



华南师范大学

本科学生实验（实践）报告

院 系：计算机学院

实验课程：编译原理

实验项目：C++单词拼装分类器

指导老师：黄煜廉

开课时间：2023 ~ 2024 年度第 2 学期

专 业：计算机科学与技术

班 级：计科 2 班

学 生：林泽勋

学 号：20212821020

华南师范大学教务处

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

一、实验内容

必做内容：

1. 把 C++源代码中的各类单词（记号）进行拼装分类。

C++语言包含了几种类型的单词（记号）：标识符，关键字，数（包括二进制、八进制、十进制、十六进制的整数、浮点数），字符串、注释、特殊符号（分界符）和运算符等【详细的单词类别及拼装规则见详细阅读下面提供的网址中的页面文字说明（参考文献[2][3]）】。

2. 要求应用程序应为 Windows 界面（窗口式图形界面）：使用对话框打开一个 C++源文件，并使用对话框列出所有可以拼装的单词（记号）及其分类。
3. 实验设计实现需要采用的编程语言：C++程序设计语言
4. 根据实验的要求组织必要的测试数据（要能对实验中要求的各类单词都能做完备的测试）
5. 书写完善的软件设计文档。

选做内容：

预编译系统的实现----打造具有个人风格的 XC++语言（单词替换）

1. 描述具有风格的 XC++的单词有哪些，分别对应原 C++的是哪些单词。
2. 实现这个单词替换方案。
3. 需要按上述 1,2 的内容书写相应的设计文档。

二、实验目的

1. 理解编译原理中的词法分析概念，并掌握各种类型的单词（记号）在 C++源代码中的分类和识别规则。
2. 实践应用程序开发，设计一个基于 Windows 界面的工具，允许用户通过对话框打开 C++ 源文件，并对其中的单词进行识别和分类。
3. 掌握使用 C++ 程序设计语言实现词法分析器的技能，包括文本处理、字符串匹配和数据分类等方面。
4. 构建全面的测试数据集，覆盖各种情况下的标识符、关键字、数值、字符串、注释、特殊符号和运算符等，以确保词法分析器的准确性和鲁棒性。
5. 撰写详细的软件设计文档，包括词法分析器的设计思路、实现细节、算法逻辑以及界面交互说明，以便他人理解和复现实验结果。
6. 通过这个实验，我可以深入了解编译原理中词法分析的重要性和实际应

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

用，提升编程能力和对程序语言结构的理解。

三、实验文档：

(一)必做内容：

1. 系统总体结构

系统使用 Qt 框架，C++程序设计语言实现。主要包含五个组件，显示文件的组件，上传 C++文件的按钮，上传 XC++文件的按钮（选做中的内容），分词按钮，显示分词结果的列表组件。利用 Qt 事件 connect 进行绑定实现业务逻辑，完成分词并显示的功能。

2. 分词算法设计

在详细介绍分词算法之前，这里介绍 mapping.json 文件的结构，其中包含一系列键值对，表示某个关键词、运算符、分隔符号及其类别（如：“int”：“KEYWORD”，通过查阅参考文献[2][3]得到），并将 mapping.json 转换为 qt 中的 QHash 类型。在接下来的分析中，我们会给出详细的流程图，并对各部分类型的 token 进行详细讲解，并给出部分较为重要的测试的测试结果。下图即为分词算法的完整流程图：

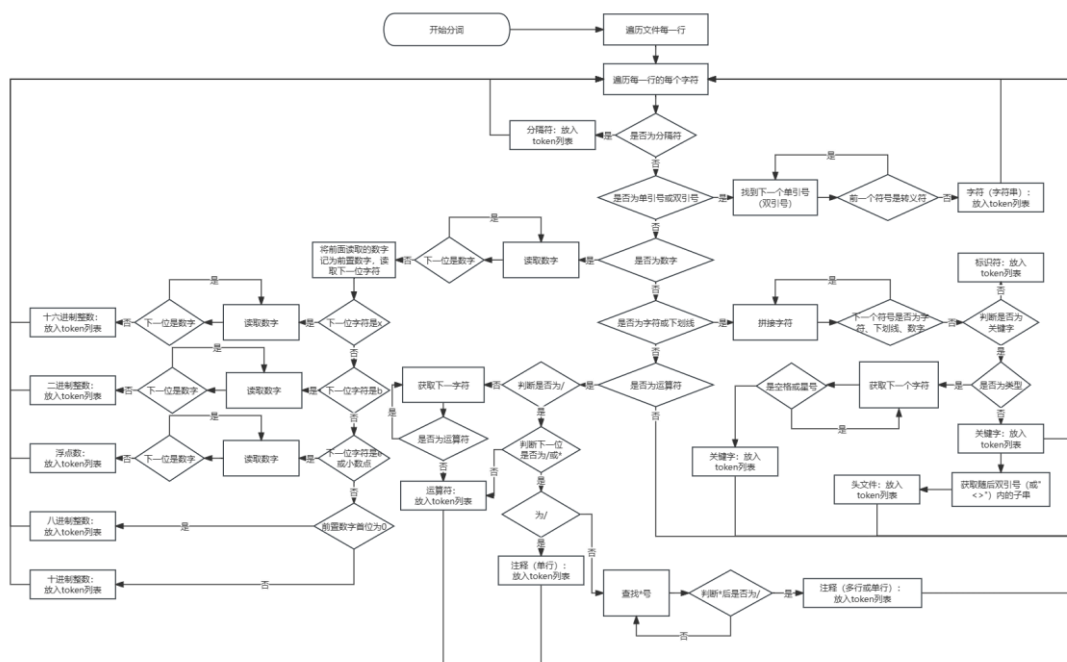


图 1 分词算法流程图

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

详细解释:

1) 判断是否为分隔符:

判断是否在 hash 中出现并判断是否为分隔符, 若均是, 则该字符为标识符。代码与流程图如下:

```
if (hash.contains(line[pos]) && hash[line[pos]] == "SEPARATOR") {  
    // 分隔符  
    tokenList.append(qMakePair(QString(line[pos++]), QString("分界符号")));  
}
```

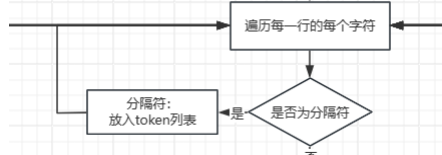


图 2-3 判断分界符号的代码与流程图

2) 判断是否为字符或字符串:

