**《编译原理》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | **谭勇** | | | **年级** | **2019级** |
| **学号** | | **20194278** | | | **专业、班级** | **计科4班** |
| **实验名称** | 错误处理作业 | | | | | |
| **实验时间** | **2022．5．7** | | **实验地点** | **DS3401** | | |
| **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； □报告规范；  实验代码得分（满分7分）：  实验报告得分（满分3分）：  评价教师签名： | | | | | | |
| 一、实验目的  1、掌握语法制导定义和语法翻译方案  2、编写语义分析程序，根据给定的文法设计并实现错误处理程序，能诊察出常见的语法和语义错误，进行错误局部化处理，并输出错误信息。 | | | | | | |
| 请根据给定的文法设计并实现错误处理程序，能诊察出常见的语法和语义错误，进行错误局部化处理，并输出错误信息。为了方便自动评测，输入输出及处理要求如下：  （1）输入的被编译源文件统一命名为testfile.txt；错误信息输出到命名为error.txt的结果文件中；  （2）结果文件中包含如下两种信息：错误所在的行号 错误的类别码 （行号与类别码之间只有一个空格，类别码严格按照表格中的小写英文字母）           其中错误类别码按下表中的定义输出，行号从1开始计数：   |  |  |  | | --- | --- | --- | | **错误类型** | **错误类别码** | **解释及举例** | | 非法符号或不符合词法 | a | 例如字符与字符串中出现非法的符号，符号串中无任何符号 | | 名字重定义 | b | 同一个作用域内出现相同的名字（不区分大小写） | | 未定义的名字 | c | 引用未定义的名字 | | 函数参数个数不匹配 | d | 函数调用时实参个数大于或小于形参个数 | | 函数参数类型不匹配 | e | 函数调用时形参为整型，实参为字符型；或形参为字符型，实参为整型 | | 条件判断中出现不合法的类型 | f | 条件判断的左右表达式只能为整型，其中任一表达式为字符型即报错，例如’a’==1 | | 无返回值的函数存在不匹配的return语句 | g | 无返回值的函数中可以没有return语句，也可以有形如return;的语句，若出现了形如return(表达式);或return();的语句均报此错误 | | 有返回值的函数缺少return语句或存在不匹配的return语句 | h | 例如有返回值的函数无任何返回语句；或有形如return;的语句；或有形如return();的语句；或return语句中表达式类型与返回值类型不一致 | | 数组元素的下标只能是整型表达式 | i | 数组元素的下标不能是字符型 | | 不能改变常量的值 | j | 这里的常量指的是声明为const的标识符。例如 const int a=1;在后续代码中如果出现了修改a值的代码，如给a赋值或用scanf获取a的值，则报错。 | | 应为分号 | k | 应该出现分号的地方没有分号，例如int x=1缺少分号 （7种语句末尾，for语句中，常量定义末尾，变量定义末尾） | | 应为右小括号’)’ | l | 应该出现右小括号的地方没有右小括号，例如fun(a,b;，缺少右小括号（有/无参数函数定义，主函数，带括号的表达式，if，while，for，switch，有/无参数函数调用，读、写、return） | | 应为右中括号’]’ | m | 应该出现右中括号的地方没有右中括号，例如int arr[2;缺少右中括号（一维/二维数组变量定义有/无初始化，因子中的一维/二维数组元素，赋值语句中的数组元素） | | 数组初始化个数不匹配 | n | 任一维度的元素个数不匹配，或缺少某一维的元素即报错。  例如int a[2][2]={{1,2,3},{1,2}} | | <常量>类型不一致 | o | 变量定义及初始化和switch语句中的<常量>必须与声明的类型一致。int x=’c’;  int y;  switch(y){  case(‘1’) ….  } | | 缺少缺省语句 | p | switch语句中，缺少<缺省>语句。 |   （3）所有错误都不会出现恶意换行的情况，包括字符、字符串中的换行符、函数调用等等。  （4）其他类型的错误，错误的行号以能够断定发现出错的一个符号的行号为准。例如有返回值的函数缺少返回语句的错误，只有当识别到函数末尾的}时仍未出现返回语句，才可以断定出错，报错行号即为}的行号。    【输入形式】testfile.txt中的存在语法或语义错误的测试程序。  【输出形式】按如上要求将错误处理结果输出至error.txt中。  