitsspeichers
- Daten mit fester Größe → Stack
- Daten mit unbekannter oder variabler Größe → Heap
- Wird Speicher im Heap allokiert, wird der Zeiger auf den Stack gelegt
- Stack schneller
- Hauptzweck der Eigentümerschaft ist die Verwaltung des Heaps

    → unterschiedliches Verhalten, je nach dem, ob Daten im Stack oder im Heap gespeichert sind

Ownership Modell Regeln

- Jeder Wert in Rust hat einen Eigentümer (owner).
- Es kann immer nur einen Eigentümer zur gleichen Zeit geben
- Wenn der Eigentümer den Gültigkeitsbereich verlässt, wird der Wert aufgeräumt

Ownership Modell Regeln

```
{                       // s ist hier nicht gültig, es wurde noch nicht deklariert
    let s = "Hallo";    // s ist ab dieser Stelle gültig

    // etwas mit s machen
}                       // dieser Gültigkeitsbereich ist nun vorbei,
                        // und s ist nicht mehr gültig
```

Ownership Modell Regeln

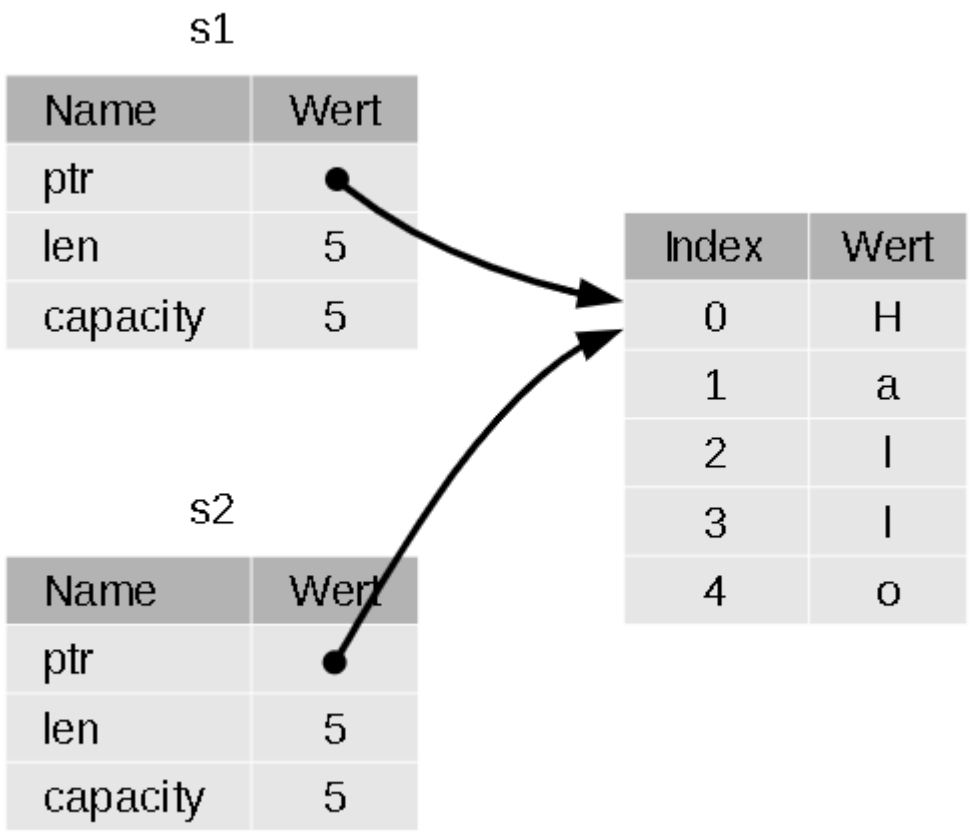
```
{
    let s = String::from("Hallo"); // s ist ab dieser Stelle gültig

    // etwas mit s machen
}                                  // dieser Gültigkeitsbereich ist nun vorbei,
                                   // und s ist nicht mehr gültig
```

Ownership Modell Regeln

```
let x = 5;
let y = x;
```

```
let s1 = String::from("Hallo");
let s2 = s1;
```

