Android Codestyle v1.0

deggan

Contents

deggan

Background

deggan

Why?

"Different Java programmers can **have different styles** and approaches to the way they program. By using standard Java naming conventions they make their code **easier to read** for themselves and **for other programmers**. Readability of Java code is important because it means less time is spent trying to figure out what the code does, leaving more time to fix or modify it." [source]

deggan

Preface

deggan

Using Letter Case

Using the right letter case is the key to following a naming convention:

- **lowercase** is where all the letters in a word are written without any capitalization (e.g., while, if, mypackage).
- **UPPER_CASE** is where all the letters in a word are written in capitals. When there are more than two words in the name use underscores to separate them (e.g., MAX_HOURS, FIRST_DAY_OF_WEEK).
- **CamelCase** (also known as Upper CamelCase) is where each new word begins with a capital letter (e.g., CamelCase, CustomerAccount, PlayingCard).
- **mixedCase** (also known as Lower CamelCase) is the same as CamelCase except the first letter of the name is in lowercase (e.g., hasChildren, customerFirstName, customerLastName). [source]

deggan

Naming Conventions

deggan

Naming Conventions

"When choosing a name for an identifier, make sure **it's meaningful**. For instance, if your program deals with customer accounts then choose names that make sense for dealing with customers and their accounts (e.g., customerName, accountDetails)."

"Don't worry about the length of the name. A longer name that sums up the **identifier perfectly is preferable** to a shorter name that might be quick to type but ambiguous." [source]

deggan

Deggan Naming Conventions

deggan

Packages

- Names should be in **lowercase**.

Example:

```
package com.deggan.view;
```

deggan

Classes

- Names should be in **CamelCase**.
- Try to use **nouns**.

  (because a class is normally representing something in the real world)

Example:

```
public class Purchase {...}
```

deggan

Interfaces

- Names should be in **CamelCase**.
- Note: some programmers like to distinguish interfaces by beginning the name with an "I".

Example:

```
public interface InboxInterface {...}
public interface IMerchant {...}
```

deggan

Methods

- Names should be in **mixedCase**.
- Use **verbs** to describe what the method does.

Example:

```
private void getLogin(String username, String password) {...}
private void performUpdateStatus() {...}
```

deggan

Variables

- Names should be in **mixedCase**.
- Names should **represent what the value** of the variable represents
- Only use very short names when the variables are short-lived. (such as in for loops)

Example:

```
int numberOfLevels = 4;
String currentVersion = "" + BuildConfig.VERSION_CODE;
```

deggan

Constants

- Names should be in **UPPER_CASE**.

Example:

```
public static final String TYPE_SPEED_TEST = "...";
```

deggan

Deggan Java Code Style

deggan

Deggan Java Variables

- Names should be in **mixedCase**.
- **Prefix** with **type** of variables (see the table).

Example:

Button ***buttonLogin***;

EditText ***editName***;

deggan

Deggan Java Variables (Table)

No	Layout	Type
1	TextView	text...
2	Button	button...
3	ImageButton	button...
4	RecyclerView	recycler...
5	ImageView	image...
6	Switch	switch...
7	CheckBox	check...
8	RadioButton	radio...
9	RadioGroup	radiogroup...
10	View	view...

No	Layout	Type
11	ScrollView	layout...
12	RelativeLayout	layout...
13	LinearLayout	layout...
14	FrameLayout	layout...
15	SeekBar	seek...
16	ProgressBar	progress...
17	WebView	web...
18	CardView	card...
19	TabLayout	tab...
20	Toolbar	toolbar...

deggan

Deggan Java Classes

- Names should be in **CamelCase**.
- **Postfix** with **type** of class (See the table).

Example:

public class ***InputDialog*** *extends DialogFragment {...}*

public class ***BuyVoucherActivity*** *{...}*

deggan

Deggan Java Classes (Table)

No	Class	Type
1	Activity Class	...Activity.java
2	Fragment Class	...Fragment.java
3	Dialog Class	...Dialog.java
4	Adapter Class	...Adapter.java
5	Presenter Class	...Presenter.java
6	Service Class	...Service.java
7	Utilities Class	...Utils.java
8	Manager Class	...Manager.java
9	Config Class	...Config.java
10	Helper Class	...Helper.java

deggan

Deggan XML Code Style

deggan

Deggan XML Files

- Names should be in **lowercase with underscore**.
- **Prefix** with **type** of layout (activity_user_balance.xml).
- **Postfix** with **name** of layout (See the table).

