**Здесь описаны правила написания кода, начиная от названия методов и переменных, заканчивая конструированием класса.**

# **Название методов**

Метод следует называть следующим образом:

1. Начинается название с маленькой буквы. (**g**oToService)
2. Все последующие слова с большой буквы.(go**T**o**S**ervice)
3. Первым в названии идет глагол (что делает метод). (**go**ToService, **check**Status)
4. Затем идет то над чем выполняется действие или куда ведет метод. (go**ToService**, check**Status**, check**FreeLogin**).
5. Имя метода должно отражать то, что он выполняет внутри.
6. Методы, отвечающие за передвижения между activity, имеют следующую структуру: goTo + имя activity
7. …

# **Название переменных**

Переменные следует называть следующим образом:

1. Начинается название с маленькой буквы. (**p**assword)
2. Все последующие слова с большой буквы.(user**P**hone)
3. Переменная это не действие, так что никаких глаголов (**~~go, sent …~~**)
4. Булевскую переменную следует называть так, чтобы она отражала состояние TRUE (if(**freeLogin**), if(**status**), if(**~~noStatus~~**))
5. Все переменные типа Boolean изначально имеют значение FALSE
6. Константы следует писать в верхнем регистре (**TAG, ERROR …**)
7. Если константа составная, разделять их нижним подчеркиванием (ERROR**\_**DEL**\_**ZERO)
8. Переменная должна нести смысловой характер! (**~~X~~**, coordinateX)
9. Имена не должны содержать цифры (**~~userPhone1~~**)
10. Следует использовать антонимы (begin/end, first/last)
11. Используй каждую переменную только с 1-ой целью
12. Переменная объекта называется также, как id объекта
13. …

# **Название объектов на layout**

Объекты следует называть следующим образом:

1. Button 🡪 имя кнопки + имя layout + Btn (**findServiceSearchServiceBtn**)
2. Button 🡪 имя кнопки не содержит больше 2-х слов
3. EditText (TextBox) 🡪 имя поля + имя layout + Input (**passwordRegisrationInput**)
4. EditText (TextBox) 🡪 имя поля не содержит больше 2-х слов
5. …

# **Название класса / activity**

Классы следует называть следующим образом:

1. Название класса должно отражать реальный объект (**profile, service, authorization**).
2. Название класса не должно быть действием!
3. Начинается название с маленькой буквы. (**p**rofile)
4. Все последующие слова с большой буквы.(server**S**earch)

# **Структура класса**

Классы следует конструировать следующим образом:

1. Константы
2. Переменные
3. Кнопки
4. EditTexts
5. База данных
6. Работа с файлами
7. Работа с xml-файлами
8. Метод onCreate()
9. Метод onClick()
10. Методы выполняющие изменения в классе
11. Методы перемещения (**goTo**…)

# **Структура класса DBHelper**

Класс отвечающий за базу данных стоит конструировать так:

1. Версия базы данных
2. Имя базы данных
3. Имена таблиц
4. С комментарием (пометкой) перечисления названия строк, который хранятся в базе данных ( // users **public static final** String ***KEY\_PHONE*** = **"phone"**; )
5. Метод onCreate() содержит String переменные, в которые положен текст, создающий таблицу.

String users = **"create table "**+ ***TABLE\_CONTACTS\_USERS*** + **"("** + ***KEY\_ID*** + **" integer primary key,"** + ***KEY\_PHONE*** + **" text,"** + ***KEY\_PASS*** + **" text"** + **")"**;

1. …

# **Правила переноса при написании кода**

Так следует переносить код:

1. При формировании большой строки она имеет следующий формат   
   msg += **"msg" + value  
   + "msg1" + value1**То есть, на каждой новой строке сообщение и ее значение, потом перенос по знаку «+»  
   msg += **" Name = "** + cursor.getString(indexName)  
    + **" Cost = "** + cursor.getString(indexMinCost)  
    + **" Descr = "** + cursor.getString(indexDescription)  
    + **" "**;
2. Если передаваемых параметров больше 4 пишем «в стобик»  
   Cursor cursor = database.query(  
   DBHelper.***TABLE\_CONTACTS\_SERVICES***,  
    **null**,  
    **null**,  
    **null**,  
    **null**,  
    **null**,  
    **null**,  
    **null**);  
   Но  
   registration(database,name,cost,description);

# **Правила работы со строками**

В нашем приложении при сравнении все строки мы опускаем в нижний регистр!

1. Чтобы не сравнивались РАЗНЫЕ строки **N**ame && **n**ame

# **Папка fragments**

Папка fragments хранит классы, которые отвечают за создание динамических view объектов на странице

# **Никогда не хранить значения в БД = null!**

Чаще всего null не может быть преобразована в int или другой тип, поэтому программа ломается, мы решили, если нет значения для данного столбца, то если это int -> 0, если string -> “”(пустая строка), если bool -> false;