Programmiersprachen im Vergleich

Artur Papoyan

November 26, 2025

Abstract

1 Introduction

- Motivation
- Overview of the paper structure

2 Rust Highlights

2.1 Origin and Design Philosophy

- History and Background
- Goal Memory Safety without Garbage Collector
- Role of the Rust Foundation

2.2 Ownership and Borrowing

- Ownership model: who owns what?
- Borrowing: shared and exclusive references
- Borrow Checker and lifetimes

2.3 Memory Management Without Garbage Collector

- Automatic deallocation via scope-based drop
- Comparison to Garbage Collector

2.4 Concurrency and Synchronization

- Fearless Concurrency
- Safety + Performance
- Zero-Cost Abstractions
- Send/Sync
- Threads, Channels
- async/await + runtime ecosystem

2.5 Type System and Language Characteristics

- Static typing
- Traits and Generics
- Pattern Matching
- Error handling: Result, Option

2.6 Tooling and Ecosystem

- Cargo (Package Manager and Build System)
- Crates.io (Ecosystem)
- rustup and Toolchains
- rustdoc, Testing, Benchmarking

2.7 Current Developments

2.8 Comparison with Other Languages

- Rust vs. C++

3 Practical Section

4 Conclusion