

Bachelorarbeit

Ein Computeralgebrasystem in Rust



Verfasser	Bernd Haßfurther <nergloM@posteo.de>
Matrikel-Nr.	4372280
Betreuerin	Prof. Dr. Lena Oden
Datum	28. August 2022

Inhaltsverzeichnis

1. Vorstellung Computeralgebrasystem
2. Vorstellung Rust
3. Einen Term lesen
4. Implementierung des CAS
5. Vergleich zu SymPy
6. Zusammenfassung und Fazit
7. Quellen

Was ist ein CAS?

► Performance

Was ist ein CAS?

- ▶ Performance
- ▶ Verlässlichkeit

Was ist ein CAS?

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Funktionsumfang der Implementierung

► Performance

Funktionsumfang der Implementierung

- ▶ Performance
- ▶ Verlässlichkeit

Funktionsumfang der Implementierung

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Ziele von Rust

► Performance

Ziele von Rust

- ▶ Performance
- ▶ Verlässlichkeit

Ziele von Rust

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Syntax anhand von Beispielen

► Performance

Syntax anhand von Beispielen

- ▶ Performance
- ▶ Verlässlichkeit

Syntax anhand von Beispielen

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Weitere Konzepte in Rust

► Performance

Weitere Konzepte in Rust

- ▶ Performance
- ▶ Verlässlichkeit

Weitere Konzepte in Rust

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Vertiefung des Ownership und Borrowing

► Performance

Vertiefung des Ownership und Borrowing

- ▶ Performance
- ▶ Verlässlichkeit

Vertiefung des Ownership und Borrowing

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Stack, Heap, Copy und Clone

► Performance

Stack, Heap, Copy und Clone

- ▶ Performance
- ▶ Verlässlichkeit

Stack, Heap, Copy und Clone

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Tokenizer

► Performance

Tokenizer

- ▶ Performance
- ▶ Verlässlichkeit

Tokenizer

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Parser

► Performance

Parser

- ▶ Performance
- ▶ Verlässlichkeit

Parser

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Überlauf und Ungenauigkeit

► Performance

Überlauf und Ungenauigkeit

- ▶ Performance
- ▶ Verlässlichkeit

Überlauf und Ungenauigkeit

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Überlegungen zur Datenstruktur

► Performance

Überlegungen zur Datenstruktur

- ▶ Performance
- ▶ Verlässlichkeit

Überlegungen zur Datenstruktur

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Vergleich zu existierenden Lösungen

► Performance

Vergleich zu existierenden Lösungen

- ▶ Performance
- ▶ Verlässlichkeit

Vergleich zu existierenden Lösungen

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Datenstruktur in Rust

► Performance

Datenstruktur in Rust

- ▶ Performance
- ▶ Verlässlichkeit

Datenstruktur in Rust

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Grundfunktionalitäten

► Performance

Grundfunktionalitäten

- ▶ Performance
- ▶ Verlässlichkeit

Grundfunktionalitäten

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Erweiterung des CAS mit EvalFn

► Performance

Erweiterung des CAS mit EvalFn

- ▶ Performance
- ▶ Verlässlichkeit

Erweiterung des CAS mit EvalFn

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Konkrete Erweiterungen des CAS

► Performance

Konkrete Erweiterungen des CAS

- ▶ Performance
- ▶ Verlässlichkeit

Konkrete Erweiterungen des CAS

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Implementierung von mathematischen Funktionen

► Performance

Implementierung von mathematischen Funktionen

- ▶ Performance
- ▶ Verlässlichkeit

Implementierung von mathematischen Funktionen

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Parsen von Termen

► Performance

Parsen von Termen

- ▶ Performance
- ▶ Verlässlichkeit

Parsen von Termen

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Performance

► Performance

Performance

- ▶ Performance
- ▶ Verlässlichkeit

Performance

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Verbesserungsideen und deren Ansätze

► Performance

Verbesserungsideen und deren Ansätze

- ▶ Performance
- ▶ Verlässlichkeit

Verbesserungsideen und deren Ansätze

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

Vor- und Nachteile Rust

► Performance

Vor- und Nachteile Rust

- ▶ Performance
- ▶ Verlässlichkeit

Vor- und Nachteile Rust

- ▶ Performance
- ▶ Verlässlichkeit
- ▶ Produktivität

(vgl. [1])

[1] Rust Website: *Why Rust?*
<https://www.rust-lang.org/>.
[abgerufen am 17.08.2021].