

ANDROID DEVELOPMENT STANDARD

Author: Le Duc Tuan

Preface

SIGNATURE	
Subject	Date
Originator: Le Duc Tuan	19 September 2016
Prepared by: Le Duc Tuan	19 September 2016
Distributed to: Android Team	


Preface

HISTORY				
Version	Date	A, M, D	Description	Author
1.0	19 September 2016	A	Create document	Le Duc Tuan

Goals

A decorative graphic consisting of several parallel white diagonal lines of varying lengths, located in the bottom right corner of the slide.

Goals

- ❖ Working in the same bases and conventions
 - ❖ Ease of project backup
 - ❖ Ease of maintenance
- 
- A series of several parallel white diagonal lines in the bottom right corner of the slide, extending from the middle of the right edge towards the bottom left.

Language

Language

Conventions

- ❖ Development and documentation language is *English*
- ❖ All class, method and variable names are in English and follow below rules
- ❖ Gives simplest and shortest names as possible
- ❖ Avoids using plural (uses collection type)
- ❖ Words are lowercase when alone, otherwise use Camel Case
- ❖ Gives the generic term as possible, avoid gender and false specialization
- ❖ Avoids confusing acronyms, use full text to describe the meaning

Examples (*Do* # *Don't*)

- ❖ `performSearch()` # `recherche()`
- ❖ `createUserList()` # `createListOfUsers()`
- ❖ `mCatList` # `mCats`
- ❖ `person`, `personAge`, `personWithName`
- ❖ `PersonClass` # `ManClass`, `WomanClass`
- ❖ `HTTP`, `URL` # `AAA` (Battery Type/Codon Lysine, etc.)

Project Name

Project Name

Conventions

- ❖ The project name must follow client requirement first, otherwise uses this convention
- ❖ Each project name must be identified by a prefix and postfix (if required) as the rules follow
 - [Company Name]Project Name[Version]*
- ❖ No duplications with other projects

Examples (*Do* # *Don't*)

- ❖ GoogleGmailV1 # Gmail
- ❖ FacebookMessenger # Messenger

Naming Convention

A decorative graphic consisting of several parallel white diagonal lines of varying lengths, located in the bottom right corner of the slide.

Naming Convention

This section describes the naming convention of following subjects:

- ❖ Application Package Name
- ❖ Classes & Members (variables)
- ❖ Accessor Members
- ❖ Methods & Parameters
- ❖ Implementation
- ❖ Listeners
- ❖ Constants
- ❖ Layouts
- ❖ Values
- ❖ Identifiers
- ❖ Localization

Application Package Name

Conventions

- ❖ Uses the package name assigned by client or uses this convention
- ❖ All lowercase and as follow
[com.][company name.][project name]
- ❖ Packages separated by application functions
- ❖ Any other packages that are commonly used throughout the project must be put into [common] package

Examples

Do

- ▶ `com.uniqcompany.project`
 - ❖ `com.uniqcompany.project.login`
 - ❖ `com.uniqcompany.project.common.gcm`

Don't

- ▶ `com.facebook.katana` (duplicate with facebook application)
- ▶ `uniqcompany.project` (missing com.)

Classes & Members

Conventions [Classes]

- ❖ Starts with company name/project abbreviation and use Camel Case
- ❖ Ends with class type or functional usage
[company/project name][class type]

Examples

Do

- ❖ UniqCompanyLoginFragment
- ❖ UniqCompanyHomeActivity
- ❖ UniqCompanyUserModel

Don't

- ❖ HomeFragment (missing company prefix)
- ❖ CustomView (not specific)
- ❖ UpdateUserInfo (class type Info?)

Classes & Members (cont.)

Conventions [Members]

- ❖ Starts with “m” and use Camel Case
- ❖ Avoids using numbers, add functional usage or purpose of the variable instead
- ❖ Gives simplest and shortest member name as possible
- ❖ For boolean type, the prefix followed by “is” and end with an adjective

Examples

Do

- ❖ mProduct
- ❖ mProductList
- ❖ mIsProductValid

Don't

- ❖ checked (missing “is” & “m”, not specific)
- ❖ mProduct3 (no numbers, purpose instead)
- ❖ mainProduct (missing “m”)

Accessor Members

Conventions

- ❖ Removes “m” for both getters and setters
- ❖ Setter
 - ❖ Followed by variable name
 - ❖ Uses Camel Case
- ❖ Getter
 - ❖ Followed by variable name
 - ❖ Uses Camel Case
 - ❖ For boolean type, “is” followed by the variable name

Examples

Do

- ❖ `setProduct(), getProduct()`
- ❖ `setProductList(), getProductList()`
- ❖ `setProductValid(), isProductValid()`

Don't

- ❖ `setproduct(), SetProduct()` (Camel Case)
- ❖ `productValid()` (missing “is”)
- ❖ `isProduct()` (noun)

Methods & Parameters

Conventions [Methods]

- ❖ Starts with a lowercase verb/"is"
- ❖ Uses Camel Case
- ❖ Avoids abbreviation
- ❖ Gives simplest and shortest method name as possible

Examples

Do

- ❖ `performSearch()`
- ❖ `initialize()`
- ❖ `findProductById()`
- ❖ `isProductAvailable()`

Don't

- ❖ `setproduct()`, `Setproduct()` (Camel Case)
- ❖ `isProduct()` (noun)

Methods & Parameters (cont.)

