Ключевые слова в Java

Источники:

1. <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html>
2. <https://en.wikibooks.org/wiki/Java_Programming/Keywords>
3. <http://www.codejava.net/java-core/the-java-language/java-keywords>
4. <http://strongexperts.narod.ru/ru/articles/archive/java2/2006/apr2006-002/apr2006-002.htm>

Ключевые (зарезервированные) слова не могут использоваться в качестве идентификаторов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| abstract | continue | for | new | switch |
| assert\*\*\* | default | goto\* | package | synchronized |
| boolean | do | if | private | this |
| break | double | implements | protected | throw |
| Byte | else | import | public | throws |
| Case | enum\*\*\*\* | instanceof | return | transient |
| Catch | extends | int | short | try |
| Char | final | interface | static | void |
| Class | finally | long | strictfp\*\* | volatile |
| const\* | float | native | super | while |

|  |  |  |
| --- | --- | --- |
| \* |  | not used |
| \*\* |  | added in 1.2 |
| \*\*\* |  | added in 1.4 |
| \*\*\*\* |  | added in 5.0 |

Зарезервированы также true, false и null, но они являются литералами (значениями).

По категориям их можно разделить следующим образом:

* Примитивы: boolean, byte, char, double, float, int, long, short. Каждое из этих слов может быть использовано для объявления переменных

**char** a = **'a'**;  
**int**[] numbers = {4, 8, 15, 16, 23, 42};

* Циклы и ветвления: if, else, switch, case, default, while, do, break, continue, for. Зарезервировано, но не используется goto.

*// java.util.ArrayList#indexOf*

**public int** indexOf(Object o) {  
 **if** (o == **null**) {  
 **for** (**int** i = 0; i < **size**; i++)  
 **if** (**elementData**[i]==**null**)  
 **return** i;  
 } **else** {  
 **for** (**int** i = 0; i < **size**; i++)  
 **if** (o.equals(**elementData**[i]))  
 **return** i;  
 }  
 **return** -1;  
}

*// java.util.Arrays#deepEquals*

**public static boolean** deepEquals(Object[] a1, Object[] a2) {  
 **if** (a1 == a2) **return true**;  
 **if** (a1 == **null** || a2==**null**) **return false**;  
 **int** length = a1.length;  
 **if** (a2.length != length) **return false**;  
  
 **for** (**int** i = 0; i < length; i++) {  
 Object e1 = a1[i];  
 Object e2 = a2[i];  
  
 **if** (e1 == e2)  
 **continue**;  
 **if** (e1 == **null**)  
 **return false**;  
  
 *// Figure out whether the two elements are equal* **boolean** eq = deepEquals0(e1, e2);  
  
 **if** (!eq)  
 **return false**;  
 }  
 **return true**;  
}

*// java.lang.Long#formatUnsignedLong*

**static int** formatUnsignedLong(**long** val, **int** shift, **char**[] buf, **int** offset, **int** len) {  
 **int** charPos = len;  
 **int** radix = 1 << shift;  
 **int** mask = radix - 1;  
 **do** {  
 buf[offset + --charPos] = Integer.digits[((**int**) val) & mask];  
 val >>>= shift;  
 } **while** (val != 0 && charPos > 0);  
  
 **return** charPos;  
}

*// java.io.RandomAccessFile#readLine*

**public final** String readLine() **throws** IOException {  
 StringBuffer input = **new** StringBuffer();  
 **int** c = -1;  
 **boolean** eol = **false**;  
  
 **while** (!eol) {  
 **switch** (c = read()) {  
 **case** -1:  
 **case '\n'**:  
 eol = **true**;  
 **break**;  
 **case '\r'**:  
 eol = **true**;  
 **long** cur = getFilePointer();  
 **if** ((read()) != **'\n'**) {  
 seek(cur);  
 }  
 **break**;  
 **default**:  
 input.append((**char**)c);  
 **break**;  
 }  
 }  
  
 **if** ((c == -1) && (input.length() == 0)) {  
 **return null**;  
 }  
 **return** input.toString();  
}

* Исключения: try, catch, finally, throw, throws.
* Области видимости: private, <package-private>, protected, public.

**public void** throwException() **throws** Exception {  
 **try** {  
 **throw new** Throwable();  
 } **catch** (Throwable ignored) {  
 **throw new** Throwable();  
 } **finally** {  
 **throw new** Exception();  
 }  
}

Для классов верхнего уровня применяются только <package-private> и public. Остальные классы, методы, конструкторы, поля могут иметь любую область видимости.

**package** com.company;  
  
**import** java.util.Date;  
  
**import static** java.lang.Math.\*;  
  
**public class** Main {  
 **public static void** main(String... args) **throws** Throwable {  
 System.***out***.println(**new** Date());  
 System.***out***.println(*cos*(***PI***));  
 }  
}

* Пакеты: package, import.
* Объявление: class, interface, enum, extends, implements, static, final, const (не используется), void (почему не к примитивам?), abstract, native, strictfp, synchronized, volatile, transient.

**public native static void** method1();  
**public static native void** method2();  
**static public native void** method3();  
**static native public void** method4();  
**native public static void** method5();  
**native static public void** method6();  
**strictfp public static void** method7() {}  
**strictfp static public void** method8() {}  
**public strictfp static void** method9() {}  
**public static strictfp void** method10() {}  
**static public strictfp void** method11() {}  
**static strictfp public void** method12() {}  
**abstract public void** method13();  
**public abstract void** method14();

*// To be continued*

* Другие операторы: new, return, synchronized, instanceof, assert.

*// javac Main.java  
// java Main -ea***public abstract class** Main {  
 **public static void** main(String[] args) {  
 **assert false** : **"Human-readable message"**; *// throws AssertionError* }  
}

* Получение контекстных ссылок: this, super.