

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Metodika konvencií písania zdrojového kódu

Tím č.20 – RavensTeam

1. Účel a dedikácia metodiky

V projekte, ktorý je vyvíjaný agilnou metódou (tímom ľudí kde dochádza k častým zmenám v kóde) je potrebné mať presne definované pravidlá písania zdrojového kódu tak, aby bol ľahko pochopiteľný a modifikovateľný pre členov tímu, ako aj pre ostatných vývojárov. Náš projekt je navyše súčasťou väčšieho systému, na ktorého vývoji budú pravdepodobne pracovať aj iné tímy programátorov v budúcnosti.

Táto metodika definuje konvencie písania zdrojového kódu jednotlivu pre programovacie jazyky, ktoré pri vývoji používame. Jej účelom je zachovať udržateľnosť zdrojových kódov vytvorených v rámci projektu. Služi pre súčasných aj budúcich vývojárov systému DevACTs, ktorého je projekt CodeCrutches súčasťou.

Dodržiavanie tejto metodiky je overované v rámci procesu posudzovania kódu.

2. Všeobecné zásady pri písaní zdrojového kódu

Pri vytváraní zdrojového kódu je nutné dodržiavať nasledujúce pravidlá, bez ohľadu na programovací jazyk, v ktorom je kód vytváraný:

- Po každej úprave kódu je potrebné vykonať jeho formátovanie:
 - Visual Studio IDE: `CTRL` + `K` + `D`
 - Eclipse IDE: `CTRL` + `Shift` + `F`
- Používať komentáre
 - Prednostne v syntaxe určenej daným programovacím jazykom pre automatické generovanie dokumentácie
- Výstižné pomenovávanie premenných a elementov
 - Nepoužívať jednopísmenové a nič nevypovedajúce názvy premenných s výnimkou počítadiel
- **Ak nie je v metodike definované inak, používajú sa zaužívané konvencie pre jednotlivé programovacie jazyky:**
 - C# / .NET: <https://msdn.microsoft.com/en-us/library/ff926074.aspx>
 - HTML5: http://www.w3schools.com/html/html5_syntax.asp
 - JavaScript: http://www.w3schools.com/js/js_conventions.asp
 - CSS: <http://cssguidelin.es>

3. Zásady písania zdrojového kódu pre jednotlivé programovacie jazyky

Nakoľko zaužívané konvencie písania zdrojových kódov sa líšia v závislosti od programovacích jazykov, táto kapitola popisuje základné zásady písania zdrojového kódu pre jednotlivé programovacie jazyky využívané pri vývoji projektu CodeCrutches.

3.1. C# / .NET

3.1.1. Model – View – Controller

- **Model**
 - Databázový model
 - Názov: rovnaký ako pomenovanie v databáze v singulári
 - Umiestnenie: ~/Models

- ViewModel
 - Názov: končiaci „ViewModel“ príp. „ViewModels“
 - Umiestnenie: ~/Models
- **Controller**
 - Názov: končiaci slovom „Controller“
 - Umiestnenie: ~/Controllers
- **View**
 - Názov: akcia kontroléru, ktorá daný View renderuje
 - Umiestnenie: ~/Views/názov_kontroléra

3.1.2. Komentáre

Používať automaticky generovanú štruktúru komentárov, ktorá sa zobrazí po troch stlačeniach lomky („///
“).

- **Triedy**

Priamo nad začiatkom definície triedy:

```
/// <summary>
/// popis triedy
/// </summary>
public class IndexViewModel
```

- **Metódy**

Priamo nad začiatkom definície metódy:

```
/// <summary>
/// popis metódy
/// </summary>
/// <param name="tag">popis parametra s názvom „tag“</param>
/// <returns>typ_návratovej_hodnoty : popis</returns>
public static int cislo (int a)
```

- **Ostatné časti kódu**

Komentovať všetky časti kódu, ktoré môžu byť nejasné:

- viacriadkovým komentárom nad danou časťou kódu:

```
/* viacriadkový
 * komentár
 * */
Body body = new Body();
```

- jednoriadkovým komentárom priamo nad daným riadkom kódu:

```
// jednoriadkový komentár
Body body = new Body();
```

3.1.3. Názvy

- používať veľké začiatkové písmeno pri:
 - názvoch tried
 - názvoch metód
- používať malé začiatkové písmeno pri:
 - názvoch premenných
- používať anglické názvy

- používať celoslovné názvy vystihujúce podstatu premennej pri dvoj- a menej slovných názvoch
- používať skratky pri troj- a viacslovných názvoch premenných, kde slovo je dlhšie ako 8 znakov:

```
// nesprávne
UserRegistrationForm userRegistrationForm;

// správne
UserRegistrationForm userRegForm;
```

- jednotlivé slová názvu premennej neoddeľovať podčiarkovníkom, ale veľkým písmenom
- názvy tried podstatným menom
- názvy rozhraní začínať písmenom „I“ (napr. `public interface IProject`)
- názvy súborov definujúcich triedu rovnaké ako názov triedy, ktorú definujú

3.1.4. Typy premenných

- používať preddefinované typy premenných namiesto systémových (`string` namiesto `String`, `bool` namiesto `Boolean` a pod.)

