

JavaScript is disabled on your browser.

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

jminusminus

## Class LookaheadScanner

- [java.lang.Object](#)
  - [jminusminus.LookaheadScanner](#)

.

<https://powcoder.com>

```
class LookaheadScanner  
extends Object
```

Add WeChat powcoder

A lexical analyzer for j-- that interfaces with the hand-written parser (Parser.java). It provides a backtracking mechanism, and makes use of the underlying hand-written Scanner.

### • Field Summary

Fields	
Modifier and Type	Field and Description
boolean	<b><a href="#">isLookingAhead</a></b> Whether we are looking ahead.

### • Constructor Summary

Constructors	
Constructor and Description	
<b><a href="#">LookaheadScanner</a></b> (String fileName)	Construct a LookaheadScanner from a file name.

- **Method Summary**

Methods	
Modifier and Type	Method and Description
boolean	<b>errorHasOccured()</b> Has an error occurred up to now in lexical analysis?
String	<b>fileName()</b> Return the name of the source file.
void	<b>next()</b> Scan to the next token in the input.
TokenInfo	<b>previousToken()</b> The previously scanned token.
void	<b>recordPosition()</b> Record the current position in the input, so that we can start looking ahead in the input (and later return to this position).
void	<b>returnToPosition()</b> Return to the previously recorded position in the input stream of tokens.
TokenInfo	<b>token()</b> The currently scanned token.

Assignment Project Exam Help

- **Methods inherited from class java.lang.Object**

<https://powcoder.com>  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- **Field Detail**

- **isLookingAhead**

public boolean isLookingAhead

Whether we are looking ahead.

- **Constructor Detail**

- **LookaheadScanner**

public LookaheadScanner(String fileName)  
throws FileNotFoundException

Construct a LookaheadScanner from a file name.

**Parameters:**

fileName - the name of the file containing the source.

**Throws:**

FileNotFoundException - when the named file cannot be found.

- **Method Detail**

- **next**

public void next()

Scan to the next token in the input.

- **recordPosition**

```
public void recordPosition()
```

Record the current position in the input, so that we can start looking ahead in the input (and later return to this position). We'll queue up the current and subsequent tokens until `returnToPosition()` is invoked. These `recordPosition`'s can be nested.

- **returnToPosition**

```
public void returnToPosition()
```

Return to the previously recorded position in the input stream of tokens. If this is a nested lookahead, then return to the previous token queue.

- **token**

```
public TokenInfo token()
```

The currently scanned token.

**Returns:**

the current token.

- **previousToken**

```
public TokenInfo previousToken()
```

The previously scanned token. We use this in the parser to get at a token's semantic info (for example an identifier's name), after we've scanned it.

**Returns:**

the previous token.

- **errorHasOccurred**

```
public boolean errorHasOccurred()
```

Has an error occurred up to now in lexical analysis?

**Returns:**

true or false.

- **fileName**

```
public String fileName()
```

Return the name of the source file.

**Returns:**

name of the source file.

- Overview
- Package
- **Class**
- Tree
- Deprecated
- Index
- Help

- **Prev Class**
- **Next Class**

- Frames
- No Frames

- All Classes

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

**Assignment Project Exam Help**

**<https://powcoder.com>**

**Add WeChat powcoder**