Pascal Scanner

The purpose of this assignment is to give you practice writing a scanner for Pascal.

Start with the Scanner and Token classes in the SimpleScanner project that we saw in class.

Modify the classes to handle the following Pascal reserved word tokens:

```
PROGRAM BEGIN END REPEAT UNTIL WRITE WRITELN DIV MOD
AND OR NOT CONST TYPE VAR PROCEDURE FUNCTION
WHILE DO FOR TO DOWNTO IF THEN ELSE CASE OF
```

Handle the following Pascal special symbol tokens:

```
.   ,   :   :=   ;    +   -   *   /   (    )
=   <>  <   <=   >   >=   ..   '   [   ]   ^
```

Also recognize these tokens:

```
IDENTIFIER INTEGER REAL CHARACTER STRING END_OF_FILE ERROR
```

You can make any modifications that you deem necessary to the other classes. For a more complete list of Pascal tokens, see the syntax diagrams.

Comments

Your scanner should treat each comment as it would treat a blank – comments should be ignored. Pascal comments are enclosed in curly braces { and }.

Strings and character literals

A literal Pascal string is enclosed in single quotes. If a single quote character is part of a string, it is represented by two consecutive single quotes. For example, `'It''s'` contains the characters It's. It is possible to have the empty string: `''`

Test Files

You may test your code on the test input files in resources directory of the Java project, using the command line or the unit tests provided.

Submission

- ✓ Check in your working code to your YU GitHub repository in your **`Compilers`** branch.
- ✓ The project (.pom file) should be rooted in the **`/hw2-scanner`** directory.