判断当前位置是否为单引号 (双引号), 若是, 则不断获取下一个字符, 直到下一个字符单引号 (双引号), 提取出字符串置入 tokenList。代码与流程图如下:

```
} else if (pos != len && (line[pos] == '"' || line[pos] == '\'')) {  
    // 字符串或字符  
    int endIndex = line.indexOf(line[pos], pos + 1);  
    if (endIndex != -1) {  
        while (endIndex != -1 && line[endIndex - 1] == '\\') endIndex = line.indexOf(line[pos], endIndex + 1);  
        if (endIndex == -1) {  
            break;  
        }  
        QString str = line.mid(pos, endIndex - pos + 1);  
        tokenList.append(qMakePair(str, line[pos] == '"' ? QString("字符串") : QString("字符")));  
        pos = endIndex + 1;  
    } else {  
        break;  
    }  
}
```

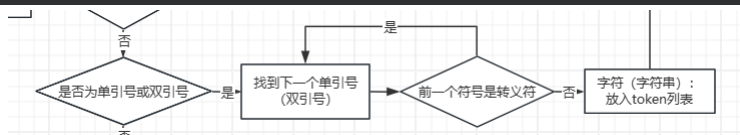


图 4-5 判断字符与字符串的代码与流程图

3) 判断是否为各种数字类型:

先不断获取当前字符, 直到不为数字为止, 取出的字符视为前置数字。判断下一字符是否为 x、b、e、. (这四种字符)。若是这四种字符, 则不断获取后续的数字字符 (满足对应类型: x 需要后续字符为数字或 A-F, b 需要后续字符为 0 或 1, e 和 . 需要后续字符为数字), 得到的完整字符串分别为十六进制整数、二进制整数、浮点数 (科学计数法)、浮点数。若后续字符不为上述的四

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专 业 计算机科学与技术 年级、班级 22 级计科 2 班

课程名称	编译原理	实验项目	C++单词拼装分类器
------	------	------	------------

实验时间 2024 年 3 月 12 日

实验指导老师	黄煜廉	实验评分
--------	-----	------

种字符，则判断前置数字首位是否为 0，若首位为 0，则判断后续是否都是 0-7 的数字，若是，则为八进制整数，否则，为十进制整数。

```

else if (line[pos].isDigit()) {
    // 数字类型
    QString number;
    // 读取前置数字
    while (pos < len && (line[pos].isDigit())) number += line[pos++];

    if (line[pos] == 'x') {
        number += line[pos++];
        while (pos < len && ((line[pos].isDigit() || (line[pos].toLower() >= 'a' && line[pos].toLower() <= 'f')))) number += line[pos++];
        tokenList.append(qMakePair(number, QString("十六进制整数")));
    } else if (line[pos] == 'b') {
        number += line[pos++];
        while (pos < len && (line[pos] == '0' || line[pos] == '1')) number += line[pos++];
        tokenList.append(qMakePair(number, QString("二进制整数")));
    } else if (pos < len && (line[pos].toLower() == 'e' || line[pos] == '.')) {
        number += line[pos++];
        while (pos < len && (line[pos].isDigit())) number += line[pos++];
        tokenList.append(qMakePair(number, QString("浮点数")));
    } else {
        if (number[0] == '0') {
            bool isOctal = true;
            for (int i = 1; i < number.size(); i++) {
                if (number[i] < '0' || number[i] > '8') {
                    isOctal = false;
                    break;
                }
            }
            if (isOctal) {
                tokenList.append(qMakePair(number, QString("八进制整数")));
            } else {
                tokenList.append(qMakePair(number, QString("整数")));
            }
        } else {
            tokenList.append(qMakePair(number, QString("整数")));
        }
    }
}
}