Example:

activity_buy_voucher.xml

dialog_information.xml

deggan

Deggan XML Files (Table)

No	Class	Type
1	Activity Class	activity_…xml
2	Fragment Class	fragment_…xml
3	Dialog Class	dialog_…xml
4	Adapter Item Class	item_…xml
5	View Class	layout_…xml
6	Widget Class	widget_…xml

deggan

Deggan XML Variables

- Names should be in **lowercase with underscore**.
- **Prefix** with **name** of layout (activity_user_balance.xml).
- **Postfix** with **type** of layout (See the table).

Example:

```
<Button android:id="@+id/back_button ... />
<TextView android:id="@+id/user_balance_text ... />
```

deggan

Deggan XML Variables (Table)

No	Layout	Type
1	TextView	..._text
2	Button	..._button
3	ImageButton	..._button
4	RecyclerView	..._recycler
5	ImageView	..._image
6	Switch	..._switch
7	CheckBox	..._check
8	RadioButton	..._radio
9	RadioGroup	..._radio_group
10	View	..._view

No	Layout	Type
11	ScrollView	..._layout
12	RelativeLayout	..._layout
13	LinearLayout	..._layout
14	FrameLayout	..._layout
15	SeekBar	..._seek
16	ProgressBar	..._progress
17	WebView	..._web
18	CardView	..._card
19	TabLayout	..._tab
20	Toolbar	..._toolbar

deggan

Logging

deggan

Logging

- Create log messages that appear in logcat.
- The highest to lowest priority (error >warning > information > debug > verbose)
- Reference: https://developer.android.com/studio/debug/am-logcat.html

Example:

```
Log.i(TAG, "setDialogFeedback: onClosedPressed");
Log.d(TAG, "getInbox: onSuccess " + response);
```

deggan

Deggan Logging

- Make log messages easy to understand
- For default log always use Log.d(...); or setLog(...)
- For message log always use English with space delimiter
- Log explain: [Function Name] + [SubFunction] + [Purpose] + [Comment]

Example:

```
Log.d(TAG, "setDialog: onInitiated");
Log.d(TAG, "getInbox: getInboxStatus onSuccess " + response);
```

deggan

Debugging

deggan

Debugging

- Debugging is the process of finding and resolving defects or problems within a computer program that prevent correct operation of computer software or a system. [source]
- In the Debugger window, use the variables pane to inspect variables when the system stops the app on a breakpoint. The variables pane also evaluate ad-hoc expressions using static methods and/or variables available within the selected frame.

deggan

Lint Checks

deggan

Lint Checks

- Android Studio provides **a code scanning tool** called lint that can help you to identify and correct problems with the structural quality of your code without your having to execute the app or write test cases.
- Each problem detected by the tool is reported with a description message and a severity level, so that you can quickly prioritize the critical improvements that need to be made. [source]

1. Find: **“F2”** 2. Fix: **“alt + Enter”** 3. Format: **“ctrl + alt + P”**

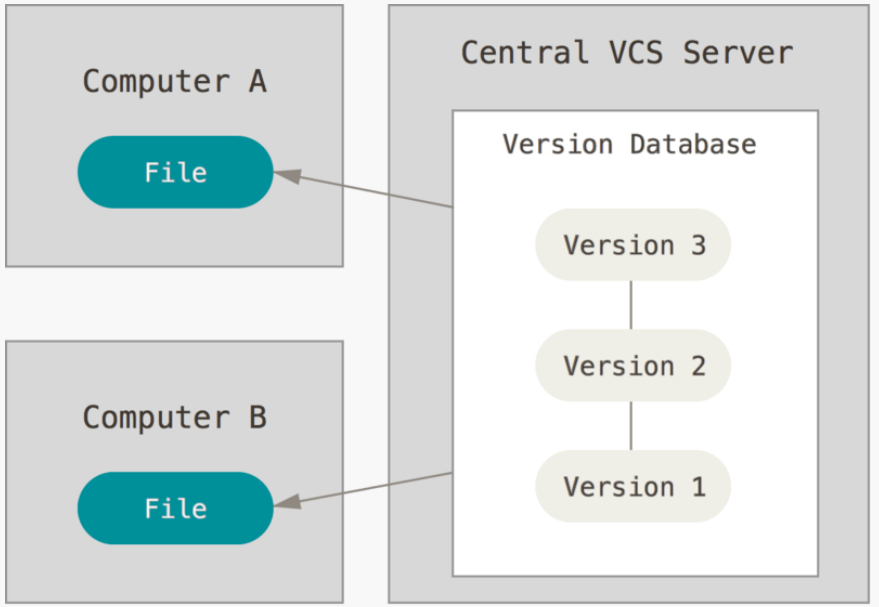
deggan

Git Version Control

deggan

Version Control

- Version control is a system that records **changes to a file or set of files** over time so that you can recall specific versions later. [source]
- Git allows to have multiple local branches that can be entirely independent of each other.
- The creation, merging, and deletion of those lines of development takes seconds.



