

Milestone 1: C++ Compiler

- Organisation
- AST-Schnittstelle
- Pipeline-Schnittstellen
- Frontend
 - Parser
 - Testcases
- Backend
- Texteditor

Organisation

- Aufteilung in drei Untergruppen
 - Lexer/Parser
 - Backend
 - Texteditor
- Lexer/Parser sowie Backend haben gemeinsam am AST gearbeitet.
- Aufteilung siehe Organigramm

AST-Schnittstelle

- Eigentlich ASG Schnittstelle
 - Graph in Textdatei als Adjazensliste
 - Mit Haskell Gruppe abgesprochen.
 - Sollte austauschbar sein und Cross-testing ermöglichen (noch nicht getestet)
- Ausschnitt aus einer Beispieldatei auf der nächsten Folie

AST-Schnittstelle

```
49 [main]
50 1; [HelloWorld]; 2; 0
51 2; o; 3; 0
52 3; #; 0; 0
```

Frontend:Parser

- Das Frontend hat umgesetzt:
 - Datei einlesen
 - Funktionen parsen:
 - Rails ablaufen
 - Richtung korrigieren
 - Befehle parsen
 - AST erstellen
 - Den AST serialisieren/deserialisieren

Frontend: Testcases

- Das Frontend hat verschiedene Testcases definiert:
 - Digits.txt – Ast wird korrekt erzeugt
 - Func.txt – Ast wird korrekt erzeugt
 - Helloworld.txt – Ast wird korrekt erzeugt
 - hw_braces.txt – Ast wird korrekt erzeugt
 - Crash.txt – korrekte Fehlermeldung
 - Empty.txt – korrekte Fehlermeldung
 - hw_braces2.txt – korrekte Fehlermeldung
 - Hello-world-kompliziert.txt – fälschlicherweise Fehlermeldung