# 课程实验说明书

# 实验计划

课程实践教学16学时。在实践教学方面，主要体现为词法分析程序和语法分析程序的编程实践。本说明书包括某程序设计语言线程规格的词法规则、语法规则和对应的课程设计要求，请根据具体要求，完成课程设计作业。实验教学计划以及学时分配如下表。

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **实验项目** | **学时** | **实验内容和基本要求** |
| 11 | 词法分析设计 | 2 | 画出某一语言词法的状态转换图 |
| 12 | 词法分析程序实现 | 5 | 编程实现某语言的词法分析程序 |
|  | 5月19日之前请提交词法分析程序相关文档和程序 | | |
| 13 | 抽象语法树设计 | 3 | 抽象语法树设计和修改等价文法 |
| 14 | 语法分析程序实现 | 6 | 语法分析程序实现 |
|  | 6月9日之前请提交语法分析程序相关文档和程序 | | |

# 线程规格说明的词法规则和课程设计要求

## 线程规格说明的词法规则

**1）语言的关键字（18个）：**

**thread, features, flows, properties, end, none, in,**  **out,** **data, port, event, parameter, flow , source, sink** , **path, constant** , **access**

**注：关键字是保留字，并且必须是小写。**

**2）语言的专用符号8个：**

**=>** **+=> ； : :: {**  **} ->**

1. **标示符identifier和浮点数decimal的词法规则通过下列正则表达式定义：**

identifier = identifier\_letter (underline?letter\_or\_digit)\*

identifier\_letter=a|..|z|A|..|Z|

letter\_or\_digit = identifier\_letter | digit

digit = 0|..|9

underline=\_

decimal = sign? numeral . numeral

numeral= digit (digit)\*

sign = + | -

**4）空白、换行符和制表符在词法分析时忽略掉。**

## 词法分析程序课程设计要求

词法分析程序的设计与实现需要按照以下要求完成。

1. 基于词法规则设计词法分析器（25分）

画出状态转换图（确定化），并提供必要的文字说明。提交状态转换图.doc

1. 词法分析程序的编程实现（75分）
2. 编程实现词法分析器，提交可执行词法分析程序的源程序

用java（或C++或C语言）手工编程实现词法分析程序，该词法分析程序能够读取测试输入文件中的源程序，并将其词法分析的结果即token序列或词法分析的所有错误信息输出到tokenOut.txt中，以便于检查你的词法分析程序的正确性。输出的错误信息要包括错误所在行数，错误类型等信息（编译原理课程设计文件夹下有测试输入源文件test1.txt、test2.txt、test3.txt）。

1. 提交所生成的词法分析程序的测试方案.doc

注：将状态转换图.doc、源程序、以及词法分析程序测试方案.doc文件放在一个文件夹下，按照“班级\_学号\_姓名”的命名方式打包提交。如果还有其它需要说明的问题须写在readme.doc中。

# 线程规格说明的语法规则和课程设计要求

## 线程规格说明的语法规则

Thread语法规则G[Thread]如下：

Thread -->**thread identifier** [ **features** feature ] [ **flows** flow\_spec ] [ **properties** association**;** ] **end identifier** **;**

feature-->port\_spec|Parameter| **none** **;**

port\_spec --> **identifier** **:** IOtype port\_type [ **{** { association } **}** ] **;**

port\_type -->**data port** [ reference ] | **event data port** [ reference ]| **event port**

Parameter -->**identifier** **:** IOtype **parameter** [ reference ][ **{** { association } **}** ] **;**

IOtype-->**in** | **out** | **in out**

flow\_spec -->flow\_source\_spec| flow\_sink\_spec| flow\_path\_spec| **none;**

flow\_source\_spec -->**identifier :** **flow source** **identifier** [ **{** { association }**}** ] **;**

flow\_sink\_spec -->**identifier :** **flow sink** **identifier**[ **{** { association } **}** ] **;**

flow\_path\_spec -->**identifier** **: flow path** **identifier** **->identifier;**

association -->[ **identifier** **::** ] **identifier** splitter [ **constant** ] **access** **decimal** | **none**

splitter--> **=>** | **+=>**

reference -->[ package\_name **::** ] **identifier**

package\_name --> { **identifier ::** } **identifier**

**（空白、换行符和制表符在语法分析时忽略掉。）**

## 语法分析程序课程设计要求

语法分析程序设计与实现需要按照以下要求完成。

1)为描述语法规则的文法给出相应的抽象语法树结构，提交相应的语法树定 义的文件,语法树的定义尽量简洁易懂，必要时提供说明（25分）。

2)写出识别语法范畴的语法分析程序，提交相应的可执行源程序。该语法分 析程序的具体要求如下（75分）：

(1)该语法分析程序需要读取词法分析的结果即读取文件tokenOut.txt中 的单词序列进行语法分析。

(2) 将tokenOut.txt中的表达式经语法分析后将结果表示为相应的抽象语法树。

(3) 将语法分析的结果打印到文件SyntaxOut.txt中。

(4)给出测试方法描述,提交readme.doc,如果你还有其它需要说明的问题 须写在readme.doc中。

注：将语法树定义的文件、源程序、以及测试方法描述文件放在一个文件夹下，按照“班级\_学号\_姓名”的命名方式打包提交。如果还有其它需要说明的问题须写在readme.doc中。