

# 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.

# Schnellstart

## Voraussetzungen

- .NET SDK 10

## Build und Tests

```
dotnet build
```

```
dotnet test
```

## CLI-Beispiele

```
# PL/0 Quelltext ausfuehren
dotnet run --project src/Pl0.Cli -- run
tests/data/pl0/valid/feature_io_q_bang_relops.pl0
```

```
# Kompilieren nach .pcode
dotnet run --project src/Pl0.Cli -- compile
tests/data/pl0/valid/feature_const_var_assignment.pl0 --out /tmp/example.pcode
```

# Installation und Nutzung

## Installation

1. .NET SDK 10 installieren.
2. Repository klonen.
3. `dotnet build` ausfuehren.

## Nutzung

- CLI: `dotnet run --project src/Pl0.Cli -- <command> <datei.pl0>`
- Unterstuetzte Commands: `compile`, `run`, `run-pcode`

## Beispiel

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

# Lizenz

Dieses Projekt steht unter der MIT-Lizenz. Details stehen in der Datei [LICENSE](#) im Repository.

# Support und Feedback

Bitte fuer Support und Feedback ein Issue in diesem GitHub-Repository erstellen.