Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16, Bratislava 4

Manažment zdravotného stavu pacienta prostredníctvom monitoringu emócií [eMotion]

Metodika pre konvencie písania zdrojových kódov pre Android aplikáciu

Tím: číslo 2, eMotion
Pedagogický vedúci tímu: Ing. Gašpar Peter
Externý vedúci tímu: Ing. Lehocki Fedor, PhD.
Členovia tímu: Bc. Bobotová Zuzana, Bc. Černák Dávid, Bc. Gondová Veronika, Bc. Matlovič Tomáš, Bc. Pavlovič Tomáš, Bc. Šmihla Ján
Akademický rok: 2016 / 2017
Vypracoval: Ján Šmihla
Verzia číslo: 2
Dátum poslednej zmeny: 11.12.2016

1 Úvod

Táto metodika sa zaoberá konvenciou kódu, ktorú je potrebné dodržiavať pri vývoji mobilnej aplikácie. Metodika je záväzná pre všetky zdrojové kódy, ktoré tím vyvíja v rámci mobilnej aplikácie a pre všetkých členov, ktorí sú súčasťou tímu.

Slovník pojmov

- **UpperCamelCase** -spôsob písania viacslovných názvov, s tým, že všetky slová sú spojené bez medzier do jedného a každé slovo začína veľkým písmenom.
- **lowercase_underscore** - spôsob písania viacslovných názvov, s tým, že všetky slová sú spojené podčiarkovníkom (_)
- **memory leak** - únik pamäte, častý problém pri vývoji Android aplikácií. Vzniká, keď sa zabudne odstrániť nejaká referencia na objekt, ktorý už nie je potrebný.

2 Štruktúra projektu

Android projekt sa delí na zdrojové kódy písane v jazyku Java (java) a súbory zdrojov písane v jazyku XML (res).

3 Názvy

Lokálne premenné

Názvy premenných musia začínať malým písmenom. V prípade, že sa názov skladá z viacerých slov, názov má všetky slová spojené bez medzery tak, že každé ďalšie slovo začína veľkým písmenom (localVariable).

Premenné v triedach

Private premenné musia začínať s malým písmenom m a ďalšie písmeno musí byť veľké. Tým označujeme, že sa jedná o class member premennú. (mClassMember).

Konštanty
Konštanty musia mať všetky písmená veľké, slová oddelené podčiarkovníkom (SIMPLE_CONSTANT).

Metódy
Názvy premenných musia začínať malým písmenom. V prípade, že sa názov skladá z viacerých slov, názov má všetky slová spojené bez medzery tak, že každé ďalšie slovo začína veľkým písmenom (simpleFunction()).

Triedy
Názvy premenných musia začínať veľkým písmenom. V prípade, že sa názov skladá z viacerých slov, názov má všetky slová spojené, bez medzery, tak, že každé ďalšie slovo začína veľkým písmenom (SimpleClass).

Balíky
Názvy balíkov musia obsahovať len malé písmená. Treba vyberať len jednoslovne názvy. V prípade, že sa nedá balík pomenovať len s jedným slovom, názov má slová spojené bez toho aby ďalšie slovo začínalo s veľkým písmenom. (simplepackage)

4 Štýly písania

Java

Dĺžka riadku
Riadok musí mať šírku maximálne 150 znakov (na šírku okna kódu,) vrátane odsadzovania a aj komentárov, aby sa nemuselo horizontálne skrolovať. Ako náhle sa limit prekročí, treba čiaru zlomiť alebo napísať kód jednoduchšie.

Výnimkou sú dlhé URL adresy v komentároch, alebo importy.

Dĺžka funkcie
Dĺžka funkcie musí byť vrátane hlavičky maximálne na 100 riadkov, aby sa nemuselo vertikálne skrolovať. Ak počet riadkov presahuje limit, je potrebné funkciu rozbiť na menšie podfunkcie.

Na rozdiel od dĺžky riadku, porušenie tohto pravidla sa dá v špeciálnych prípadoch tolerovať.