【评分标准】        本次作业的每个测试程序各包含1-3个错误，均来自上表；若一个测试程序中包含多个错误，准确报出第一个错误能得到60%的分数，其余错误则按实报错误占应报错误的比例得分  【特别提醒】  （1）上表中只列举了部分错误类型和报告该错误类型的情况，未包含的错误类型或错误情况，需要自行设计，本次作业考核不涉及；  （2）完成本次作业时，请勿输出词法分析和语法分析作业要求输出的内容；  （3）本次考核之外，发现错误时最好直接输出描述信息，而不是仅给出错误类别码。  （4）**每一行中最多只有一个错误。**  【参考资料】  根据PASCAL-S文法的定义，阅读编译器源代码，了解符号表的设计实现方案和错误处理实现方案；在此基础上，为自己的编译器添加符号表管理、错误处理功能（包括语法错误和语义错误），编译器源代码见[pascals-compiler.docx](https://course.educg.net/userfiles/file/2021/16157911116670131914646.docx)。  【开发语言及环境】用C/C++实现，平台支持C++11标准，源代码文件必须使用UTF-8编码，才能够输出评测平台能够识别的中文(如果不确定源文件的编码，推荐使用notepad++查看)  【辅助工具】[错误处理测试程序库.zip](https://course.educg.net/userfiles/file/2021/16157911588540220464646.zip)  【提交形式】将所开发的语法分析程序的源文件（.cpp/.c/.h，不含工程文件）打包为zip或rar后提交（注意在MAC下压缩会产生额外的文件到压缩包中，需删掉额外文件后提交），实验报告一起打包提交。 | | | | | | |
| 三、实验过程或算法（源程序）  目标：  编译程序对于语法和语义正确的源程序要正确地编译生成等价的目标代码；而对于错误的源程序不能一发现就停止，而是要能检查出错误的性质和出错位置，并使编译能继续下去，同时尽可能多而准确地发现错误和指出各种错误。  其中的错误可大致划分为：  语法错误：程序结构不符合语法（包括词法）规则的错误。如漏掉分号、小数点写成逗号等错误。  语义错误：程序不符合语义规则或超越具体计算机系统的限制。如标识符先声明再使用、溢出错误等。  本实验建立在词法分析以及语法分析的基础上，需要对其中产生的错误进行错误处理。为了定位错误所在，需要在token中额外储存一个行数。    创建一个结构体和一个vector用来储存错误，最后一并输出。    错误类型简单来说可以分类为语法错误，语义错误，系统限制等。   1. a类错误   在出现char常量和string常量的地方进行检查，判断是否存在非法字符。    构建符号表大致如下：    分别存放变量数，变量名，变量类型，参数表等内容。   1. b类错误定义为重定义类型错误。在定义变量、常量、函数时进行判断，在符号表中进行检索，判断是否重定义。      1. c类类型错误   在使用变量、常量、函数时进行判断，在符号表中进行检索，判断是否已定义。对应代码如下：     1. d类类型错误   d类错误判断参数是否匹配。在调用函数时进行参数表的判断，检查是否参数个数不匹配。     1. e类错误   e类错误判断参数类型是否匹配。     1. f类错误   在进行条件判断时查看前后是否匹配。     1. g类错误   在使用返回语句时进行判断，如果当前函数类型为void，且当前语句不为“return;”，则为g类错误。     1. h类错误   判断在有返回值函数定义结束时中没有出现返回语句，这就是定义的h类错误。     1. i类错误   在使用数组时，方括号内的表达式类型不为int。   1. j类错误   在读语句与赋值语句中，若变量为常量，则为j类错误。   1. k类错误   在应该有分号的语句中进行判断，若没有分号出现的话则定义为k类错误。   1. l类错误   在应该右圆括号的语句（读写语句、返回语句、函数调用、条件语句、循环语句、情况语句、主程序）中进行判断，若没有则为l类错误。     1. m类错误   在应该右方括号（变量定义、因子、赋值语句）的语句中进行判断，若没有则为m类错误。   1. n类错误   若初始化时，数量不相符，即为n类错误。   1. o类错误   变量定义和情况语句中的<常量>必须与声明的类型一致。     1. p类错误   在情况语句结束时，判断是否缺少<缺省>语句。 | | | | | | |
| 四、实验结果及分析和（或）源程序调试过程  平台运行结果（最终得分：8.8分）： | | | | | | |
| 五、实验总结  本次实验中遇到的问题：  1. 实验过程中发现前面写的语法分析程序有不完善的地方，导致出现错误的时候排查半天，找不到错误的原因。  2.在处理函数中重定义时不知道如何解决等等，没有获得满分。  3.g类错误时，没有考虑诸如return;这样的情况，导致在无返回值函数定义中只要出现return就会直接返回错误，在增加判断语句后问题解决。  总结：  通过本次实验，我锻炼了代码编写能力，理解并掌握了语法制导定义和语法翻译的原理以及实现，更深入的了解了编译器的底层实现，对底层的各种数据结构也更加熟悉了，对符号表等结构有了更深入的体会，为代码优化打下了基础。 | | | | | | |