**《编译原理》实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2018级计科5班** | | **姓名** | **李林轩** |
| **实验题目** | 语法分析程序的设计与实现 | | | | |
| **实验时间** | **2021/4/24** | | **实验地点** | **DS1421** | |
| **实验成绩** |  | | **实验性质** | **□验证性 □设计性 ■综合性** | |
| 教师评价：  □算法/实验过程正确；□源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | | | |
| 一、实验目的  🡪了解编译器中语法分析环节的作用和意义  🡪掌握语法分析的方法和步骤  🡪熟悉自底向上、自顶向下等各类语法分析方法；掌握语法树的构造方法 | | | | | |
| 二、实验项目内容  一）实验内容  设计并实现实验一中的C语言子集的语法分析程序，要求实现以下功能：   1. 必做项： 2. 可以识别出用C语言子集中数据类型和语句集编写的源程序，并以语法分析树的形式输出分析结果。   （基本数据类型至少包含整数、布尔类型；至少包含赋值语句、变量申明语句、算数加/减法表达式、布尔表达式、循环语句、分支语句）   1. 检查源程序中存在的语法错误，并报告错误所在的位置。 2. 可选项：   需在给定C语言子集的基础上进行扩充,实现下列要求：   1. 能识别函数调用语句 2. 对源程序中出现的错误进行适当的恢复，使得语法分析可以继续进行，检查并报告源程序中存在的所有语法错误及错误位置   二）实验要求：   1. 自由选用程序设计语言（java，python，c/c++）作为实现语言，手工编写语法分析程序。 2. 提交实验报告及源代码。实验报告需严格遵循学校文档规范，内容包含对应文法、语法分析测试用例。   三）C语言子集  数据类型: **int**, 无符号整数, 取值范围0-9999  int a;  int a,b;  int a = 1;  算术运算符: **+**,**-**  a = b + 1;  a = b + c;  赋值运算符: **=**  a = 1;  关系运算符: **==** ,**>**,**<,<>,>=,<=**  a = (b==c);  a = (b>c);  a = (b<c);  逻辑运算符: **&&**, **||, !**  a = (b&&c);  a = (b||c);  a = (!b);  条件语句: **if**  if(a==b)  {  };  循环语句: **while**  while(a==b)  {  };  输入,输出: **get**,**put**  get(a);  put(a);  语句结束符: **;**  **加分项：**  条件语句 **if else**  if(a==b)  {  };  else  {  }; | | | | | |
| 三、实验过程或算法 | | | | | |
| 内容要点，供参考：   1. 语言说明：设计的语法分析器包含C语言哪些子集，可用文法进行描述 2. 选用语法分析方法及相应设计：阐明所选用方法（LL、LR…），根据选定方法对文法所做的改动，或者计算的分析表 3. 数据结构说明 4. 语法树及输出说明 5. 错误处理 6. 程序说明：可选择其中重点对象进行说明，不需要在这里列出全部源代码 7. 测试样例及结果展示：给出能涵盖全部要素的测试样例 8. 语言说明   本语法分析器在实验一词法分析器的基础上，实现了C语言子集的语法分析器。下面对设计的文法进行说明   1. 选用语法分析方法及相应设计 2. 程序数据结构说明 3. 语法树以及输出说明 4. 错误处理 5. 关键程序说明 6. 测试样例以及结果展示 | | | | | |
| 四、实验测试结果  内容要点，供参考：  测试样例及结果展示：给出能涵盖全部要素的测试样例，包括正确源码样例及包含各种错误类型的样例   1. 必选测试样例: 2. 简易计算器   //输入数据num1,num2,op，根据op确定操作进行运算，最后输出运算结果ans  int num1,num2,op,ans;  get(num1,num2,op);  if(op==0)  {  ans = num1 + num2;  };  if(op==1)  {  ans = num1 - num2;  };  if(op==2)  {  ans = num1 & num2;  };  if(op==3)  {  ans = num1 | num2;  };  put(ans);   1. 跑马灯   //循环输入op，改变输出结果out，输入0则结束程序  int num0,num1,out,op;  num1 = 3333;  num2 = 6666;  num3 = 9999;  op = 1;  while(op>0)  {  if(op==1)  {  out = num1;  };  if(op==2)  {  out = num2;  };  if(op==2)  {  out = num3;  };  put(out);  get(op);  };   1. 自定义测试样例(包含自定义的词法错误类型): | | | | | |
| 五、实验总结  内容要点，供参考：  1、实验过程中所遇到的问题及解决办法 | | | | | |