Ownership Modell Regeln

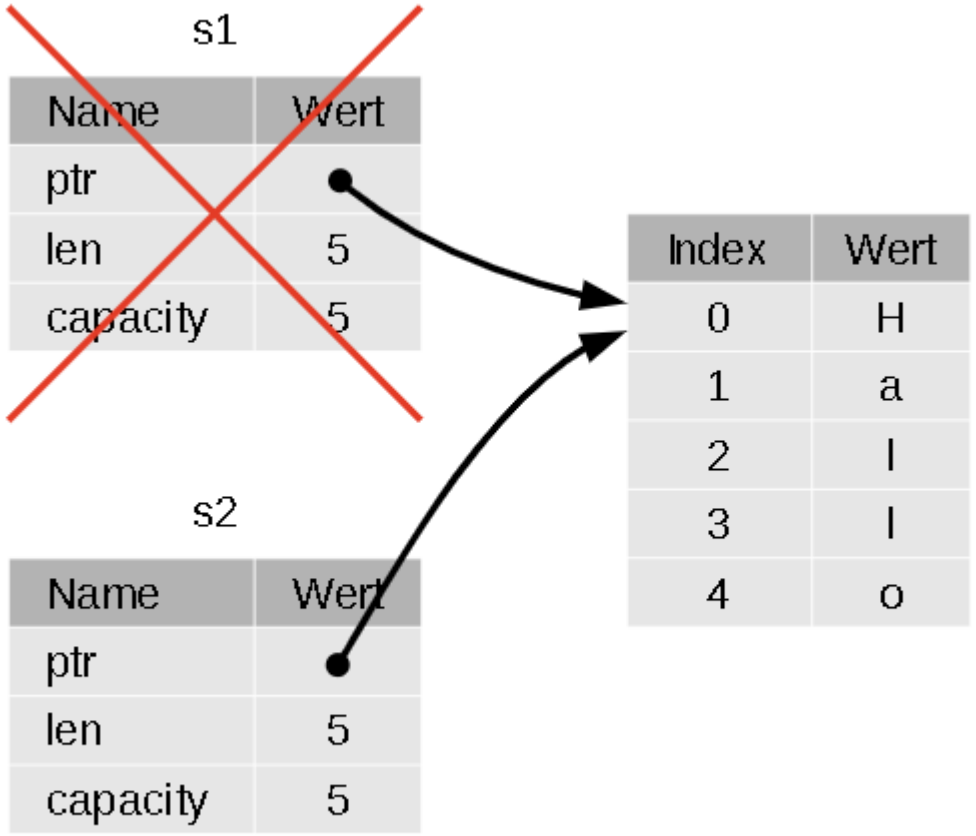


Ownership Modell Regeln

```
let s1 = String::from("Hallo");
let s2 = s1;

println!("{s1} Welt!");
```

→ Kompiler-Fehler

Ownership Modell Regeln



→ s1 wurde in s2 verschoben

Ökosystem

- Opensource
- Pakete können installiert werden (crates.io)
- Die meisten C Bibliotheken können importiert werden
- Insgesamt >163.000 Crates
- Betriebssystem entkoppelt

Performance

- Anspruch ähnlich schnell zu sein wie C/C++
- Zero cost abstractions
- Kein Overhead durch einen Garbage Collector
- Compile-Time Evaluation
- Direct Memory Access
- Aber: Rust braucht deutlich länger zum Kompilieren

Zuverlässigkeit

- Gute Typensicherheit
- „Wenn Ich Rust kompiliert bekomme, dann läuft es auch“
- Stürzt nur sehr selten zur Laufzeit ab
- Memory Safety
- Thread Safety
- Variablen sind per default immutable

Nachteile von Rust

- Steile Lernkurve
- Kompilieren dauert deutlich länger
- Kleineres Ökosystem als bei C++ oder Python
- Man schreibt oft mehr Code, um dieselbe Aufgabe zu lösen

Installation

- Installieren: https://rust-lang.org/tools/install
- Online ausprobieren: https://play.rust-lang.org/?version=stable&mode=debug&edition=2024

Codebeispiele

- Alle gezeigten Codebeispiele sind in meinem GitHub-Repo zu finden: https://github.com/SimonJ2222/Rust

Fragen?