****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验九

**姓名：任宇**

**学号：33920212204567**

**学院：信息学院**

**专业：软件工程**

**完成时间：2024-04-24**

1. **实验目的及要求**

* 熟悉字符串及正则表达式

1. **实验题目及实现过程**
2. 题目一：

编写程序完成：

1）输出以下新闻片段中出现的单词（每个单词只输出一次）。

2）输出以下新闻片段中包含the的句子。

1. 实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

1. 实现过程

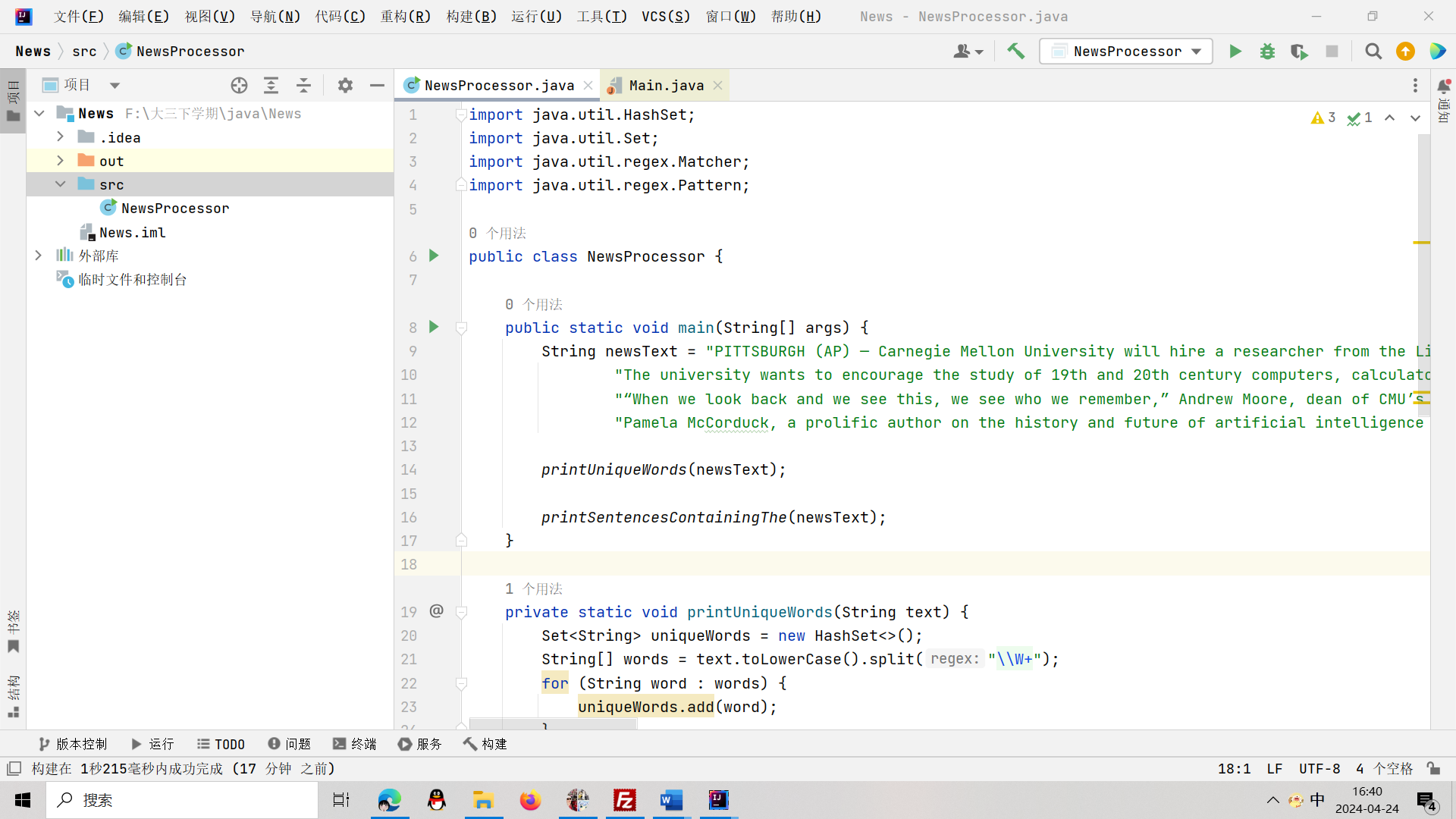
对于本题，只设计一个NewsProcessor类，其中包含两个方法：printUniqueWords（输出新闻中所有单词）和printSentencesContainingThe（输出新闻中包含the的所有句子）。程序实现的重点是相应的正则表达式，分析如下：