```

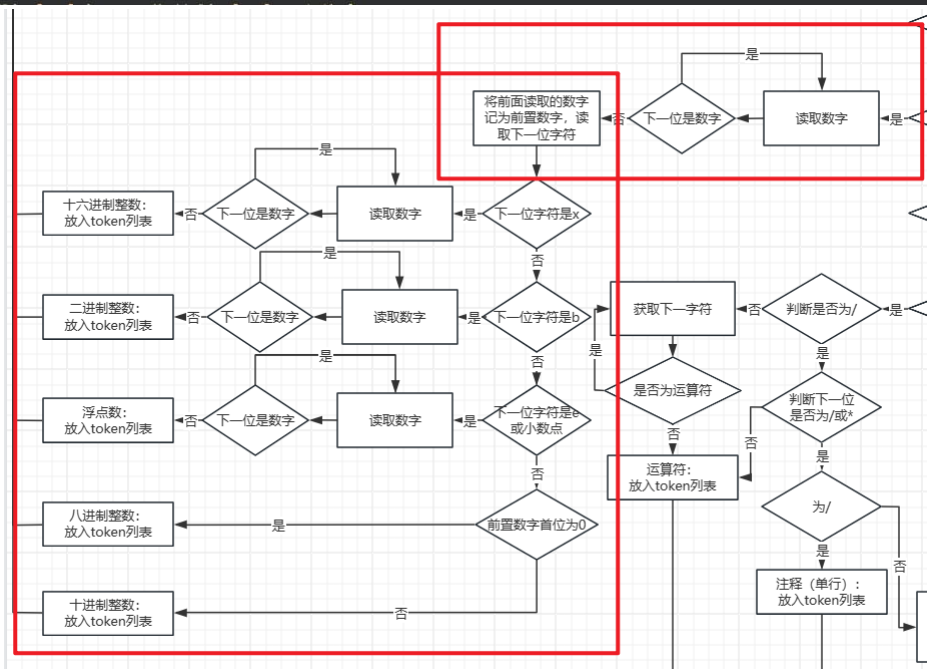


图 6-7 判断各种数字的代码与流程图

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

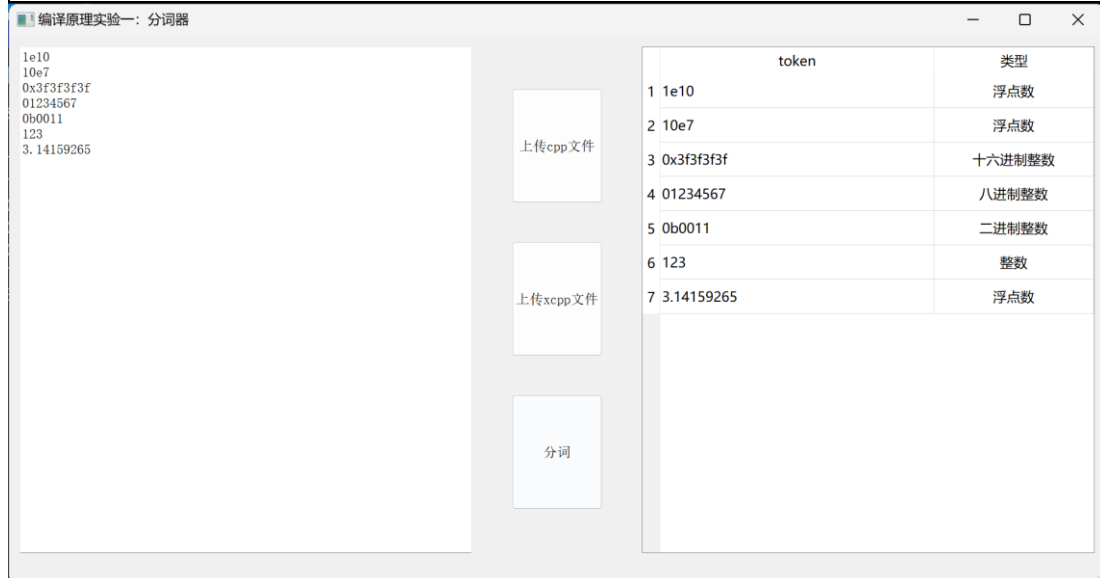
实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

● 测试各类数字：

表 1 各类数字测试数据

1e10
10e7
0x3f3f3f3f
01234567
0b0011
123
3.14159265



	token	类型
1	1e10	浮点数
2	10e7	浮点数
3	0x3f3f3f3f	十六进制整数
4	01234567	八进制整数
5	0b0011	二进制整数
6	123	整数
7	3.14159265	浮点数

图 8 各种数字测试结果

4) 判断是否为标识符或关键词：

若首字符为字母或下划线，则不断获取后续字符（当后续字符为字母、下划线或数字），直到不满足条件，得到一个字符串。判断该字符串是否为关键字（同样利用 hash 判断），若不是，则作为标识符放入 tokenList；若是，则判断是否为类型（如：int、float 等），若是则获取随后的*，以得到指针类型。随后将字符串作为关键字放入 tokenList。判断字符串是否为 include，若是，则获取后续的头文件字符串（如：<xxxx>或"xxxx"）。

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
else if (line[pos].isLetter() || line[pos] == '_') {
    // 标识符, 关键字类型
    QString identifier;
    while (pos < len && ((line[pos].isLetter() || line[pos] == '_' || line[pos].isDigit()))) identifier += line[pos++];

    if (hash.contains(identifier) && hash[identifier] == "KEYWORD") {
        // 捕获指针类型
        if (LexerWindow::isType(identifier)) {
            while (pos < len && (line[pos] == ' ' || line[pos] == "*/")) {
                if (line[pos] == ' ') pos++;
                else identifier += line[pos++];
            }
        }
        tokenList.append(qMakePair(identifier, QString("关键字")));
    }
    if (identifier == "include" || identifier == "import") {
        if (line.indexOf('<', pos) != -1) {
            int beginHeaderIndex = line.indexOf('<', pos);
            int endHeaderIndex = line.indexOf('>', pos);
            tokenList.append(qMakePair(line.mid(beginHeaderIndex, endHeaderIndex - beginHeaderIndex + 1), QString("头文件")));
            pos = endHeaderIndex + 1;
        } else {
            int beginHeaderIndex = line.indexOf('\\', pos);
            int endHeaderIndex = line.indexOf('\\', beginHeaderIndex + 1);
            tokenList.append(qMakePair(line.mid(beginHeaderIndex, endHeaderIndex - beginHeaderIndex + 1), QString("头文件")));
            pos = endHeaderIndex + 1;
        }
    }
} else {
    tokenList.append(qMakePair(identifier, QString("标识符")));
}
```

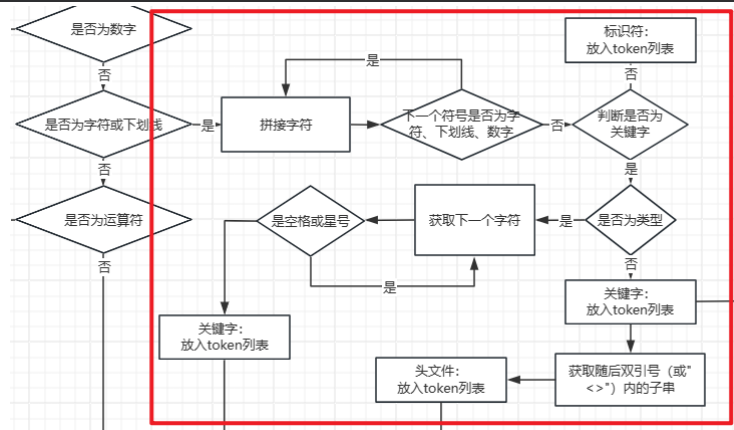


图 9-10 判断标识符、关键字的代码与流程图

● 测试指针变量:

表 2 各类指针变量测试数据

```
int ** pp;
int * p;
void *** vppp;
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020
专 业 计算机科学与技术 年级、班级 22 级计科 2 班
课程名称 编译原理 实验项目 C++单词拼装分类器
实验时间 2024 年 3 月 12 日
实验指导老师 黄煜廉 实验评分

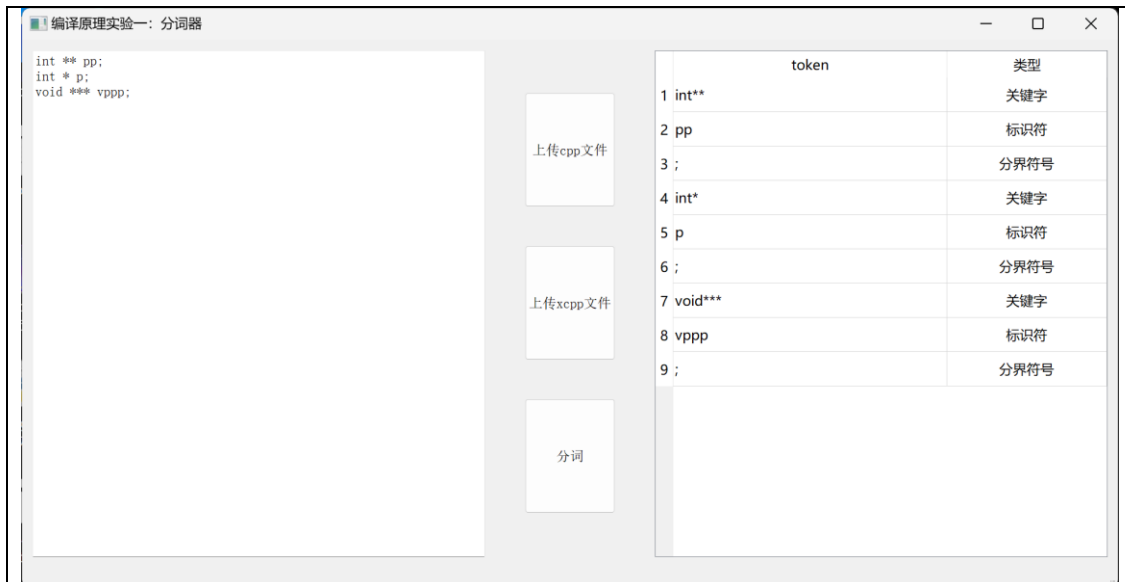


图 11 各种指针变量测试结果

5) 判断是否为运算符或注释:

