AP COMPUTER SCIENCE
JAVA CONCEPTS IV: IDENTIFIERS AND KEYWORDS

PAUL L. BAILEY

Abstract. This documents amalgamates various descriptions found on the internet, mostly from Oracle or Wikipedia. The initial section is a synthesis, but very little beyond this originates with the author listed above, who acted merely as an editor.

1. Identitifiers, Keywords, and Reserved Words

1.1. **Identifiers.** *Identitifiers* are the names of variables, methods, classes, packages and interfaces.

The names of variables, methods, classes, interfaces, and packages must following these rules:

- An identifier is a sequence of one or more characters.
- The first characters must be a letter, a dollar sign (ASCII 36), or an underline (ASCII 95).
- The subsequent characters must be a valid first characters or a digit.
- The string must not be a reserved word.

1.2. **Reserved Words and Keywords.** *Reserved words* are strings that cannot be used as identifiers, because they are part of the language itself. As of Java 5.0, there are two reserved words that are not actually used; these are `goto` and `const`. Moreover, there are three reserved words that are literals; these are `null`, `true`, and `false`. Some authors say that these latter five reserved words are not keywords. We will henceforth blur any distinction between reserved words and keywords.

Date: September 20, 2019.

2. Keywords

The Java language (as of version 5.0) has 50 keywords that are not literals. These words are used as part of the Java language itself, and as such, may not be used as identifiers in code.

`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`

In addition to the 50 keywords, there are three additional reserved words which are literal constants.

`true`	`false`	`null`

Three keywords have been added in Java 9.0.

`exports`	`module`	`requires`

We now classify our favorite keywords.

2.1. **Primitive Types.** These keywords indicate types of variables or methods.

`byte`	`short`	`int`	`long`
`float`	`double`		
`boolean`	`char`	`void`	

2.2. **Literals.** These keywords are literal values.

`true`	`false`	`null`

2.3. **Flow control keywords.** These keywords control which code is executed.

`if`	`else`	
`switch`	`case`	`default`
`while`	`for`	`do...while`
`break`	`continue`	`return`

2.4. **Object oriented keywords.** These keywords relate to classes and inheritance.

`class`	`extends`		
`interface`	`implements`		
`abstract`	`super`	`this`	`new`

2.5. **Method, and variable modifiers.** These keywords modify the behavior of a class, method, or variable.

`public`	`private`	`protected`	`static`

2.6. **Error handling and debugging.** These keywords provide tools to aid in error handing and debugging.

`try`	`catch`	`finally`
`throw`	`throws`	`assert`

We have previously listed the primitive types. We now summarize the other keywords that are in the AP Java subset, are inherited from the `C++` programming language, or are otherwise significant.

Reserved Word	AP	Purpose
`abstract`		Abstract methods have no implementation
`assert`		Assert that a condition should be true, for debugging
`break`		Break out of a loop or a `case`
`case`		Case behavior in a `switch` statement
`catch`		Catch an exception after a `try` statement
`class`	√	Declare a new class
`continue`		Transfer to the top of the loop
`default`		Default behavior in a `switch` statement
`do`		Begin a `do...while` loop
`else`	√	Execute code if an `if` condition is false
`enum`		Declare a type which is a list of values
`extends`	√	Indicates a class extends a parent class
`final`	√	Indicates a constant variable
`finally`		The last code to execute after a `try` statement
`for`	√	Begin a `for` loop
`implements`	√	Indicates a class implements an interface
`import`	√	Reference an external package
`if`	√	Execute code if a condition is true
`interface`	√	Declare a new interface
`new`	√	Allocate memory for a new object
`private`	√	Indicates an entity may be not referenced outside the class
`protected`		Indicates an entity may only be referenced by child classes
`public`	√	Indicates an entity may be reference outside the class
`return`	√	Exits a method
`static`	√	Indicates an entity is not specific to an instance
`super`	√	Invokes a parent's constructor
`switch`		Begin a `switch ... case` flow structure
`this`	√	Reference to the current object
`throw`		Throw an exception
`throws`		Indicates a method throws exceptions
`try`		Try to execute some code that may throw an exception
`while`	√	Begin a `while` loop

3. Keywords

abstract. The abstract keyword is used to declare a class or method to be abstract. An abstract method has no implementation; all classes containing abstract methods must themselves be abstract, although not all abstract classes have abstract methods. Objects of a class which is abstract cannot be instantiated, but can be extended by other classes. All subclasses of an abstract class must either provide implementations for all abstract methods, or must also be abstract.

