

Metodika kódu

November 13, 2021

Autor: Marián Jarník

Úvod

Cieľom tejto metodiky je stanoviť pravidlá pre písanie kódu. Náš tímový projekt nadväzuje na prácu z minulého roka, a z tohto dôvodu sa pri implementácii nových funkcionalít držíme predovšetkým metodík, ktoré boli stanovené predchádzajúcim tímom. Niektoré sme však upravili, aby vyhovovali pre nás zaužívaným praktikám.

Pomenovanie premenných a funkcií

- Názvy premenných, funkcií a komentárov vyskytujúcich sa v kóde sú písané v angličtine
- Viacslovné premenné a funkcie sú písané štýlom camelCase (napr. defaultZoom)
- Všetky premenné majú začínať písmenami
- Enum hodnoty pomenované UPPERCASE

Pravidlá písania if statements

- Jednoduché (jednoriadkové) telo podmienky je tiež uzatvorené v zátvorkách { }
- Otváracia zátvorka { na konci prvého riadku
- Medzera pred otváracou zátvorkou {
- Medzera pred otváracou zátvorkou (

Komentáre

- Jednoduché, zrozumiteľné a len ak sú potrebné
- Zarovnané vždy doľava
- Samotný obsah komentára je oddelený od značky komentára medzerou

- Neuvádzať autora

Názvy súborov

- Java a Kotlin súbory PascalCase, ostatné zdrojové súbory camelCase
- Ostatné súbory (textové, obrázky, ...) oddelené podčiarkovníkom

Iné

- V aritmetických operáciách sú znamienka oddelené medzerami
- Riadok obsahuje maximálne 1 príkaz
- Maximálna dĺžka riadku je 80 znakov
- Ak je to možné, uzatvárať HTML tagy priamo
 - Správne: <myTag />
 - Nesprávne: <myTag></myTag>
- Premenné definovať vtedy, keď sa ich chystáme použiť
 - Oproti definovaniu na začiatku bloku

Spring Boot

- Preferovanie vkladania závislostí (dependency injection) pomocou konštruktora
- Usporiadanie metód v kóde je nasledovné: public protected private
 - Príklad:

```
class Trieda {  
  
    public A() { .. }  
  
    public B() { .. }  
  
    protected X() { .. }
```

```
private ppp(){ .. }
```

```
}
```

Javascript

- Nepoužívať bodkočiarku, pokiaľ nie je potrebná
- Uprednostnenie definovania premenných je v poradí:
 - const
 - let
 - var

Kotlin

- Nepoužívať bodkočiarku, pokiaľ nie je potrebná
- Primárny layout vo fragmentoch je ConstraintLayout
- Uprednostnenie definovania premenných je v poradí:
 - val
 - var

Tabuľka revízie

Dátum	Autor	Opis zmien
17. Dec 2020	Tím 09 – 2020	<ul style="list-style-type: none">• Vytvorenie dokumentu
13. Nov. 2021	Marián Jarník	<ul style="list-style-type: none">• Pridanie úvodu• Pridanie sekcie o Kotline• Úpravy v iných sekciách
14. Nov. 2021	Tím 13 – 2021	<ul style="list-style-type: none">• Revízia metodiky