判断首字符是否为运算符，若是，则判断是否为/，若不是，则表明为普通运算符，接着判断后续是否为运算符，拼接后放入 tokenList 中，若首字符为/，则判断接下来的字符是否为/或*，若是/，则表明为单行注释，该行接下来的所有字符作为注释放入 tokenList 中。若是*，则需要找到当前字符之后的第一个*，判断*之后的字符是否为/（即判断是否注释结束），若不是，则找到下一个*进行上述操作，直到找到该行末尾，若该行找不到*/，则将该行剩下的部分作为注释放入 tokenList，并从下一行的开头开始找*/（找不到就将该行作为注释，找到了就将*/之前的部分作为注释）

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
else if (hash.contains(line[pos]) && hash[line[pos]] == "OPERATOR") {
    // 运算符
    QString op;
    if (line[pos] == '/') {
        // 判断是否为注释
        if (pos != len && (line[pos + 1] == '/' || line[pos + 1] == '*')) {
            if (line[pos + 1] == '/') {
                tokenList.append(qMakePair(line.mid(pos), QString("注释")));
                break;
            } else {
                int endStar = line.indexOf('*', pos + 2);
                // 找第一行
                while (endStar != -1 && endStar < len - 1) {
                    if (line[endStar + 1] == '/') {
                        tokenList.append(qMakePair(line.mid(pos, endStar + 2 - pos), QString("注释")));
                        pos = endStar + 2;
                        break;
                    }
                    endStar = line.indexOf('*', endStar + 1);
                }
                if (endStar == -1 || endStar == len - 1) {
                    tokenList.append(qMakePair(line.mid(pos), QString("注释")));
                    // 接下来继续找
                    while (index + 1 < lines.size()) {
                        bool flag = true;
                        index++;
                        QString newLine = lines[index];
                        len = newLine.size();
                        pos = 0;

                        // 跳过空格与tab
                        while (pos < len && (newLine[pos] == ' ' || newLine[pos] == '\t')) pos++;
                        if (pos == len) continue;

                        endStar = newLine.indexOf('*', pos);
                        while (endStar != -1 && endStar < len - 1) {
                            if (newLine[endStar + 1] == '/') {
                                tokenList.append(qMakePair(newLine.mid(pos, endStar + 2 - pos), QString("注释")));
                                pos = endStar + 2;
                                break;
                            }
                            endStar = newLine.indexOf('*', endStar + 1);
                        }
                        if (endStar == -1 || endStar == len - 1) {
                            tokenList.append(qMakePair(newLine.mid(pos), QString("注释")));
                        }
                        if (!flag) break;
                    }
                }
            }
        }
    }
}
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

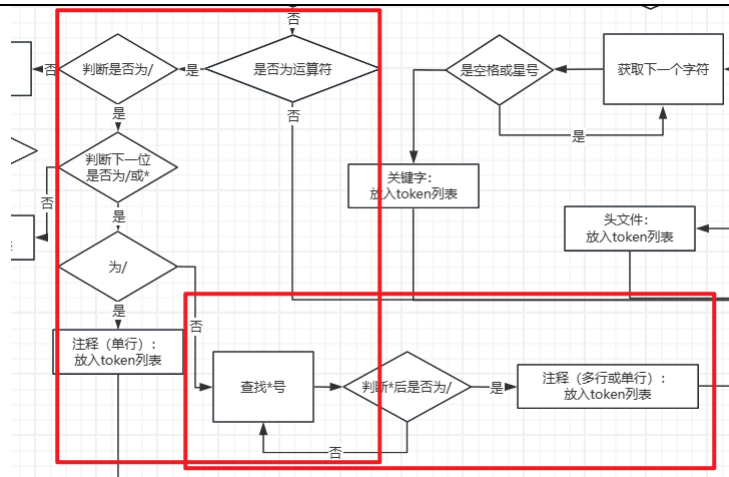


图 12-14 判断运算符、注释的代码与流程图

● 测试多行注释:

表 3 各类注释测试数据

```
/* 第一行多行注释  
第二行多行注释  
第三行多行注释 */
```

```
// 第一行单行注释
```

```
// 第二行单行注释
```



图 15 各种注释测试结果

华南师范大学实验报告

学生姓名 林泽勋 学 号 20212821020
专 业 计算机科学与技术 年级、班级 22 级计科 2 班
课程名称 编译原理 实验项目 C++单词拼装分类器
实验时间 2024 年 3 月 12 日
实验指导老师 黄煜廉 实验评分 _____

3. 完整系统测试

此处使用题目所给的 test.cpp 以及本系统自身的代码（qt 的 main 启动函数、lexerwindow.h、lexerwindow.cpp）进行测试，结果完全正确，测试过程如下所示：（算法设计中对某一功能的单独测试不再赘述）