3.1.5. Formátovanie

- pri viacriadkovej časti kódu ohraničenej zátvorkami tieto zátvorky vertikálne zarovnávať:

```
// správne
public static TaskTagHandler Instance
{
    get
    {
        if (_instance == null)
        {
            _instance = new TaskTagHandler();
        }
        return _instance;
    }
}
```

3.2. HTML5

3.2.1. Komentáre

- pri písaní komentárov brať ohľad na to, že sú viditeľné pre používateľa aplikácie
- používať len na miestach, kde je to potrebné, napríklad pri označovaní začiatku a konca elementu:

```
<!--#main start-->
<div id="main">
    ...veľa rôznych elementov...
</div>
<!--#main end-->
```

3.2.2. Názvy

- používať výhradne len názvy a atribúty elementov definované v HTML5 (case-sensitive)
- pokiaľ sa v projekte používa Bootstrap Framework:
 - využívať grid systém Bootstrapu pri tvorení layoutu http://www.w3schools.com/bootstrap/bootstrap_grid_system.asp
 - využívať triedy, ktoré Bootstrap definuje pre jednotlivé elementy:

```

<!--správne-->
<table class="table">
  ...
</table>

<!--nesprávne-->
<table>
  ...
</table>

```

- používať triedy slúžiace na zvýraznenie elementu (*danger*, *warning*,...)
- nerobiť od nuly niečo, čo už je vytvorené v rámci Bootstrapu (panely, carousel, ikony, modálne okná a pod.)
 - kompletný tutoriál dostupný na <http://www.w3schools.com/bootstrap/default.asp>
- názvy tried a identifikátorov s malými písmenami, oddelené pomlčkou:

```

<!--správne-->
<i class="code-crutches" />

<!--nesprávne-->
<i class="codeCrutches" />

```

3.2.3. Syntax

- dodržiavať syntax definovanú HTML5
- výsledný kód validovať prostredníctvom nástroja:
 - <https://validator.w3.org/>

3.3. JavaScript

Nakoľko je v projekte CodeCrutches momentálne využívaný jazyk JavaScript len v malom rozsahu, definujeme len niektoré základné pravidlá pri písaní kódu v tomto jazyku. Podrobnejšie konvencie používané pre JavaScript sú dostupné na adrese http://www.w3schools.com/js/js_conventions.asp.

- pri viacslovných názvoch premenných jednotlivé slová neoddeľovať, ale použiť veľké začiatkové písmeno druhého a ďalšieho slova (*codeCrutches* miesto *code-crutches*)
- ak daný skript nepotrebuje byť vykonávaný skôr ako je načítaný celý dokument (event *Document.ready()*), jeho referenciu pridať až na konci HTML dokumentu priamo nad tag `</body>`
- pri zostavovaní verejnej verzie projektu („*public release*“) vykonať minifikáciu kódu:
 - napr. prostredníctvom nástroja <https://jscompress.com>

3.4. CSS

Táto časť definuje základné pravidlá čisto pre jazyk CSS. Pravidlá pre pomenovávanie tried a identifikátorov HTML elementov popisuje časť [3.2 HTML5](#) tohto dokumentu. Podrobné konvencie zaužívané pre CSS sú dostupné na stránke <http://cssguidelin.es>.

- dodržiavať organizáciu kódu, držať definície patriace k jednému prvku (príp. skupine prvkov) pri sebe
- používať čo možno najpresnejšie selektory:

```

/*nesprávne*/
.menu li {
...
}
/*správne*/
ul.menu>li {
...
}

```

- vyhýbať sa používaniu `!important`, pokiaľ to nie je nutné
- oddeľovať definície patriace k rozdielnym skupinám prvkov (príp. k iným stránkam) prostredníctvom komentárov:

```

/*header*/
div#header {...}
div#header ul.menu {...}
div#header ul.menu>li>a {...}

/*content*/
div.banner {...}
div.sidebar {...}

```

- pri zostavovaní verejnej verzie projektu („*public release*“) vykonať minifikáciu kódu:
 - napr. prostredníctvom nástroja <https://cssminifier.com>