

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Bc. Michal Chylik

Manažment zdrojových kódov
Dokumentácia zdrojových kódov za použitia
phpDocumentor

Predmet: Manažment projektov softvérových a informačných systémov
Študijný odbor: Softvérové inžinierstvo
Školský rok: 2011/2012

Obsah

1. Úvod.....	1
2. Pojmy a skratky.....	1
3. Súvisiace metodiky	1
4. Manažment zdrojových kódov.....	2
4.1. Role a zodpovednosti	2
4.2. Procesy	2
4.2.1. Vytvorenie štábnej kultúry komentovania zdrojových kódov.....	3
4.2.2. Tvorba komentárov zdrojového kódu.....	3
4.2.3. Kontrola komentárov zdrojového kódu.....	3
4.2.4. Generovanie dokumentácie zo zdrojového kódu.....	4
4.2.5. Uchovávanie dokumentácie.....	4
4.3. Stavový diagram zdrojového kódu.....	5
5. Dokumentácia zdrojových kódov v PHPDoc	6
5.1. Značky definované v PHPDoc	6
5.2. Tvorba komentárov zdrojového kódu	8
5.2.1. Vytvorenie základného bloku pre komentár.....	8
5.2.2. Dokumentácia súboru	8
5.2.3. Dokumentácia triedy.....	9
5.2.4. Dokumentácia metódy	9
5.3. Generovanie dokumentácie prostredníctvom phpDocumentor.....	10

1. Úvod

Cieľom metodiky je definovať jednotný postup komentovania zdrojových kódov v projektoch programovaných v jazyku PHP, za účelom automatického generovania dokumentácie. Dokument je určený pre tvorcov zdrojových kódov, ktorí sú povinný riadiť sa definovanými postupmi. Ďalej pre manažéra kvality a manažéra dokumentácie, ktorí majú za úlohu dohliadať na to, či tvorcovia zdrojových kódov dodržia pravidlá stanovené metodikou. Na generovanie dokumentácie sa používa phpDocumentor.

Metodika zachytáva manažment zdrojových kódov na dvoch úrovniach. Vyššia úroveň je zachytená v kapitole 3, nižšia, ktorá určuje konkrétnu syntax komentárov zdrojových kódov, je v kapitole 4.

2. Pojmy a skratky

- PHPDoc – formálny štandard pre komentovanie zdrojových kódov napísaných v jazyku PHP.
- phpDocumentor – open source generátor dokumentácie, ktorý dokáže zo zdrojových kódov napísaných v PHP, v ktorých sa uplatňuje formálny štandard PHPDoc, generovať dokumentáciu.

3. Súvisiace metodiky

- Metodika inštalácie a konfigurácie nástroja phpDocumentor

4. Manažment zdrojových kódov

4.1. Role a zodpovednosti

V Tab.1 sú uvedené identifikované role, ktoré vystupujú v procesoch manažmentu zdrojových kódov a definované ich zodpovednosti.

Rola	Zodpovednosť
Programátor	<ul style="list-style-type: none"> • Tvorba komentárov zdrojového kódu spôsobom určeným metodikou
Hlavný programátor	<ul style="list-style-type: none"> • Tvorba komentárov zdrojového kódu spôsobom určeným metodikou • Vytvorenie a správa štábnej kultúry komentovania zdrojových kódov
Manažér kvality	<ul style="list-style-type: none"> • Zodpovednosť za správne okomentovanie zdrojového kódu • Komunikácia s programátormi
Manažér dokumentácie	<ul style="list-style-type: none"> • Zodpovednosť za generovanie dokumentácie zo zdrojových kódov a jej následné uchovávanie

Tab.1 Role a zodpovednosti

4.2. Procesy

Číslo procesu	Názov procesu	Kapitoly
1	Vytvorenie štábnej kultúry komentovania zdrojových kódov	4.2.1
2	Tvorba komentárov zdrojového kódu	4.2.2
3	Kontrola komentárov zdrojového kódu	4.2.3, 5.2
4	Generovanie dokumentácie zo zdrojového kódu	4.2.4, 5.3
5	Uchovávanie dokumentácie	4.2.5

Tab.2 Procesy

4.2.1.Vytvorenie štábnej kultúry komentovania zdrojových kódov

Vstup	Projekt nemá dokument v ktorom by bola popísaná štábna kultúra komentovania zdrojových kódov
Výstup	Dokument so štábnou kultúrou komentovania zdrojových kódov pre daný projekt
Zodpovedný	Hlavný programátor
Proces	<ul style="list-style-type: none"> Hlavný programátor vytvorí dokument, ktorý podrobne popisuje akým spôsobom je potrebné písať komentáre zdrojového kódu

Tab.3 Proces 1

4.2.2.Tvorba komentárov zdrojového kódu

Vstup	Neokomentovaný zdrojový kód, Zle okomentovaný zdrojový kód
Výstup	Okomentovaný zdrojový kód
Zodpovedný	Programátor
Proces	<ul style="list-style-type: none"> Programátor vloží na príslušné miesta v zdrojovom kóde komentáre podľa určenej štábnej kultúry