assert. The assert keyword is used to make an assertion - a statement which the programmer believes is always true at that point in the program. If assertions are enabled when the program is run and it turns out that an assertion is false, an AssertionError is thrown and the program terminates. This keyword is intended to aid in debugging.

boolean. The boolean keyword is used to declare a field that can store a boolean value; that is, either true or false. This keyword is also used to declare that a method returns a value of the primitive type boolean.

break. Used to resume program execution at the statement immediately following the current enclosing block or statement. If followed by a label, the program resumes execution at the statement immediately following the enclosing labeled statement or block.

byte. The byte keyword is used to declare a field that can store an 8-bit signed two's complement integer. This keyword is also used to declare that a method returns a value of the primitive type byte.

case. The case keyword is used to create individual cases in a switch statement; see switch.

catch. Defines an exception handler - a group of statements that are executed if an exception is thrown in the block defined by a preceding try keyword. The code is executed only if the class of the thrown exception is assignment compatible with the exception class declared by the catch clause.

char. The char keyword is used to declare a field that can store a 16-bit Unicode character]. This keyword is also used to declare that a method returns a value of the primitive type char.

class. A type that defines the implementation of a particular kind of object. A class definition defines instance and class fields, methods, and inner classes as well as specifying the interfaces the class implements and the immediate superclass of the class. If the superclass is not explicitly specified, the superclass is implicitly Object.

const. Although reserved as a keyword in Java, const is not used and has no function. For defining constants in java, see the 'final' reserved word.

continue. Used to resume program execution at the end of the current loop body. If followed by a label, continue resumes execution at the end of the enclosing labeled loop body.

default. The default can optionally be used in a switch statement to label a block of statements to be executed if no case matches the specified value; see switch. Alternatively, the default keyword can also be used to declare default values in a Java annotation.

do. The do keyword is used in conjunction with while to create a do-while loop, which executes a block of statements associated with the loop and then tests a boolean expression associated with the while. If the expression evaluates to true, the block is executed again; this continues until the expression evaluates to false.

double. The double keyword is used to declare a field that can hold a 64-bit double precision IEEE 754 floating-point number. This keyword is also used to declare that a method returns a value of the primitive type double.

else. The else keyword is used in conjunction with if to create an if-else statement, which tests a boolean expression; if the expression evaluates to true, the block of statements associated with the if are evaluated; if it evaluates to false, the block of statements associated with the else are evaluated.

enum. A Java keyword used to declare an enumerated type. Enumerations extend the base class Enum.

extends. Used in a class declaration to specify the superclass; used in an interface declaration to specify one or more superinterfaces. Class X extends class Y to add functionality, either by adding fields or methods to class Y, or by overriding methods of class Y. An interface Z extends one or more interfaces by adding methods. Class X is said to be a subclass of class Y; Interface Z is said to be a subinterface of the interfaces it extends. Also used to specify an upper bound on a type parameter in Generics.

false. A boolean literal value.

final. Define an entity once that cannot be changed nor derived from later. More specifically: a final class cannot be subclassed, a final method cannot be overridden, and a final variable can occur at most once as a left-hand expression. All methods in a final class are implicitly final.

finally. Used to define a block of statements for a block defined previously by the try keyword. The finally block is executed after execution exits the try block and any associated catch clauses regardless of whether an exception was thrown or caught, or execution left method in the middle of the try or catch blocks using the return keyword.

float. The float keyword is used to declare a field that can hold a 32-bit single precision IEEE 754 floating-point number. This keyword is also used to declare that a method returns a value of the primitive type float.

for. The for keyword is used to create a for loop, which specifies a variable initialization, a boolean expression, and an incrementation. The variable initialization is performed first, and then the boolean expression is evaluated. If the expression evaluates to true, the block of statements associated with the loop are executed, and then the incrementation is performed. The boolean expression is then evaluated again; this continues until the expression evaluates to false. The for keyword can also be used to create a so-called "enhanced for loop", which specifies an array or Iterable object; each iteration of the loop executes the associated block of statements using a different element in the array or Iterable.

goto. Although reserved as a keyword in Java, goto is not used and has no function.

if. The if keyword is used to create an if statement, which tests a boolean expression; if the expression evaluates to true, the block of statements associated with the if statement is executed. This keyword can also be used to create an if-else statement; see else.

implements. Included in a class declaration to specify one or more interfaces that are implemented by the current class. A class inherits the types and abstract methods declared by the interfaces.

import. Used at the beginning of a source file to specify classes or entire Java packages to be referred to later without including their package names in the reference.

instanceof. A binary operator that takes an object reference as its first operand and a class or interface as its second operand and produces a boolean result. The instanceof operator evaluates to true if and only if the runtime type of the object is assignment compatible with the class or interface.

