

It took me about 2 weeks to finally accept the current program structure and working on it.

A program is made of many ProgramComponent, where each of them contains many children ProgramComponent of arbitrary types. Each ProgramComponent has 2 operations: Parse and Build. “Parse” operation parses a file and see if it satisfies KPL’s BNF specification. “Build” operation analyze the program syntactically.

This is the screenshots of example3.kpl parsing result. Only the first and end part is captured.

First part:

```
rev@rev:~/Memory/Knowledge/Compiler/Lab/Parser/cpp$ ./compiler test/example3.kpl
Parsing Program
>>Program
>>Identifier
>>Semicolon
Parse Declarations (Block)
Parsing Const Declaration
Const Declaration Parsed
Parsing Types Declaration.
Type Declaration Parsed
Parse Var Declaration.
>>Var
>>Identifier
>>Colon
Parsing Type
>>Integer
Type Parsed
>>Semicolon
>>Identifier
>>Colon
Parsing Type
>>Integer
Type Parsed
>>Semicolon
>>Identifier
>>Colon
Parsing Type
```

End part:

```
>>ParenthesisClose  
Call Parsed  
Statement Parsed  
>>Semicolon  
Parsing Statement  
Parsing Call  
>>Call  
>>Identifier  
Call Parsed  
Statement Parsed  
>>End  
Begin-End Parsed  
Statement Parsed  
For Parsed  
Statement Parsed  
>>End  
Begin-End Parsed  
Declarations (Block) Parsed  
>>Period  
Program Parsed
```

You can see the full debugging messages by compiling the program (I have a Makefile for this project. Maybe DevC++ can use it?)

Here's the parsing result of example5.kpl:

First part:

```
rev@rev:~/Memory/Knowledge/Compiler/Lab/Parser/cpp$ ./compiler test/example5.kpl  
Parsing Program  
>>Program  
>>Identifier  
>>Semicolon  
Parse Declarations (Block)  
Parsing Const Declaration  
Const Declaration Parsed  
Parsing Types Declaration.  
Type Declaration Parsed  
Parse Var Declaration.  
>>Var  
>>Identifier  
>>Colon  
Parsing Type  
>>Integer  
Type Parsed  
>>Semicolon  
>>Identifier  
>>Colon  
Parsing Type  
>>Integer  
Type Parsed  
>>Semicolon  
>>Identifier
```

End part:

```
Statement Parsed
For Parsed
Statement Parsed
>>Semicolon
Parsing Statement
Parsing Call
>>Call
>>Identifier
>>ParenthesisOpen
Parsing Expression
Parsing Term
Parsing Factor
>>Identifier
Factor Parsed
Term Parsed
Expression Parsed
>>ParenthesisClose
Call Parsed
Statement Parsed
>>End
Begin-End Parsed
Declarations (Block) Parsed
>>Period
Program Parsed
```

If you have any feedback, I'd love to hear from you :3