Tab.4 Proces 2

4.2.3.Kontrola komentárov zdrojového kódu

Vstup	Okomentovaný zdrojový kód
Výstup	Schválenie okomentovania zdrojového kódu, Označenie okomentovania zdrojového kódu za nedostatočné.
Zodpovedný	Manažér kvality
Proces	<ul style="list-style-type: none"> Manažér kvality skontroluje zdrojový kód, ktorý mu odovzdá hlavný programátor

Tab.5 Proces 3

4.2.4. Generovanie dokumentácie zo zdrojového kódu

Vstup	Schválený okomentovaný zdrojový kód
Výstup	Vygenerovaná dokumentácia
Zodpovedný	Manažér dokumentácie
Proces	<ul style="list-style-type: none"> • Manažér dokumentácie využije jeden z dostupných nástrojov na generovanie dokumentácie pre jazyk, v ktorom bol napísaný zdrojový kód a vygeneruje dokumentáciu zo zdrojového kódu

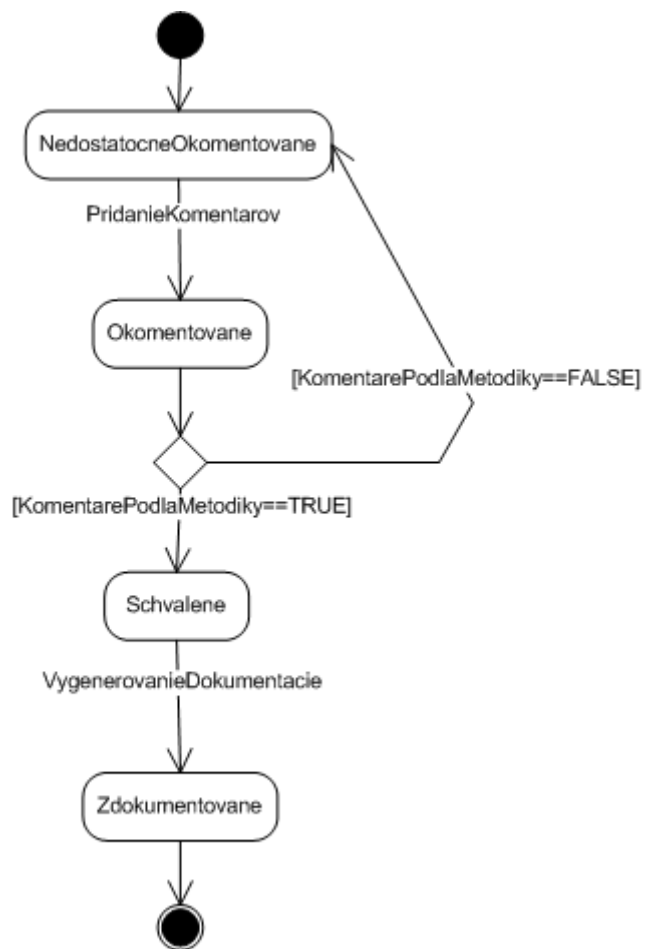
Tab.6 Proces 4

4.2.5. Uchovávanie dokumentácie

Vstup	Vygenerovaná dokumentácia
Výstup	Dokumentácia uložená v úložisku
Zodpovedný	Manažér dokumentácie
Proces	<ul style="list-style-type: none"> • Manažér dokumentácie po vygenerovaní zoberie súbor s dokumentáciou, zistí poslednú verziu súboru v úložisku, podľa dohodnutých pravidiel inkrementuje verziu a pod menom zahrňujúcim verziu, odloží súbor do úložiska

Tab.7 Proces 5

4.3. Stavový diagram zdrojového kódu



Obr.1 Stavový diagram

5. Dokumentácia zdrojových kódov v PHPDoc

Pre dokumentovanie zdrojových kódov v jazyku PHP bol vytvorený formálny štandard PHPDoc. Pri dodržiavaní pravidiel popísaných v PHPDoc je zaručené, že zo zdrojových kódov bude možné automaticky generovať kvalitnú dokumentáciu prostredníctvom nástroja phpDocumentor.

5.1. Značky definované v PHPDoc

PHPDoc definuje veľký počet značiek, ktoré sa využívajú pri tvorbe komentárov a každá z nich má špecifický význam. Všetky značky začínajú znakom @.

Značka	Popis
@access	Riadenie prístupu phpDocumentora k elementu
@author	Meno autora daného elementu
@copyright	Informácie o autorských právach
@deprecated	Elementy, ktoré sú už zastarané a nemali by sa používať
@example	Vloží externý súbor s príkladom definovaný touto značkou
@global	Definovanie a popis globálnej premennej
@ignore	Prinúti phpDocumentor ignorovať daný element
@internal	Súkromná dokumentácia pre vývojový tím
@license	Zobrazí odkaz na licenciu umiestnenú na internete
@link	Zobrazí odkaz na dokument definovaný týmto tagom
@name	Definuje prezývku pre globálnu premennú

