




Java keywords

Keywords of the Java programming
language (Java 8)



Keywords of interest

In the Programming in Java book, we are concerned with the following keywords:

abstract	continue	for	package	this
assert	default	if	private	throw
boolean	double	implements	protected	throws
break	else	import	public	try
byte	enum	instanceof	return	void
case	extends	int	short	while
catch	final	interface	static	
char	finally	long	super	
class	float	new	switch	

Keywords of interest - classes

The following keywords are relevant when declaring a class (or enum or interface):

Context (before the actual class definition):

import package

Access level:

private protected public

Other modifier:

abstract final static

Entity (what are we declaring):

enum interface class

Hierarchy:

implements extends

Keywords of interest - fields (variables) of a class

The following keywords are relevant when declaring fields:

Access level:

private protected public

Other modifier:

final static

Type:

boolean byte char double float int long short

Note that classes like String when used as types are not keywords, they are identifiers (names)!

Keywords of interest - method declarations

For declaring methods of a class:

Access level:

private protected public

Other modifier:

abstract final static

Type:

boolean byte char float double int long short void

Exception signal:

throws

Keywords of interest - code which does something

The following keywords are used when writing code which doesn't do something, like the body of constructors and methods (and initializers):

Control flow:

continue for if break else return switch case while finally

Errors and exceptions:

assert throw try catch

(Local) Variable declarations:

boolean byte char double float int long short
final

Tests:

instanceof

Dereferencing

this super

Creation:

new

The rest (outside of the scope of this book/course)

Thread synchronization:

synchronized

Serialization:

transient

Floating point calculations:

strictfp

Concurrency/atomicity:

volatile

Methods written in other languages (differ between Java installations):

native

Java 8 interface default method modifier:

default

Further reading

- <https://docs.oracle.com/javase/specs/jls/se8/html/jls-3.html#jls-3.9>
- <https://docs.oracle.com/javase/tutorial/essential/concurrency/atomic.html>
- <https://docs.oracle.com/javase/tutorial/essential/concurrency/syncrrgb.html>
- <https://docs.oracle.com/javase/tutorial/java/landl/defaultmethods.html>

(For those who really want to know more!)