int. The int keyword is used to declare a field that can hold a 32-bit signed two's complement integer. This keyword is also used to declare that a method returns a value of the primitive type int.

interface. Used to declare a special type of class that only contains abstract methods, constant (static final) fields and static interfaces. It can later be implemented by classes that declare the interface with the implements keyword.

long. The long keyword is used to declare a field that can hold a 64-bit signed two's complement integer. This keyword is also used to declare that a method returns a value of the primitive type long.

native. Used in method declarations to specify that the method is not implemented in the same Java source file, but rather in another language.

new. Used to create an instance of a class or array.

null. A reference literal value, indicating that a class variable has no associated instance.

package. A group of types. Packages are declared with the package keyword.

private. The private keyword is used in the declaration of a method, field, or inner class; private members can only be accessed by other members of their own class.

protected. The protected keyword is used in the declaration of a method, field, or inner class; protected members can only be accessed by members of their own class, that class's subclasses or classes from the same package.

public. The public keyword is used in the declaration of a class, method, or field; public classes, methods, and fields can be accessed by the members of any class.

return. Used to finish the execution of a method. It can be followed by a value required by the method definition that is returned to the caller.

short. The short keyword is used to declare a field that can hold a 16-bit signed two's complement integer. This keyword is also used to declare that a method returns a value of the primitive type short.

static. Used to declare a field, method, or inner class as a class field. Classes maintain one copy of class fields regardless of how many instances exist of that class. static also is used to define a method as a class method. Class methods are bound to the class instead of to a specific instance, and can only operate on class fields. (Classes and interfaces declared as static members of another class or interface are actually top-level classes and are not inner classes.)

strictfp. A Java keyword used to restrict the precision and rounding of floating point calculations to ensure portability.

super. Used to access members of a class inherited by the class in which it appears. Allows a subclass to access overridden methods and hidden members of its superclass. The super keyword is also used to forward a call from a constructor to a constructor in the superclass. Also used to specify a lower bound on a type parameter in Generics.

switch. The switch keyword is used in conjunction with case and default to create a switch statement, which evaluates a variable, matches its value to a specific case, and executes the block of statements associated with that case. If no case matches the value, the optional block labeled by default is executed, if included.

synchronized. Used in the declaration of a method or code block to acquire the mutex lock for an object while the current thread executes the code. For static methods, the object locked is the class' Class. Guarantees that at most one thread at a time operating on the same object executes that code. The mutex lock is automatically released when execution exits the synchronized code. Fields, classes and interfaces cannot be declared as synchronized.

this. Used to represent an instance of the class in which it appears. this can be used to access class members and as a reference to the current instance. The this keyword is also used to forward a call from one constructor in a class to another constructor in the same class.

throw. Causes the declared exception instance to be thrown. This causes execution to continue with the first enclosing exception handler declared by the catch keyword to handle an assignment compatible exception type. If no such exception handler is found in the current method, then the method returns and the process is repeated in the calling method. If no exception handler is found in any method call on the stack, then the exception is passed to the thread's uncaught exception handler.

throws. Used in method declarations to specify which exceptions are not handled within the method but rather passed to the next higher level of the program. All uncaught exceptions in a method that are not instances of RuntimeException must be declared using the throws keyword.

transient. Declares that an instance field is not part of the default serialized form of an object. When an object is serialized, only the values of its non-transient instance fields are included in the default serial representation. When an object is deserialized, transient fields are initialized only to their default value. If the default form is not used, e.g. when a serialPersistentFields table is declared in the class hierarchy, all transient keywords are ignored.

true. A boolean literal value.

try. Defines a block of statements that have exception handling. If an exception is thrown inside the try block, an optional catch block can handle declared exception types. Also, an optional finally block can be declared that will be executed when execution exits the try block and catch clauses, regardless of whether an exception is thrown or not. A try block must have at least one catch clause or a finally block.

void. The void keyword is used to declare that a method does not return any value.

volatile. Used in field declarations to specify that the variable is modified asynchronously by concurrently running threads. Methods, classes and interfaces thus cannot be declared volatile.

while. The while keyword is used to create a while loop, which tests a boolean expression and executes the block of statements associated with the loop if the expression evaluates to true; this continues until the expression evaluates to false. This keyword can also be used to create a do-while loop; see do.

4. References

Reference: `http://docs.oracle.com/javase/tutorial/java/nutsandbolts/`
Reference: `http://en.wikipedia.org/wiki/List_of_Java_keywords`

Department of Mathematics, BASIS Scottsdale
Email address: `paul.bailey@basised.com`