Conventions [Parameters]

- ❖ Uses Camel Case
- ❖ Avoids abbreviation
- ❖ Gives simplest and shortest parameter name as possible

Examples

Do

- ❖ `product, productList`
- ❖ `name, id, isVisible`

Don't

- ❖ `Product` (Camel Case)
- ❖ `mProduct` (same as [Members])
- ❖ `isVisible` (Camel Case)

Implementation

Conventions [Local variables]

- ❖ Starts with "t"
- ❖ Uses Camel Case
- ❖ Avoids abbreviation
- ❖ Gives simplest and shortest variable name as possible

Examples

Do

- ❖ tProduct, tProductList
- ❖ tName, tId, tIsVisible

Don't

- ❖ product(missing "t")
- ❖ mProduct (same as [Members])
- ❖ TIsVisible (Camel Case)

Implementation (cont.)

Conventions [Loop variables]

- ❖ Uses single lowercase letter
- ❖ The common is to use “i - j - k - l” with the increment order
- ❖ Should not be more than 4 nested loops, otherwise the implementation logic is too complicated and might be wrong

Examples

Do

- ❖ `for (int i ...), for (int j ...)`

Don't

- ❖ `for (int ii ...)`
- ❖ `for (int j ...), for (int i ...)`

Listeners

Conventions

- ❖ Same rule as class naming convention
- ❖ Post-fix with “Listener” (optional)
- ❖ Methods in the listener follow same rule as method naming convention
- ❖ The name should explain what the listener does (action/behavior)
- ❖ Makes it a separated interface file

Examples

Do

- ❖ UniqCompanyLocationUpdateListener

Don't

- ❖ MyListener

Constants

Conventions

- ❖ Starts with company name/project abbreviation
- ❖ All uppercase
- ❖ Each term separated by an “_”
- ❖ Declared as “static final” variables

Examples

Do

- ❖ UNIQCOMPANY_MAX_USER_COUNT
- ❖ UNIQCOMPANY_CONNECTION_TIMEOUT

Don't

- ❖ min_length
- ❖ UNIQCOMPANYMAXUSERCOUNT
- ❖ UniqCompany_Connection_Timeout

Layouts

Conventions

- ❖ Starts with company name/project abbreviation
- ❖ Layout bind with Activity uses prefix “activity”, bind with Fragment uses “fragment”, otherwise uses “layout”
- ❖ Followed by name of binding class or layout purpose
- ❖ All lowercase
- ❖ Each term separated by an “_”

[company name]_[activity|fragment|layout]_[screen]

Examples

Do

- ❖ uniqcompany_activity_home.xml
- ❖ uniqcompany_fragment_session_result.xml
- ❖ uniqcompany_layout_home_header.xml

Don't

- ❖ home_activity.xml (wrong order)
- ❖ fragment_about.xml (missing prefix)
- ❖ layoutListviewFooter.xml (missing “_”)

Values

Conventions

- ❖ Starts with company name/project abbreviation
- ❖ Name itself should explain the purpose
- ❖ All lowercase
- ❖ Each term separated by an “_”
- ❖ Ends with “values”

[company name]_[screen]_[type]_values

Examples

Do

- ❖ `uniqcompany_string_values.xml`
- ❖ `uniqcompany_array_values.xml`
- ❖ `uniqcompany_dimension_values.xml`

Don't

- ❖ `string.xml` (missing prefixes)

Identifiers

Conventions

- ❖ Starts with layout file name (exclude the company name/project abbreviation)
- ❖ Followed by 2 (3) letters describing the view (tv - TextView, bt - Button, et - EditText, rl - RelativeLayout)
- ❖ Ends with purpose (action/behavior)
- ❖ All lowercase
- ❖ Each term separated by an “_”
- ❖ Absolutely no numeric

[activity/fragment/layout]_[screen]_[view]_[function]

Examples

Do

- ❖ activity_home_bt_login
- ❖ fragment_login_bt_signin
- ❖ layout_main_header_tv_username

Don't

- ❖ buttonLogin (Camel Case)
- ❖ username_txt (wrong order)
- ❖ view_title (no screen)
- ❖ textView1 (absolutely NO)

Localization

Conventions

- ❖ Starts with layout file name (exclude the company name/project abbreviation)
- ❖ Followed by indicators:
 - ▶ txt: for normal text content
 - ▶ btn: for button text content
 - ▶ hint: for hint text content
- ❖ All lowercase
- ❖ Each term separated by an “_”
- ❖ Ends with numeric (optional)

[activity|fragment|layout]_[screen]_[indicator]_[view]

Examples

Do

- ❖ activity_home_bt_login
- ❖ fragment_about_txt_section_1
- ❖ fragment_about_txt_section_2
- ❖ layout_main_header_hint_search

Don't

- ❖ hintSearchEditText (Camel Case)
- ❖ username_txt (wrong order)

Thank You