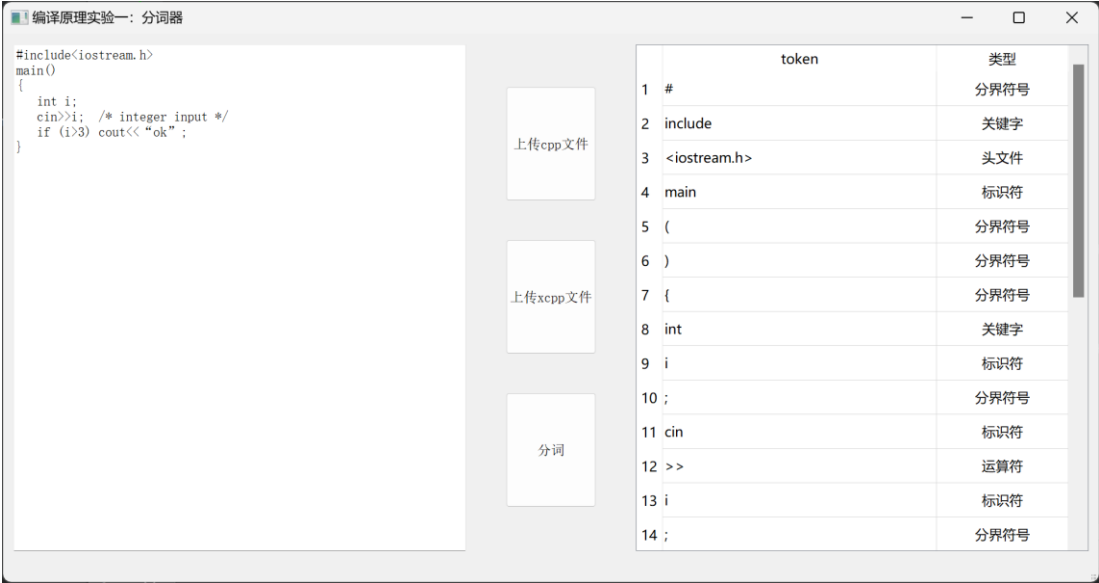


图 16 test.cpp 测试

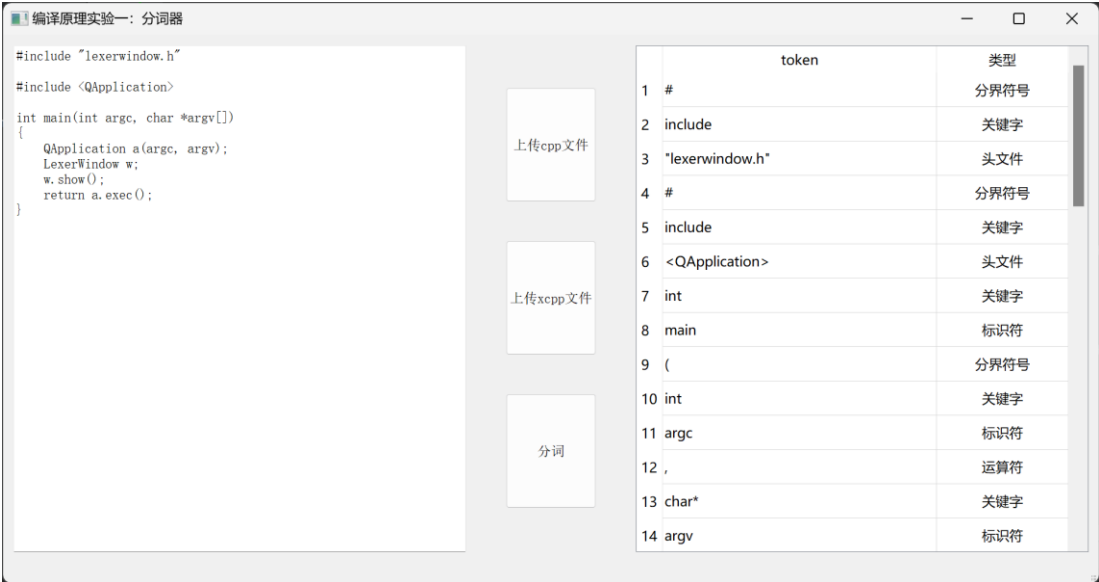


图 17 main.cpp 测试

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020
专业 计算机科学与技术 年级、班级 22级计科2班
课程名称 编译原理 实验项目 C++单词拼装分类器
实验时间 2024 年 3 月 12 日
实验指导老师 黄煜廉 实验评分

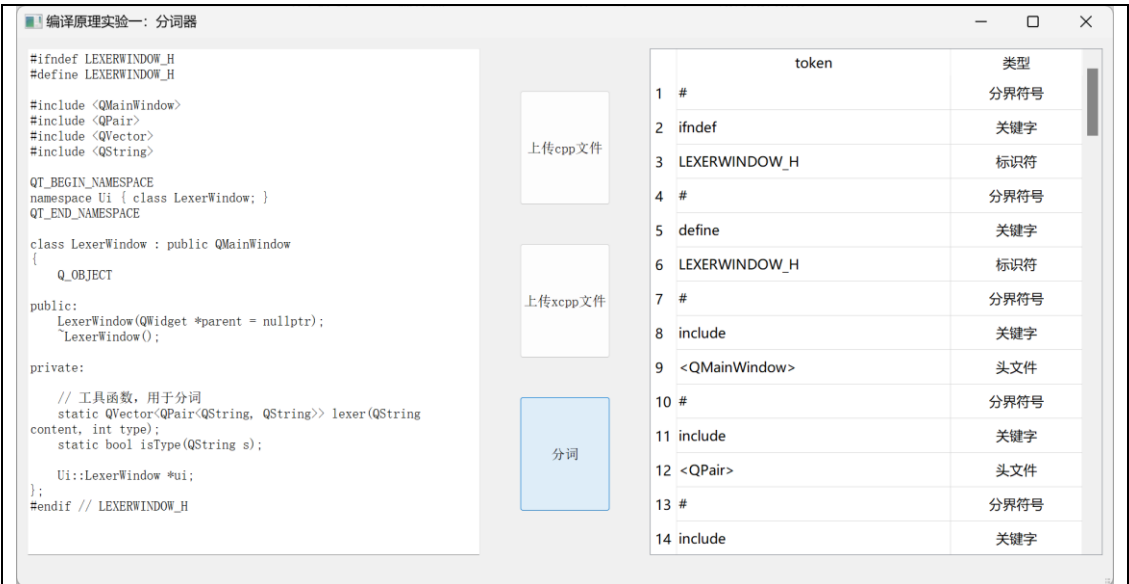


图 18 lexerwindow.h 测试



图 19 lexerwindow.cpp 测试

(二)选做内容：
在此处主要进行关键词的替换，由于避免语义出现问题，不对括号与运算符进行替换。如：将#替换为@，将关键字 include 改为 import 等。具体替换方案在表 4 列出：

表 4 XC++语言单词替换方案

原单词	替换单词
-----	------

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020
专 业 计算机科学与技术 年级、班级 22 级计科 2 班
课程名称 编译原理 实验项目 C++单词拼装分类器
实验时间 2024 年 3 月 12 日
实验指导老师 黄煜廉 实验评分

#	@
int	integer
short	integer
long	integer
char	character
for	foreach
while	foreach
goto	jump
true	True
false	False
auto	any
break	end
continue	restart