* printUniqueWords -------- text.toLowerCase().split("\\W+");

\W是一个预定义的字符类，它匹配任何非字母数字字符，包括下划线在内的字母（a-z和A-Z）、数字（0-9）以外的所有字符。在Java字符串中，反斜杠需要被转义，因此写作\\W。

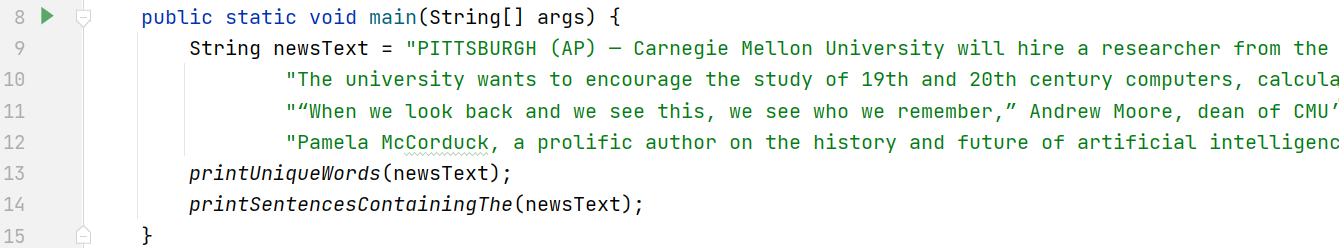
* printSentencesContainingThe ------- Pattern.compile("[^.!?]+(?:[.!?](?!['\"]?\\s|$)[^.!?]\*)\*[.!?]?['\"]?(?=\\s|$)", Pattern.MULTILINE);
* [^.!?]+：这部分匹配一个或多个不是句号（.）、问号（?）、或感叹号（!）的任何字符，会继续匹配直到碰到可能的句子结束符号为止。
* (?:[.!?](?!['\"]?\\s|$)[^.!?]\*)\*：匹配后跟一个不直接结束句子的标点符号，识别中间的缩写和引号内的标点。
* [.!?]?['\"]?(?=\\s|$)：匹配句子的结尾，可能包括一个句号、问号或感叹号，后面可能跟有一个引号。这部分使用?表示前面的字符（标点或引号）是可选的。(?=\\s|$)是一个正向前瞻，确保这个标点后面是空格或字符串的结束。

