|  |
| --- |
| 编译原理课程设计  实验报告  **指导教师：刘宇舟**  **年 级： 21**  **班 级：552104**  **小组编号：**  **组长学号姓名：李银**  **组员学号姓名：白鹤**  **组员学号姓名：高玮杰**  **2024 年 06 月 04 日**  **计算机科学与技术学院** |

|  |  |  |
| --- | --- | --- |
| **完成实验内容** | | |
| SNL编译器模块实现：   1. 词法分析器 2. 基于递归下降法实现的语法分析器 3. 基于LL(1)实现的语法分析器 | | |
| **小组成员任务完成情况** | | |
| 姓名 | 具体完成任务 | 工作量  百分比 |
| 李银 | 组织小组协作，实现词法分析器模块 | 40% |
| 白鹤 | 实现基于LL(1)实现的语法分析器 | 33% |
| 高玮杰 | 实现基于递归下降法实现的语法分析器 | 27% |
| **小组成员协作情况** | | |
| 1. 本组使用git进行代码管理和协作，将代码托管在github平台进行组内分享和审核。 2. 李银负责搭建项目架构，引入相关依赖，审核及合并拉取请求； 3. 白鹤负责基于LL1(1)的语法分析器的具体实现； 4. 高玮杰负责基于递归下降法的语法分析器的具体实现。 | | |
| **实验平台与编程语言** | | |
| 实验平台：Windows 10 家庭中文版，Windows 11 家庭中文版  开发工具：IntelliJ IDEA 2024.1.2 (Ultimate Edition)  编程语言：Java 17 | | |
| **实验方案设计** | | |
| 1. 项目整体设计  1. 词法分析器   2.1 词法分析器功能模块   * + 1. 读写器读入源文件的字符流。     2. DFA解析字符流，执行词法检查，最后产生Token序列。     3. 处理器链为DFA根据单词创建Token提供功能支持，通过字符推导相应的字符类型，通过词法类型推导Token类型。     4. 字典为处理器链提供判断的功能支持。     5. 转换器为持久化Token序列提供字段类型转换的功能支持。   1. Token类型设计  1. 语法分析器   3.1 外观设计  3.2 LL1 语法分析器  语法分析器组成  3.2.1 CFG  上下文无关文法  上下文无关文法 静态地定义语法结构。  3.2.2 First Sets  给定符号 X ，First(X) 表示可以从 X 中推导出的所有 terminal 串中 leftmost terminal 的集合  First Sets 使用给定的文法求出。  3.2.3 Follow Sets  给定符号 X ，Follow(X) 表示可以从 Start Symbol S 推导出的所有串中，可能直接接在 X 之后的 terminal 的集合。  Follow Sets 使用 CFG + First Sets 求出。  3.2.4 Predictive Table  存储每一个当前最左端的 non-terminal 以及 lookahead 得到的 next input token 来选取唯一的 production  通过 Follow Sets + First Sets 求得   1. 语法分析--递归下降   4.1递归下降功能模块  4.2功能详述  4.2.1 SyntaxCore包含各个递归下降的函数，而每个递归下降的函数中都确定相邻节点之间的关系，即构建语法关系树  4.2.2 SyntaxImpl中输出打印解析Token流检测到的错误，也输出打印文本或图像的语法树 | | |
| **程序界面及运行截图** | | |
| 1. 解析测试  1. 错误error测试  1. 输出语法树 | | |
| **源程序核心代码** | | |
| 1. 词法分析器模块  1. LL(1)语法分析器   2.1 First Sets  2.1 Follow Sets  2.3 Predict Table   1. 递归下降法的语法分析器   3.1分析函数  3.2递归下降示例函数  3.3输出语法树为文本和图像到文件   1. 系统整体 | | |