Android 101 - 1 || Outline

Index

c. Initialization

11. Imports

Syntax Definitions

Object - Objects have states and behaviors. Example: A dog has states-color, name, breed as well as behaviors -wagging, barking, eating. An object is an instance of a class.
Class - A class can be defined as a template/ blue print that describe the behaviors/states that object of its type support.
Methods - A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.
Variables - Each object has its unique set of instant variables. An object's state is created by the values assigned to these instant variables.
Package - A way to organize classes into categories or with other classes that have similar functionality. Represented as folders on a file system.

Naming

- All naming is Case Sensitive
- Lowercase, Uppercase, Camel Case, Mixed Case (Lower Camel Case)
- Classes: Camel Case, Nouns with Adjectives if necessary
- Methods: Mixed Case, Verb-Adjective-Noun
- Variables: Mixed Case, Higher level variables should have more descriptive names than short lived ones usually. (Underscore naming convention)
- Constants: All Uppercase
- Packages: Lower Case, No Symbols except periods, No Spaces, Reverse Owned Domain (ex. “com.bnotions.bestappever”)

Keywords

Keywords are reserved words in Java that have a specific purpose and can not be used for anything else but that purpose. The most used keywords are listed below this chart and should be memorized since they are used very frequently.

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

Question: Wow! That's a heck of a lot of keywords, do I really need to memorize them all?

Answer: No, for the purpose of this class the only variables you will need to know are the following,

Scope Keywords

private - The private keyword is used in the declaration of a Method, Field, or Class; private members can only be accessed by other members of their own class.
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.
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.
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.

Instantiation Keywords

new - Used to create an instance of a Class or an Object.

Type Keywords

boolean - The boolean keyword is used to declare a field that can store a boolean value; that is, either true or false.
int - The int keyword is used to declare a field that can hold a 32-bit signed whole number
void - The void keyword is used to declare that a method does not return any value.

Condition Keywords

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”.
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.

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.

Loop Keywords

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.

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.

Comments

Java comments are either explanations of the source code or descriptions of classes, interfaces, methods, and fields.

Single Line:

```
//This is a single line comment
```

Multi-LIne:

```
/*
This is a multi
line
comment
*/
```

Imports

A way to reference classes so that you can call them directly by short name such as Button instead of android.widget.Button. There are 166 packages containing 3279 classes and interfaces in Java (Not to mention all of the Android ones in the Android language), too many to remember, but thankfully an IDE usually handles import management for you.

import android.widget.Button;

Creating An Object

Declaration

Variables can be declared in two ways.

Instance Variable

If it is a variable you intend to use throughout the life of your class, then it should be declared as an “instance variable”. These variables are defined inside your class but outside of all methods.

1) Define scope: “private”
2) Define type: “String”
3) Define name in mixed case: “myName”

private String myName;

Local Variable

If it is a variable you intend to use only throughout the execution of a method, then it should be declared as a “local variable”. These variables are defined inside methods only.

1) Define type: “String”
2) Define name in mixed case: “myName”

String myName;

Instantiation

Variables can be declared and instantiated all in one line, but this is usually only done for local variables since instance variables would be instantiated as soon as the class is instantiated and therefore use unnecessary memory if you never end up using those instance variables.

1) State variable name: “myName”
2) Use an equals sign to assign a value: “=”
3) Use the new keyword to create an object in memory: “new”
4) Declare the type again: “String”
5) Add two parentheses and input any arguments that are required by the class: “(“Qwerty Thompson”);”

myName = new String(“Qwerty Thompson”);