deggan

Deggan VCS

- Visual VCS Tools **GUI is preferred**
- Command line method is also allowed
- **Android Studio** Built-in VCS is recommended
- Other recommendations:
  - **Fork** *(https://git-fork.com/)*
  - **SourceTree** *(https://www.sourcetreeapp.com)*

deggan

Deggan Commit

- Make commit message easy to understand
- For message always use English
- Commit explain: [Action taken] + [File] + [Purpose (optional)] + [Comment (optional)]

Example:

Refactor BaseActivity

Add getResponse on BaseActivity

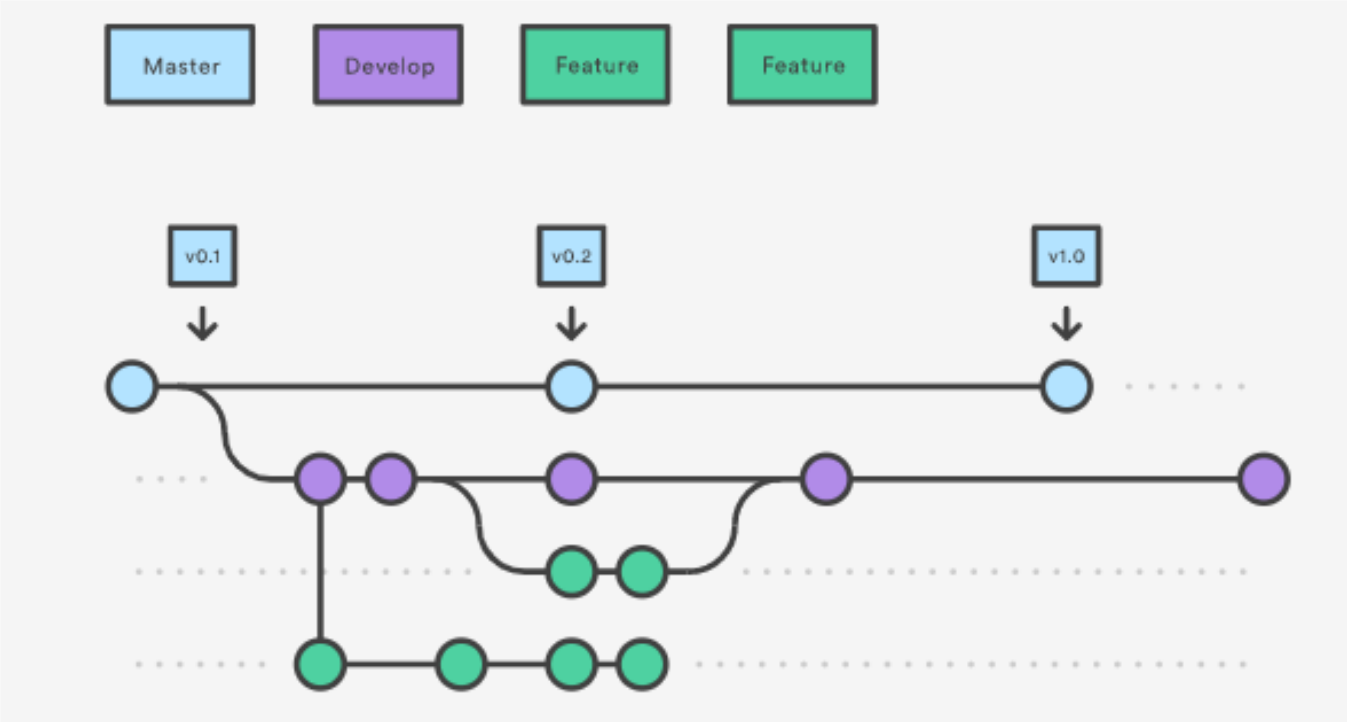
deggan

Git Flow

deggan

Using Git Flow

- Git-flow are a set of **git extensions** to provide **high-level repository operations** for **Vincent Driessen**'s branching model.



deggan

Resources

- https://google.github.io/styleguide/javaguide.html
- https://source.android.com/setup/contribute/code-style
- https://www.thoughtco.com/using-java-naming-conventions-2034199
- https://docs.oracle.com/cd/E82085_01/150/funtional_artifacts_guide/or-fasg-standards.htm

Revision History

Version	Date	User	Description
1.0	17/02/2020	Fenton Martin	Publish initial document