1. 过程截图
2. 全屏截图：

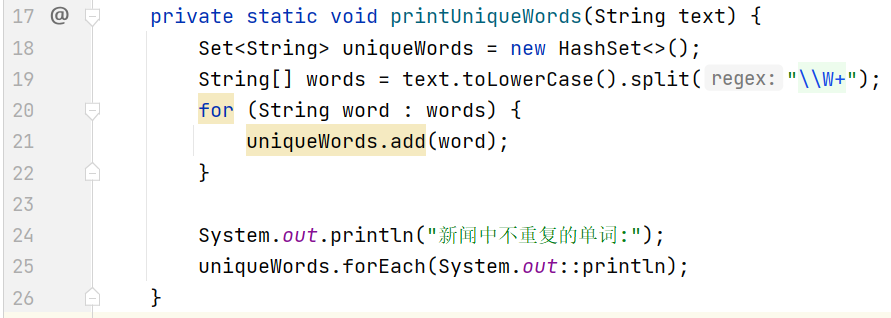


1. 主要代码：

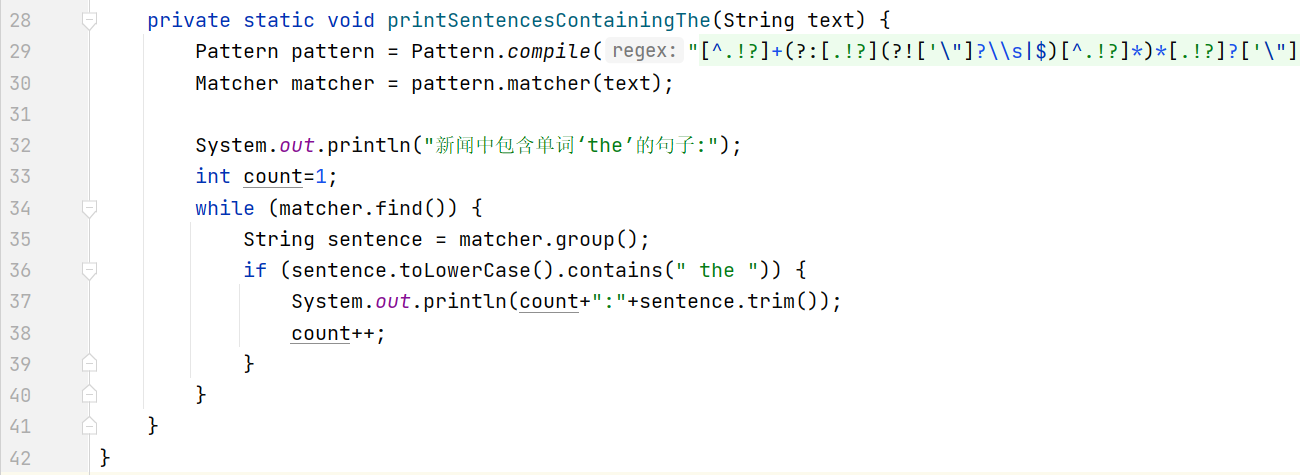
Main方法：



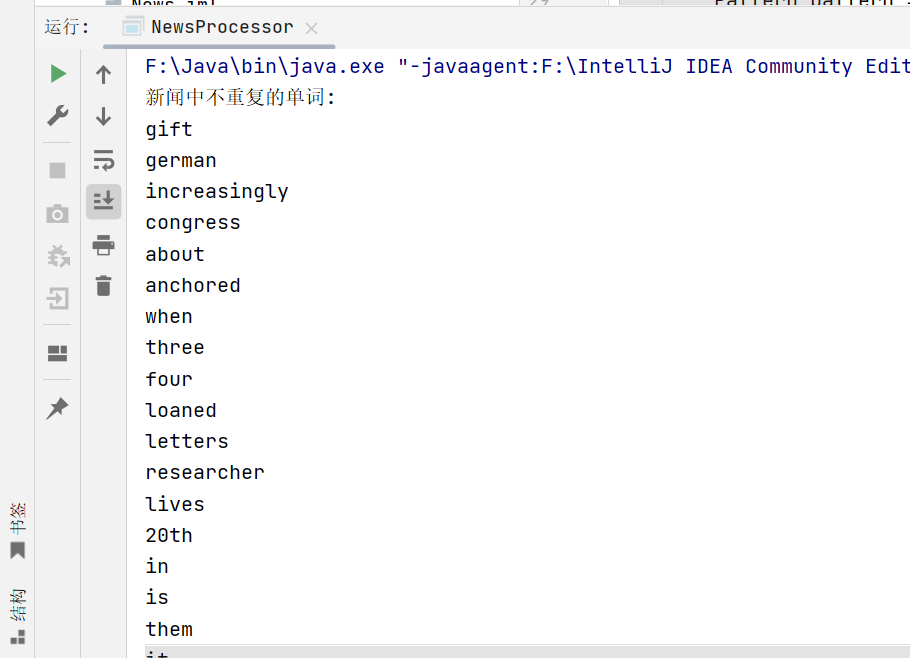
printUniqueWords方法：

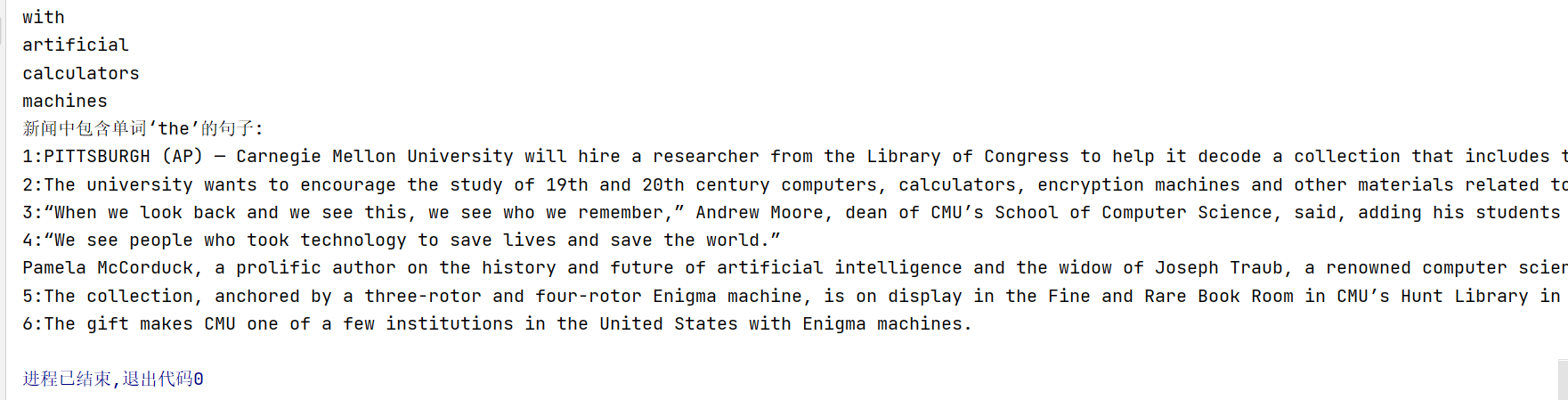


printSentencesContainingThe方法：



1. 运行结果：





1. 题目二：

用正则表达式对用户输入的用户名、密码、邮箱进行判断，若不满足输入要求则提示出错类型。

* 用户名要求：不能为空，只能由字母、数字和\_组成，第一位不能为数字。
* 密码要求：不能为空，密码长度至少8位，由字母、数字、下划线组成。
* 邮箱要求：不能为空，需包含”@”符号。”@”符号后需要出现多个由”. ”分割的词。

1. 实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

1. 实现过程

程序只包含一个Login类，用于处理用户输入的用户名、密码和邮箱的验证。设计validateUsername、validatePassword和validateEmail这三个方法用以处理用户名的验证、密码的验证和邮箱的验证，并且使用 try-catch 结构来处理输入错误，在发现错误时捕捉异常并提示用户重新输入。正则表达式按照题目的要求书写，分别如下所示：

* validateUsername ------ ^[A-Za-z\_][A-Za-z0-9\_]\*$

^匹配输入字符串的开始，[A-Za-z\_]代表首字符必须是字母或下划线，而[A-Za-z0-9\_]\*说明后续字符可以是字母、数字或下划线，\*表示零个或多个。$则匹配输入字符串的结束。