在此处我们利用不同的关键词映射文件（mapping.json:标识 C++的关键词、运算符、分隔符；Xmapping.json:标识 XC++的关键词、运算符、分隔符）来进行分词的区别。在加载 C++文件后，分词功能自动调用 mapping.json；加载 XC++文件后，分词功能自动调用 Xmapping.json。具体分词细节与上述必做内容是相似的，仅是单词的替换，此处不再赘述。测试数据与测试结果如下所示：

表 5 XC++测试数据表

```
@import <IO>

integer main() {
    integer a, b;
    a = read();
    b = read();
    foreach (integer i = 1; i <= a + b; i++) {
        if (i == a || i == b) restart;
        if (i == a + b - 1) end;
        write(i);
    }
    return 0;
}
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020
专业 计算机科学与技术 年级、班级 22级计科2班
课程名称 编译原理 实验项目 C++单词拼装分类器
实验时间 2024 年 3 月 12 日
实验指导老师 黄煜廉 实验评分

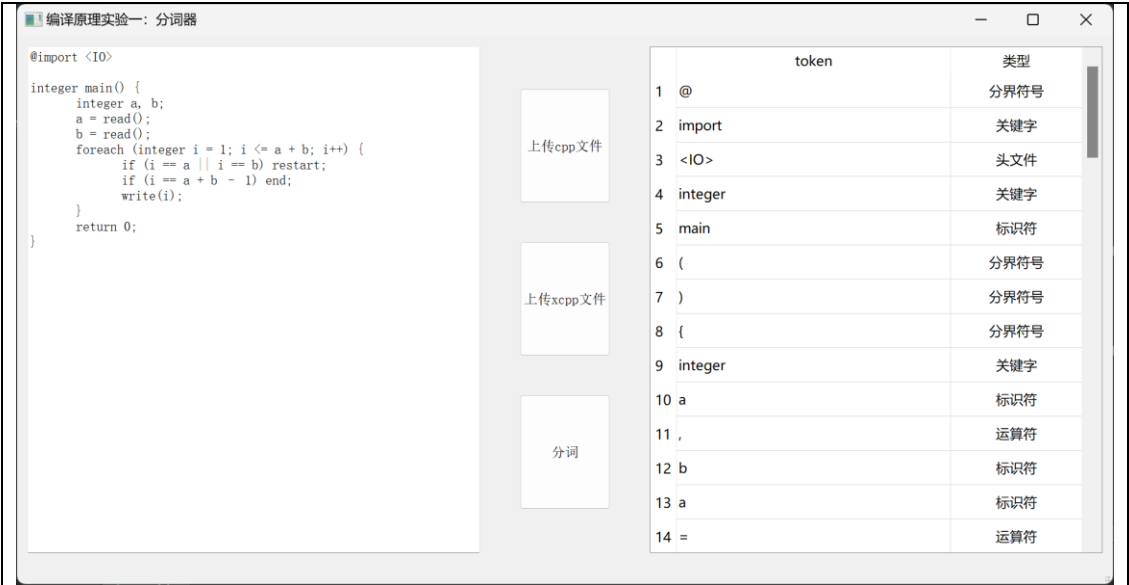


图 20 XC++语言分词测试结果

四、实验总结（心得体会）

在本次编译原理实验中，我们深入学习了词法分析的概念，并掌握了在 C++ 源代码中各种类型单词（token）的分类和识别规则。通过实践应用程序开发，我们成功设计了一个基于 Windows 界面的工具，允许用户通过按钮与文件管理器打开 C++ 源文件，并对其中的单词进行准确的识别和分类。

在实现词法分析器的过程中，我们运用了 C++ 程序设计语言，涉及文本处理、字符串匹配和 token 分类等方面的技能。为了验证词法分析器的准确性和鲁棒性，我们构建了全面的测试数据集，覆盖了各种情况下的标识符、关键字、数值、字符串、注释、特殊符号和运算符。

最终，我们撰写了详尽的软件设计文档，包括词法分析器的设计思路、实现细节、算法逻辑以及界面交互说明，以便他人能够理解和复现我们的实验结果。通过这个实验，我们不仅加深了对编译原理中词法分析重要性的认识，还提升了自身的编程能力，加深了对程序语言结构的理解。

总的来说，本次实验为我们提供了宝贵的学习机会，让我们更加熟练地运用所学知识，并对编译原理有了更深入的理解和实际应用。

五、参考文献：

[1]<https://blog.csdn.net/c369639963/article/details/122683773>
[2]<https://learn.microsoft.com/zh-cn/cpp/cpp/lexical-conventions?view=msvc-170>

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020
专业 计算机科学与技术 年级、班级 22级计科2班
课程名称 编译原理 实验项目 C++单词拼装分类器
实验时间 2024 年 3 月 12 日
实验指导老师 黄煜廉 实验评分 _____

- [3] <https://learn.microsoft.com/zh-cn/cpp/cpp/cpp-built-in-operators-precedence-and-associativity?view=msvc-170>
[4] https://blog.csdn.net/weixin_43544179/article/details/128793314

六、附录：核心代码兼测试数据（头文件与 qt 启动函数不再赘述）

```
1. #include "lexerwindow.h"
2. #include "ui_lexerwindow.h"
3.
4. #include <QScreen>
5. #include <QRect>
6. #include <QFile>
7. #include <QPushButton>
8. #include <QFileDialog>
9. #include <QTextStream>
10. #include <QDebug>
11. #include <QJsonDocument>
12. #include <QJsonObject>
13.
14. LexerWindow::LexerWindow(QWidget *parent)
15.     : QMainWindow(parent)
16.     , ui(new Ui::LexerWindow)
17. {
18.     ui->setupUi(this);
19.
20.     // 基本布局信息
21.     this->setWindowTitle("编译原理实验一：分词器");
22.     QScreen *screen = QGuiApplication::primaryScreen();
23.     QRect screenGeometry = screen->geometry();
24.     int screenWidth = screenGeometry.width();
25.     int screenHeight = screenGeometry.height();
26.     int newWidth = screenWidth * 0.64;
27.     int newHeight = screenHeight * 0.48;
28.     this->resize(newWidth, newHeight);
29.     ui->pushButton->setFixedHeight(200);
30.     ui->xPushButton->setFixedHeight(200);
31.     ui->uploadButton->setFixedHeight(200);
32.
33.     // 设置表格窗口
34.     ui->tableWidget->setColumnCount(2);
35.     ui->tableWidget-
36.         >setHorizontalHeaderLabels(QStringList() << "token" << "类型");
37.     ui->tableWidget->setColumnWidth(0, newWidth / 4);
38.     ui->tableWidget->horizontalHeader()-
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分 _____

```
>setStretchLastSection(true); // 最后一列自适应宽度
38.     ui->tableWidget-
>setEditTriggers(QAbstractItemView::NoEditTriggers);
39.
40.     // 判断分词类型的变量
41.     int type = 1;
42.
43.     // 按钮事件: 打开某文件获取其中的所有文本
44.     connect(ui-
>uploadButton, &QPushButton::clicked, this, [&]() {
45.         type = 1;
46.         QString fileName = QFileDialog::getOpenFileName(this,
"选择 C++文件");
47.         if (!fileName.isEmpty()) {
48.             QFile file(fileName);
49.             if (file.open(QIODevice::ReadOnly | QIODevice::Tex
t)) {
50.                 QTextStream in(&file);
51.                 in.setCodec("UTF-8"); // 设置编码格式为 UTF-8
52.                 ui->fileBrowser->setText(in.readAll());
53.                 file.close();
54.             }
55.         }
56.     });
57.     connect(ui-
>xPushButton, &QPushButton::clicked, this, [&]() {
58.         type = 2;
59.         QString fileName = QFileDialog::getOpenFileName(this,
"选择 XC++文件");
60.         if (!fileName.isEmpty()) {
61.             QFile file(fileName);
62.             if (file.open(QIODevice::ReadOnly | QIODevice::Tex
t)) {
63.                 QTextStream in(&file);
64.                 in.setCodec("UTF-8"); // 设置编码格式为 UTF-8
65.                 ui->fileBrowser->setText(in.readAll());
66.                 file.close();
67.             }
68.         }
69.     });
70.
71.     // 按钮事件: 分词
72.     connect(ui-
>pushButton, &QPushButton::clicked, this, [&]() {
```


