

Seminararbeit Traits und Enums in Rust

Mario Occhinegro
HKA University of Applied Sciences

Inhaltsverzeichnis

Zusammenfassung

1 Einleitung

2 Enums

Enumerationstypen

Auf den ersten Blick identisch.

Java Enum: Rust Enum: Auf Konkretere Unterschiede gehen wir jetzt ein

2.1 Enums in Rust

- Algebraische Datentypen

2.1.1 Match Statement

- abgeschlossen
- an Haskell angelehnt
- mehr als nur if else
- sehr ergonomisch, aussagekräftig und kurz

2.1.2 Der Enum als algebraischer Datentyp

- beliebige Struktur
- werte können sich verändern
- flexibel
- pattern matching lässt uns die einzelnen Werte benutzen

2.1.3 Rekursive Enums und Datentypen

- braucht Box (wie Zeiger)
- Box sonst, rekursive Definition ohne Direktion

2.1.4 Nested Pattern Matching

- kann noch granulareres pattern matching betreiben

2.1.5 Generische Enums

- Enums können mit generischen Werten generiert werden

2.1.6 Verwendung des Rust Enums zur Vermeidung von Nullpointer-Ausnahmen

- Java hat ähnliches Konzept aber mit Klassen
- Nullpointer, der große Milliarden € Fehler

2.2 Enums in Java

- Enums sind Instanzen
- Instanz statisch und final (per default)

2.2.1 Normale Enums

2.2.2 Enums mit Werten

2.2.3 Enum Funktionen

2.3 Vergleich von Java und Rust Enums

2.4 Rust Enum Implementationsbeispiele

2.5 Beispielfunktionalität in Java

2.5.1 Expression-Logik in Java

2.5.2 Java Enums am Limit - Idee einer Wrapperinstanz für den Typ

- Idee, was aber wenn die Instanz ein Wrapper ist
- nicht sehr ergonomisch
- statische variablen schneiden uns

output

3 Traits

3.1 Allgemeines zu Traits

3.1.1 Traits sind keine Typen

3.2 Traits in Rust

3.2.1 Default-Implementationen

3.2.2 Trait Bounds

Mehrfaches Traitbinding

Konditionelle Implementierung mit Trait Bounds

3.2.3 Dynamische Traits

3.2.4 Traits als Parameter

wie interfaces

3.2.5 Supertraits

3.2.6 Referenzierung des eigenen Typen

3.2.7 Spezifizierung von Platzhaltertypen

3.2.8 Shorthand Schreibweise

3.2.9 Schreibweise bei Uneindeutigkeit

3.3 Rust Trait Beispiele

3.4 Beispielfunktionalität in Java

4 Vergleich der beiden Ansätze

Literatur