

Main.cpp:

构造Driver, 调用parse (), print_errors(), print_ast()

spl_driver.hpp :

成员scanner parser Node root vector<>syntax_errors

spl_driver.cpp:

具体方法实现, 创建scanner, parser,

add_syntax_error添加error

get_errors获取所有error

print_error 输出error

scanner.hpp

定义Info类, 使得scanner.l 中 yyval->build<info> 储存具体值, 位置

定义Error类, Lexical_Error, Syntax_Error

SPL_Scanner 继承yyFlexLexer, 实现yylex () 使得parser调用, add_lexical_error () 添加词法错误。

scanner.l

正则表达式匹配, 额外记录非法token, 以便parser.y常规结束, 不发生error

Parser.y

构建Node树, 处理语法异常, 定义优先级

ast.hpp:

用Node节点统一表示, 多重构造函数, ast树结构为n叉树

```
class Node {
public:
    int line_no = INT32_MAX;
    std::string type;
    std::string value;
    std::vector<Node*> children;
```

ast.cpp

实现print_ast,即n叉树的前序遍历