Projekto kodo stilius

# Pagrindiniai stiliai

* Kode visur naudojama anglų kalba.
* Pavadinimai neturi viršyti 40 simbolių pavadinimų (išskyrus testavimo funkcijas).
* Kiekviena kodo funkcija turi turėti ‚JavaDoc‘ stiliaus komentarą. Aprašyti pagrindinę funkcijos mintį ir aprašyti gražinamą kintamąjį, parametrų reikalavimus ir jeigu yra ‚default values‘.
* Taip pat jeigu funkcija daro sudėtingą logiką trumpu komentaru aprašyti atliekama žingsnį.

# Java kodo stilius

* Klasių pavadinimai ‚UpperCamelCase‘ pvz.: „ManoKlasė“.
* Funkcijų pavadinimai ‚lowerCamelCase‘ pvz.: „spausdintiTesta“.
* Funkcijų parametrai išdėstomi pagal svarbumą arba panaudojimo greitį t.y. kada pirma karta kreipiamasi į parametrą funkcijoje.
* Naudojama name funkcijų padengimas jeigu metodas galima panaudoti be tam tikrų parametrų t.y. jeigu turime funkciją, kurioje yra ‚optional‘ kintamųjų.
* Jeigu reikia kartoti kodą iškeliame jį kaip atskirą metodą.
* Visi klasių kintamieji yra private/protected naudojama get/set metodai.
* Vidiniai kintamieji turi būti nurodomi su ‚this.‘ priešdėliu.
* ‚Boolean‘ tipo ‚get‘ metodams naudojamas priešdėlis ‚is‘ arba ‚has‘ pagal kontekstą pvz.: isActive(), hasColor().
* Jei reikia daug konstantų sukuriame ‚util‘ tipo klasę, kurioje vienodo tipo konstantos aprašomos kaip public.
* Konstantos turi būti static ir final taip pat visi žodžiai uppercase ir tarpams naudojama ‚\_‘ simbolis pvz.: public static final String KONSTANTA\_A = “a”.
* Kode neturi būti magic konstantų, jeigu reikia string key ar tam tikro skaičiaus aprašome juos kaip klasės private/protected konstantas.
* Kode naudojama ‚design patterns‘, kad kodas būtų suprantamas t.y. pasirinkti gerą implementacijos būdą, kuris butų tinkamas kontekste ir panaudojime. ([žiūrėti](https://sourcemaking.com/design_patterns))
* Kiekvienas programos komponentas turi turėti apibrėžtą ‚interface‘.

Klasės pavyzdys:

public class Clazz {

private String object;

public Clazz(string object) {

this.object = object;

}

public String getObject() {

return this.object;

}

public void setObject(String object) {

this.object = object;

}

}

# ‚Spring framework‘ kodo stilius

* Visos konfigūracijos atliekamos ‚xml‘ tipu.
* Kiekvienos klasės išskyrus ‚DataModel‘ ir ‚Controller‘ tipų turi būti aprašytą kaip ‚spring bean‘.
* Kiekvieno kontrolerio pavadinimas pasibaigia žodžiu ‚Controller‘ pvz.: ‚HomeController‘.
* Puslapio tinklapio šablonai saugomi ‚views/{ControllerName}/{Action}.html‘ pvz.: ‚views/home/index.html‘.
* Naudojama ‚Autowire‘ anotacija gauti reikiamas klasės pasirinktuose kontekstuose.