华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
73.         QString text = ui->fileBrowser->toPlainText();
74.         QVector<QPair<QString, QString>> list = LexerWindow::l
            exer(text, type);
75.         ui->tableWidget->setRowCount(list.size());
76.         for (int i = 0; i < list.size(); i++) {
77.             ui->tableWidget-
                >setItem(i, 0, new QTableWidgetItem(list[i].first));
78.             ui->tableWidget-
                >setItem(i, 1, new QTableWidgetItem(list[i].second));
79.         }
80.         for (int row = 0; row < ui->tableWidget-
            >rowCount(); ++row) {
81.             QTableWidgetItem *item = ui->tableWidget-
                >item(row, 1);
82.             if (item) item->setTextAlignment(Qt::AlignCenter);
83.         }
84.         ui->tableWidget->show();
85.     });
86.
87. }
88.
89. LexerWindow::~LexerWindow()
90. {
91.     delete ui;
92. }
93.
94. QVector<QPair<QString, QString>> LexerWindow::lexer(QString co
    ntent, int type)
95. {
96.     /*
97.      * 功能：将文本字符串 content 分词
98.      * 类别：关键词、运算符、数字类型、标识符，注释，分界符号
99.      * 返回：键值对序列（词：类型）
100.     */
101.
102.     // 读取 json 文件：关键字，运算符，边界符号等
103.     QString mappingPath;
104.     if (type == 1) {
105.         mappingPath = ":/json/mapping.json";
106.     } else {
107.         mappingPath = ":/json/Xmapping.json";
108.     }
109.
110.     qDebug() << mappingPath << endl;
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
111.
112.     QFile file(mappingPath);
113.     QHash<QString, QString> hash;
114.     if (file.open(QIODevice::ReadOnly | QIODevice::Text)) {
115.         QByteArray data = file.readAll();
116.         QJsonDocument jsonDocument = QJsonDocument::fromJson(d
ata);
117.
118.         if (!jsonDocument.isNull()) {
119.             QJsonObject jsonObject = jsonDocument.object();
120.
121.             // 遍历 JSON 对象的所有键值对, 并添加到哈希表中
122.             for (auto it = jsonObject.begin(); it != jsonObjec
t.end(); ++it) {
123.                 QString key = it.key();
124.                 QString value = it.value().toString();
125.                 hash.insert(key, value);
126.                 qDebug() << key << ' ' << value << endl;
127.             }
128.         }
129.     }
130.
131.     // 开始分词
132.     QVector<QPair<QString, QString>> tokenList(0);
133.     QStringList lines = content.split("\n");
134.     for (int index = 0; index < lines.size(); index++) {
135.         const QString& line = lines[index];
136.         int pos = 0;
137.         int len = line.size();
138.
139.         // 跳过空格与tab
140.         while (pos < len && (line[pos] == ' ' || line[pos] ==
'\t')) pos++;
141.         if (pos == len) continue;
142.
143.         // 单行注释
144.         if (line[pos] == '/' && line[pos + 1] == '/') {
145.             tokenList.append(qMakePair(line.mid(pos), QString(
"注释")));
146.             continue;
147.         }
148.
149.         // 遍历单行
150.         while (pos < len) {
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
151. //          qDebug() << index << ' ' << pos;
152.          if (hash.contains(line[pos]) && hash[line[pos]] ==
    "SEPARATOR") {
153.              // 分隔符
154.              tokenList.append(qMakePair(QString(line[pos++])
    ), QString("分界符号"));
155.          } else if (pos != len && (line[pos] == '\\' || lin
    e[pos] == '\\')) {
156.              // 字符串或字符
157.              int endIndex = line.indexOf(line[pos], pos + 1
    );
158.              if (endIndex != -1) {
159.                  while (endIndex != -1 && line[endIndex -
    1] == '\\') endIndex = line.indexOf(line[pos], endIndex + 1);
160.                  if (endIndex == -1) {
161.                      break;
162.                  }
163.                  QString str = line.mid(pos, endIndex -
    pos + 1);
164.                  tokenList.append(qMakePair(str, line[pos]
    == '\\' ? QString("字符串") : QString("字符")));
165.                  pos = endIndex + 1;
166.              } else {
167.                  break;
168.              }
169.          } else if (line[pos].isDigit()) {
170.              // 数字类型
171.              QString number;
172.              // 读取前置数字
173.              while (pos < len && (line[pos].isDigit())) num
    ber += line[pos++];
174.
175.              if (line[pos] == 'x') {
176.                  number += line[pos++];
177.                  while (pos < len && (line[pos].isDigit() |
    | (line[pos].toLower() >= 'a' && line[pos].toLower() <= 'f'))) nu
    mber += line[pos++];
178.                  tokenList.append(qMakePair(number, QString
    ("十六进制整数")));
179.              } else if (line[pos] == 'b') {
180.                  number += line[pos++];
181.                  while (pos < len && (line[pos] == '0' || l
    ine[pos] == '1')) number += line[pos++];
182.                  tokenList.append(qMakePair(number, QString
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
    ("二进制整数")));
183.         } else if (pos < len && (line[pos].toLower() =
    = 'e' || line[pos] == '.')) {
184.             number += line[pos++];
185.             while (pos < len && (line[pos].isDigit()))
    number += line[pos++];
186.             tokenList.append(qMakePair(number, QString
    ("浮点数")));
187.         } else {
188.             if (number[0] == '0') {
189.                 bool isOctal = true;
190.                 for (int i = 1; i < number.size(); i++)
    {
191.                     if (number[i] < '0' || number[i] >
    '8') {
192.                         isOctal = false;
193.                         break;
194.                     }
195.                 }
196.                 if (isOctal) {
197.                     tokenList.append(qMakePair(number,
    QString("八进制整数")));
198.                 } else {
199.                     tokenList.append(qMakePair(number,
    QString("整数")));
200.                 }
201.             } else {
202.                 tokenList.append(qMakePair(number, QSt
    ring("整数")));
203.             }
204.         }
205.     } else if (line[pos].isLetter() || line[pos] == '_'
    ') {
206.         // 标识符、关键字类型
207.         QString identifier;
208.         while (pos < len && (line[pos].isLetter() || l
    ine[pos] == '_' || line[pos].isDigit())) identifier += line[pos++
    ];
209.
210.         if (hash.contains(identifier) && hash[identifi
    er] == "KEYWORD") {
211.             // 捕获指针类型
212.             if (LexerWindow::isType(identifier)) {
213.                 while (pos < len && (line[pos] == ' '
    ' '))
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
