

Überblick

TinyPL0 ist eine C#-Portierung des historischen PL/0-Compilers inklusive Virtual Machine (P-Code). Das Projekt dient als Ausbildungs- und Lehrbeispiel und legt besonderen Wert auf Nachvollziehbarkeit und vollständige Dokumentation.

Zielgruppe

- Auszubildende und Studierende, die Compiler- und VM-Grundlagen lernen.
- Lehrende, die Beispiele für PL/0 und P-Code benötigen.
- Entwickler, die sich mit der Portierung von Pascal nach C# befassen.

Struktur der Dokumentation

- Einstieg und Nutzung der CLI.
- Architektur und Projektaufteilung.
- PL0-Handbuch mit Lernstufen.
- P-Code-Handbuch und P-Code-Tutorial.
- API-Referenz.
- Anhang mit Beispielprogrammen.

Architektur und Projektuebersicht

TinyPI0 ist in drei Projekte gegliedert:

- PI0.CLI: Kommandozeilenwerkzeug fuer Kompilierung und Ausfuehrung.
- PI0.Core: Lexer, Parser, Compiler und P-Code-Serialisierung.
- PI0.Vm: P-Code-Interpreter inkl. I/O-Adapter.

Die detaillierte Architektur sowie weitere Referenzen werden im Kapitel "Inhalte aus docs/" kuratiert aufgearbeitet.

Nutzung der CLI

Commands

- `compile`: PL/0 nach P-Code kompilieren.
- `run`: PL/0 direkt ausfuehren.
- `run-pcode`: P-Code-Datei ausfuehren.

Typische Optionen

- `--out`: Zielpfad fuer P-Code.
- `--list-code`: P-Code-Liste ausgeben.
- `--emit`: Emissionsmodus fuer Diagnosen.

Beispiele

```
dotnet run --project src/Pl0.Cli -- run  
tests/data/pl0/valid/feature_io_q_bang_relops.pl0
```

PL0-Handbuch

Dieses Handbuch erklaert die Sprache PL/0 und bietet pro Sprachelement Beispiele in drei Lernstufen: kurz, mittel und ausfuehrlich.

Hinweis: TinyPL0 unterstuetzt die Dialekte **classic** und **extended**. Der **extended** Dialekt umfasst ? (Eingabe) und ! (Ausgabe).

Inhalte

- Sprachelemente (Konstanten, Variablen, Prozeduren, Bedingungen, Schleifen, Ein-/Ausgabe)
- Syntax-Referenz
- Best Practices und typische Fehler

P-Code-Handbuch

Dieses Kapitel erklart den P-Code der PL/0-VM und bietet Beispiele je Instruktion. Die detaillierte Instruktionsliste wird im kuratierten Kapitel aus [docs/VM_INSTRUCTION_SET.md](#) aufbereitet.

Register und Speicher

- P: Program Counter
- B: Base Pointer
- T: Top-of-Stack
- I: Instruction Register

Die VM arbeitet mit einem Stack und einer separaten Code-Area.

Tutorial: Direkte P-Code-Programmierung

Dieses Tutorial fuehrt Auszubildende in die direkte P-Code-Programmierung ein. Es baut schrittweise von einfachen Instruktionen bis zu kleinen Programmen auf.

Voraussetzung: Grundkenntnisse zu Stack und einfachen Kontrollstrukturen.

API-Referenz

Die API-Referenz wird aus XML-Kommentaren generiert. Fuer Schulungszwecke sind auch interne und private Member enthalten. Die Zugriffsebene wird in der DocFX-Ansicht entsprechend gekennzeichnet.

Anhang: Beispielprogramme

Die folgenden Programme sind nach Themen gruppiert.

Zahlentheorie und Zahlenfolgen

- [Primzahltest](#)
- [Primzahlen bis N](#)
- [Fakultaet](#)
- [Potenz](#)
- [Fibonacci](#)
- [KGT](#)
- [GGT](#)
- [Summe 1 bis N](#)
- [Summe gerade/ungerade](#)
- [Arithmetische Folge](#)
- [Geometrische Folge](#)
- [Quadratzahlen](#)
- [Kubikzahlen](#)

Grundrechenarten und Operatoren

- [Ganzzahl-Division mit Rest](#)
- [Multiplikation durch Addition](#)
- [Division durch Subtraktion](#)
- [Modulo durch Subtraktion](#)
- [Summenquadrat vs. Quadratsumme](#)

Vergleich und Logik

- [Zahlenvergleich](#)
- [Paritaetstest](#)
- [Betrag](#)
- [Abstand zweier Zahlen](#)
- [Swap](#)

Ziffern und Darstellungen

- [Ziffernsumme](#)
- [Zifferanzahl](#)
- [Palindromtest](#)
- [Umkehrung einer Zahl](#)

Geometrie und Physik

- [Rechteck Umfang und Flaeche](#)
- [Quadrat Umfang und Flaeche](#)
- [Kreis Umfang und Flaeche](#)
- [Dreiecksumfang](#)
- [Pythagoras](#)
- [Geschwindigkeit](#)

Mathe-Funktionen (Festkomma)

- [Mathematische Funktionen](#)

Statistik und Auswertung

- [Mittelwert](#)
- [Minimum und Maximum](#)
- [Einfache Statistik](#)

Alltag und Wirtschaft

- [Dreisatz](#)
- [Prozentrechnung](#)
- [Zins \(einfach\)](#)
- [Zinseszins](#)
- [Rabattberechnung](#)
- [Versandkostenstaffel](#)
- [Waehrungsumrechnung](#)
- [Temperaturumrechnung](#)
- [Einheitenumrechnung](#)
- [Wechselgeld](#)
- [Notenbewertung](#)
- [BMI](#)

Zeit und Kalender

- [Zeitumrechnung](#)
- [Uhrzeitdifferenz](#)
- [Tage zu Wochen](#)
- [Altersberechnung](#)

Gleichungen und Interpolation

- [Lineare Gleichung](#)

- [Quadratische Gleichung](#)
- [Lineare Interpolation](#)

Sonstiges

- [Countdown](#)
- [Schleifenzaehler](#)

Schwierigkeit und Zeit

Legende: E = Einfach, M = Mittel, A = Anspruchsvoll.

Programm	Stufe	Zeit
Summe 1 bis N	E	10-15 min
Zahlenvergleich	E	10-15 min
Paritaetstest	E	10-15 min
Betrag	E	5-10 min
Abstand zweier Zahlen	E	5-10 min
Countdown	E	10-15 min
Arithmetische Folge	E	10-15 min
Geometrische Folge	M	15-20 min
Fibonacci	M	20-25 min
GGT	M	20-25 min
KGT	M	20-25 min
Primzahltest	M	20-25 min
Primzahlen bis N	A	25-35 min
Prozentrechnung	E	10-15 min
Zins (einfach)	E	10-15 min
Zinseszins	M	15-20 min

Programm	Stufe	Zeit
Rechteck Umfang und Flaeche	E	10-15 min
Kreis Umfang und Flaeche	M	15-20 min
Lineare Gleichung	M	15-20 min
Quadratische Gleichung	A	25-35 min
Ziffernsumme	M	15-20 min
Palindromtest	A	25-35 min

Kuratierte Inhalte aus docs/

Die folgenden Kapitel integrieren die bestehenden Inhalte aus dem Ordner [docs/](#) und ordnen sie in die Gesamtdokumentation ein.

- Architektur
- Sprachumfang (EBNF)
- VM-Instruction-Set
- Traceability-Matrix
- Qualitaet