Odsadzovanie

Text sa odsadzuje:

- o 4 medzeri v rámci bloku

```
public CalendarAdapter(Context context, List<CalendarEvent> data){
        this.mContext = context;
        this.mData = data;
}
```

- o 8 medzier v rámci lámania čiar

```
CalendarRequest req = new CalendarRequest(
                Utils.getDateNow(),
                PreferenceManager.SENSOR_CALENDAR_ID,
                eventsAll
);
```

Zlom čiary musí byť ešte pred operátorom, výnimkou je operátor priradenia (=).

Zátvorky

Zátvorka sa píše na konci riadku kódu, nie na ďalšom. Toto platí pri všetkých typoch typoch zátvoriek , (), alebo [].

```
if (events != null && events.size() > 0) {
        eventsAll.addAll(events);
}
```

Toto je zle.

```
if (events != null && events.size() > 0)
{
        eventsAll.addAll(events);
}
```

V jednoriadkových podmienkach a cykloch sa zátvorky nemusia písať, ak je telo v rovnakom riadku:
if (milliSeconds == null) return null;

Toto je zle:
if (milliSeconds == null) return null;

XML

Uzatváranie tagov

V prípade, že element neobsahuje nič, treba použiť jednoduché uzavretie tagu />.

5 Pravidlá a rady pri písaní kódu

Výnimky (exceptions)

Ignorácia výnimiek

Riešenie výnimiek je v našom projekte nutnou podmienkou v rámci overovania kódu, pri ich ignorácií hrozí, že overenie neprejde.

Funkcie

Funkcia musí mať vrátane Context a Callbacku maximálne 5 parametrov. Pri prekročení limitu treba funkciu zjednodušiť alebo obaliť viaceré parametre do štruktúry.

Prvý parameter musí byť referencia na Context a posledný referencia na Callback (ak sú potrebné). Na poradí ostatných parametrov medzi nimi už nezáleží.

Funkcia musí obsahovať najviac jeden Callback. V prípade potreby použitia viacerých callbackov, sa jednotlivé metódy Callbackov zlučujú do jedného.

Memory leaks

Stretávanie s týmto problémom je pri vývoji Android aplikácií bežné, vymenujeme si zopár zásad, ktoré treba dodržiavať, aby sme sa im vyhli.

Statické triedy a premenné

Treba sa vyvarovať používaniu statických tried pre ukladanie dát, môže dôjsť k únikom pamäte, ktoré rieši v tomto prípade systém Android tak, že inštancie statických tried jednoducho zabije a potom môže dôjsť k NullPointerException. V prípade potreby treba použiť vzor Singleton, kde pri získavaní inštancií treba overiť, či stále existuje. V každom prípade je zakázané ukladať si do statických premenných referencie na triedy Context, Activity alebo View.

Knižnice tretích strán

Používanie knižníc tretích strán nie zlé, ale musia to byť knižnice určené priamo pre Android, nie pre všeobecnú Javu, aby nedošlo k únikom pamäte.

Logovanie

Pre logovanie sa používa trieda Log. Vo verzii Release logy musia byť vypnuté. Pre zjednodušenie je možné využiť vlastné triedy.

Texty

Texty reťazcov kvôli budúcej internacionalizácii treba vkladať do súboru res/strings.xml a v kóde ich načitávať pomocou funkcie getString(R.string.name_of_string). V XML pomocou skrátky "@string/name_of_string".

Zložité výpočty a formáty

Všetky zložité výpočty a formáty sa musia vykonávať vo vedľajšej funkcii, určite nie priamo v biznis logike. Treba vytvoriť statickú funkciu v triede Utils alebo v inej vhodnej triede.

TODO komentáre

Pri prototypovóm kóde alebo kóde, ktorý daný problém rieši len dočasne, neoptimálne, nepokrýva všetky prípady, treba použiť komentár // TODO. aby sa na daný kód nezabudlo.