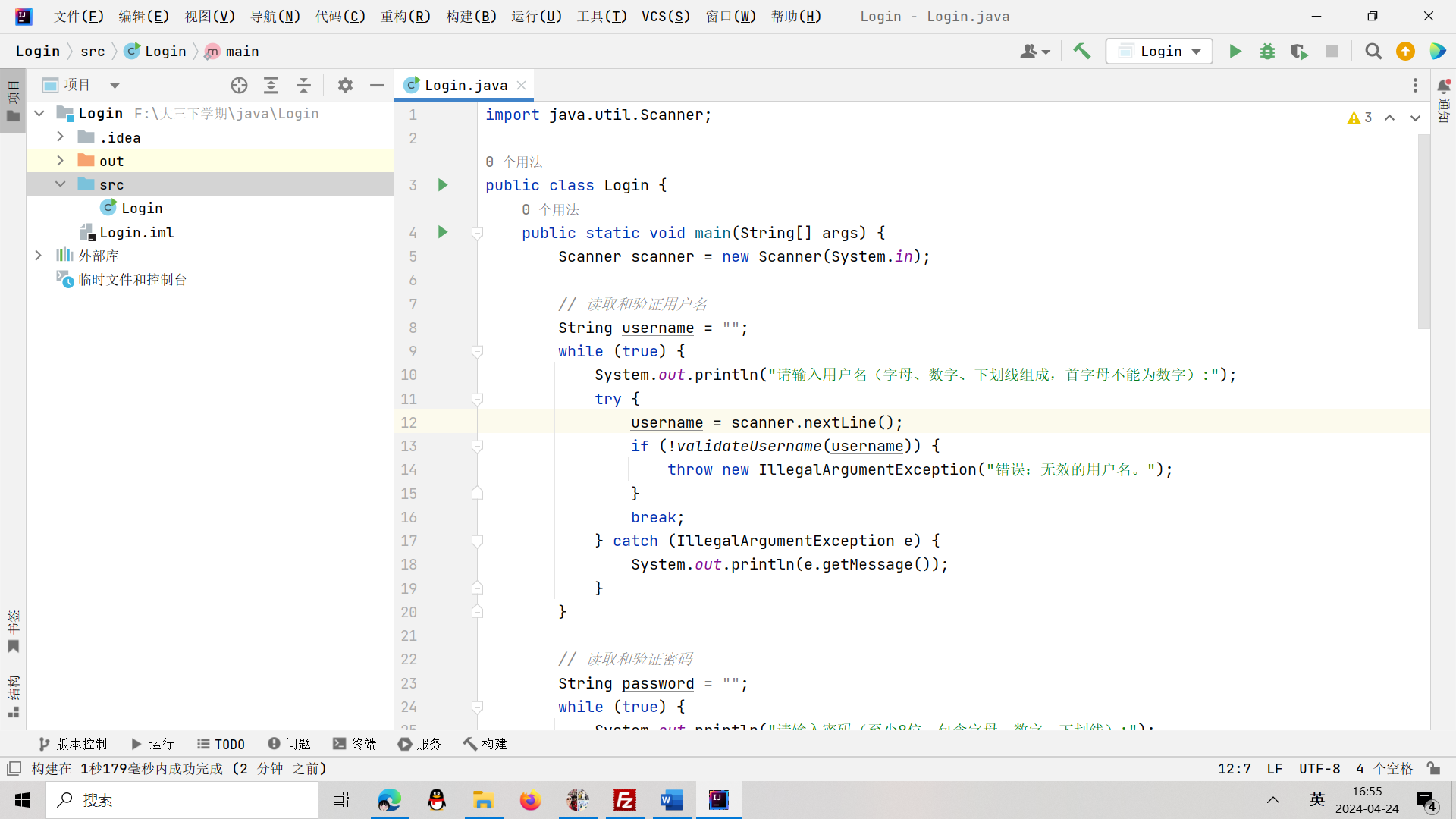
* validatePassword ------ ^\w{8,}$

^匹配输入字符串的开始，接着\\w匹配任何字母、数字或下划线。{8,}代表至少出现8次，最后$匹配输入字符串的结束。

* validateEmail ------ ^[^@\s][+@[^@\s]+\.[^@\s]+(?:\.[^@\s]+)\*$](mailto:+@[%5e@\s%5d+\.%5b%5e@\s%5d+(?:\.%5b%5e@\s%5d+)*$)

^匹配输入字符串的开始。[^@\\s]则是要求至少一个非@和非空白字符。@是字符 "@" 必须出现一次。[^@\\s]+代表"@"之后至少一个非@和非空白符。\\.代表点号.必须出现一次，用来分隔域名。[^@\\s]+则是点号之后至少一个非@和非空白字符。最后，(?:\\.[^@\\s]+)\*是一个非捕获组，匹配零个或多个由点号分隔的其他域名部分。$匹配输入字符串的结束。

1. 过程截图
2. 全屏截图：

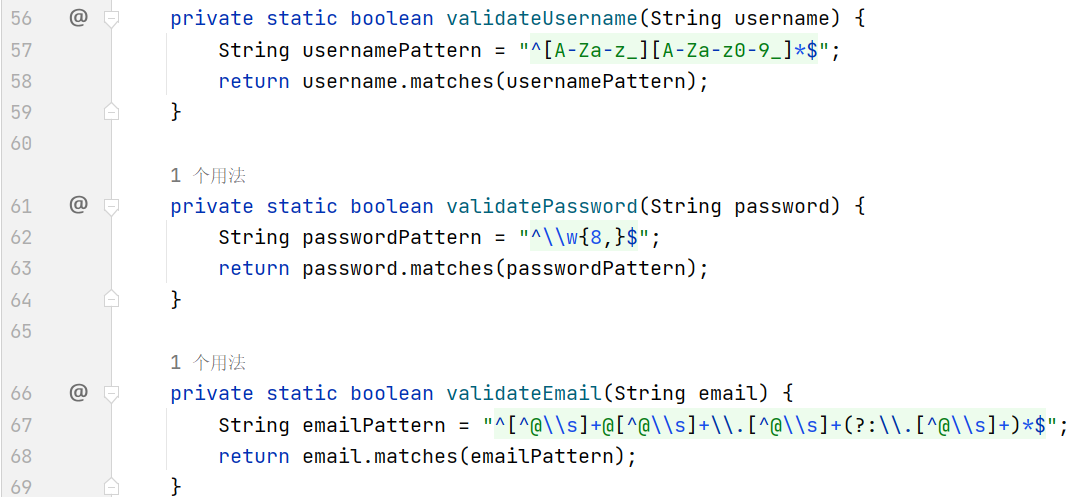


1. 主要代码：

Main类主要代码（以用户名为例，另外两个类似）：



三个正则表达式匹配检查方法：



1. 运行结果：



**三、实验总结与心得记录**

通过这次实验，我深入理解了Java中字符串处理和正则表达式的应用。在编写用户名、密码和邮箱验证功能时，我实践了各种正则表达式，学习了如何精确控制输入格式，确保用户输入的数据符合预期的规范。实验过程中，我也遇到了一些挑战，特别是在处理复杂的正则表达式时，需要精确地定义每个字符和符号的意义，这要求我不断测试和调整表达式以达到目标效果。