@package	Názov balíka
@param	Dokumentácia parametra funkcie
@return	Dokumentácia návratovej hodnoty funkcie alebo metódy.
@see	Dokumentácia vzťahov k iným elementom
@since	Určenie kedy bol element pridaný
@static	Dokumentácia statickej metódy
@staticvar	Dokumentácia statickej premennej
@todo	Dokumentácia vecí, ktoré je potrebné spraviť neskôr
@uses	Zobrazenie linku na dokumentáciu elementu a vytvorenie backlinku v dokumentácii iných elementov na tento
@var	Určenie dátového typu premennej
@version	Určenie aktuálnej verzie elementu

Tab.8 Značky v PHPDoc

5.2. Tvorba komentárov zdrojového kódu

Pre vytvorenie jednotnej dokumentácie v rámci projektu je nutné dodržiavať pravidlá popísané v častiach 4.2.1 až 4.2.4.

5.2.1. Vytvorenie základného bloku pre komentár

Blok pre komentáre podľa štandardu PHPDoc má jednoznačný tvar. Prvý riadok bloku obsahuje presne tri znaky `/**`. Každý ďalší riadok začína znakom `*` a môže obsahovať text alebo značky. Blok je ukončený riadkom s tromi znakmi `**/`.

Ukážka:

```
/**  
 * TU JE KOMENTÁR  
 * TU DRUHÝ RIADOK  
**/
```

5.2.2. Dokumentácia súboru

Každý súbor, ktorý obsahuje zdrojový kód v jazyku PHP bude začínať blokom s komentárom. V príklade je uvedené minimum, ktoré musí obsahovať každý jeden súbor. Pre oddelenie rôznych častí komentára sa používa prázdny riadok. Pre oddelenie v rámci riadku, tabulátor.

Ukážka:

```
/**  
  
 * Krátky popis súboru na jeden riadok  
  
 *  
 * Dlhý popis súboru  
 * počet  
 * riadkov.  
  
 * @package test_package
```

```
* @author    Michal Chylik
* @version   0.0.1
* @since     1.0.1
* @copyright All rights reserved....
**/
```

5.2.3. Dokumentácia triedy

Každá trieda musí v dlhom popise presne informovať o tom, čoho sa týka, čo robí a aká je jej funkcia. Krátky popis slúži na stručný, jednoriadkový, opis triedy. Komentár musí obsahovať aj predpísané značky uvedené v príklade.

Ukážka:

```
/**
 * Krátky popis triedy
 *
 * Dlhý popis triedy
 *
 * @package   triedy_vseobecne
 * @author    Michal Chylik
 * @version   1.0.13
 * @since     3.11.1
 **/
```

5.2.4. Dokumentácia metódy

Každá metóda musí hneď na začiatku obsahovať blok s komentárom. Krátky popis v stručnosti definuje metódu. Dlhý popis vysvetlí aj logiku metódy tak, aby bola zrejmá aj pre iného programátora ako je autor metódy. Ďalej blok obsahuje povinné značky uvedené v ukážke.

Ukážka:

```
/**
```

```
* Krátky popis
*
* Dlhý popis
*
* @author      Michal Chylik
* @param
* @return
**/
```

5.3. Generovanie dokumentácie prostredníctvom phpDocumentor

V systéme s nainštalovaným a nakonfigurovaným nástrojom phpDocumentor vygeneruje manažér dokumentácie prostredníctvom príkazového riadku dokumentáciu vybraných súborov.

Na vygenerovanie automatickej dokumentácie je potrebný jeden jednoriadkový príkaz phpdoc s parametrami. V Tab.9 je zoznam najdôležitejších parametrov.

Parameter	Názov	Popis
-f	-filename	Názov súboru, ktorý sa dokumentuje
-d	-directory	Názov adresára, ktorý sa dokumentuje
-t	-target	Určenie cesty, kam sa uloží dokumentácia
-ti	-title	Explicitné uvedenie názvu pre vygenerovanú dokumentáciu
-o	-output	Určenie formátu výstupu dokumentácie

Tab.9 Parametre príkazu phpdoc

Dokumentáciu vygenerujeme príkazom phpdoc za použitia prepínačov. Je nutné zadať minimálne prepínač určujúci zdrojový kód, ktorého dokumentáciu ideme generovať a prepínač určujúci miesto uloženia dokumentácie.

 **SiteController**

Description

[Description](#) | [Vars](#) | [Methods \(details\)](#)

Controller is the customized base controller class.

All controller classes for this application should extend from this base class.

Located in `/controllers/SiteController.php` (line 3)

```
CController
|
|--Controller
|
|--SiteController
```

Method Summary

[Description](#) | [Vars](#) | [Methods \(details\)](#)

```
void actionAjaxPersonRequest ()
void actionContact ()
void actionError ()
void actionGetListOfSkills ()
void actionIndex ()
void actionInsertUserSkill ([ $_FUN_ID_USER = 0])
void actionLogin ()
void actionLogout ()
void actions ()
void actionShowUserSkills ()
void getSkillDataList ()
void getSkillsOfUser ([ $_FUN_ID_USER = 0])
```

Obr.2 Ukážka vygenerovanej dokumentácie