|| line[pos] == "*")) {
214.                                     if (line[pos] == ' ') pos++;
215.                                     else identifier += line[pos++];
216.                                     }
217.                                     }
218.                                     tokenList.append(qMakePair(identifier, QSt
ring("关键字")));
219.                                     if (identifier == "include" || identifier
== "import") {
220.                                     if (line.indexOf('<', pos) != -1) {
221.                                     int beginHeaderIndex = line.indexO
f('<', pos);
222.                                     int endHeaderIndex = line.indexOf(
'>', pos);
223.                                     tokenList.append(qMakePair(line.mi
d(beginHeaderIndex, endHeaderIndex -
beginHeaderIndex + 1), QString("头文件")));
224.                                     pos = endHeaderIndex + 1;
225.                                     } else {
226.                                     int beginHeaderIndex = line.indexO
f('\\', pos);
227.                                     int endHeaderIndex = line.indexOf(
'\\', beginHeaderIndex + 1);
228.                                     tokenList.append(qMakePair(line.mi
d(beginHeaderIndex, endHeaderIndex -
beginHeaderIndex + 1), QString("头文件")));
229.                                     pos = endHeaderIndex + 1;
230.                                     }
231.                                     }
232.                                     } else {
233.                                     tokenList.append(qMakePair(identifier, QSt
ring("标识符")));
234.                                     }
235.                                     } else if (hash.contains(line[pos]) && hash[line[p
os]] == "OPERATOR") {
236.                                     // 运算符
237.                                     QString op;
238.                                     if (line[pos] == '/') {
239.                                     // 判断是否为注释
240.                                     if (pos != len && (line[pos + 1] == '/' ||
line[pos + 1] == '*')) {
241.                                     if (line[pos + 1] == '/') {
242.                                     tokenList.append(qMakePair(line.mi
d(pos), QString("注释")));
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
243.                                     break;
244.                                     } else {
245.                                     int endStar = line.indexOf('*', po
    s + 2);
246.                                     // 找第一行
247.                                     while (endStar != -
    1 && endStar < len - 1) {
248.                                     if (line[endStar + 1] == '/')
    {
249.                                     tokenList.append(qMakePair
    (line.mid(pos, endStar + 2 - pos), QString("注释")));
250.                                     pos = endStar + 2;
251.                                     break;
252.                                     }
253.                                     endStar = line.indexOf('*', en
    dStar + 1);
254.                                     }
255.                                     if (endStar == -
    1 || endStar == len - 1) {
256.                                     tokenList.append(qMakePair(lin
    e.mid(pos), QString("注释")));
257.                                     // 接下来继续找
258.                                     while (index + 1 < lines.size(
    )) {
259.                                     bool flag = true;
260.                                     index++;
261.                                     QString newLine = lines[in
    dex];
262.                                     len = newLine.size();
263.                                     pos = 0;
264.
265.                                     // 跳过空格与tab
266.                                     while (pos < len && (newLi
    ne[pos] == ' ' || newLine[pos] == '\t')) pos++;
267.                                     if (pos == len) continue;
268.
269.                                     endStar = newLine.indexOf(
    '*', pos);
270.                                     while (endStar != -
    1 && endStar < len - 1) {
271.                                     if (newLine[endStar +
    1] == '/') {
272.                                     tokenList.append(q
    MakePair(newLine.mid(pos, endStar + 2 - pos), QString("注释"));
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
273.                                     pos = endStar + 2;
274.                                     flag = false;
275.                                     break;
276.                                     }
277.                                     endStar = newLine.indexOf('*', endStar + 1);
278.                                     }
279.                                     if (endStar == -1 || endStar == len - 1) {
280.                                     tokenList.append(qMakePair(newLine.mid(pos), QString("注释")));
281.                                     }
282.                                     if (!flag) break;
283.                                     }
284.                                     }
285.                                     }
286.                                     } else {
287.                                     op += line[pos++];
288.                                     tokenList.append(qMakePair(op, QString("运算符")));
289.                                     }
290.                                     } else {
291.                                     while (pos < len && (hash.contains(line[pos]) && hash[line[pos]] == "OPERATOR")) op += line[pos++];
292.                                     tokenList.append(qMakePair(op, QString("运算符")));
293.                                     }
294.                                     } else {
295.                                     pos++;
296.                                     }
297.                                     }
298.                                     }
299.
300.         return tokenList;
301.     }
302.
303.     bool LexerWindow::isType(QString s)
304.     {
305.         if (s == "int" || s == "short" || s == "long" || s == "bool" || s == "char" ||
306.             s == "void" || s == "float" || s == "double" || s == "signed" || s == "unsigned" || s == "auto") {
307.             return true;
308.         }
```

华南师范大学实验报告

学生姓名 林泽勋 学号 20212821020

专业 计算机科学与技术 年级、班级 22级计科2班

课程名称 编译原理 实验项目 C++单词拼装分类器

实验时间 2024 年 3 月 12 日

实验指导老师 黄煜廉 实验评分

```
309.     return false;  
310. }
```