Dokumentace úlohy SYN: Zvýraznění syntaxe v PHP do IPP 2015/2016
Jméno a příjmení: Michal Ondrejó
Login: xondre08

Dokumentace k projektu IPP 2015/2016, SYN: Zvýraznění syntaxe

Analýza zadání

Cílem tohoto projektu bylo vytvořit skript v jazyce PHP, který bude zvýrazňovat části textu podle zadaného regulárního výrazu. Zvýraznění je realizované pomocí šesti HTML elementů: tučný text, kurzíva, podtrhnutí, teletype, velkost textu a barva textu. Kromě toho je možné spustit skript s parametrem „--br“ který za každý řádek přidává HTML element
.

Postup řešení

Po spuštění skriptu se nejprve zkontrolují zadané parametry. Nejprve se zkontroluje, jestli je první parametr `help` a jestli se za ním nenachází další parametr. Další parametry jsou testované zvlášť v cyklu, který prochází jednotlivé argumenty. Při nalezení parametru „--br“ se aktivuje přepínač `$BR` a při parametrech „--input“, „--output“ a „--format“ se pokusí dané soubory otevřít.

Nejprve se načte celý vstupný soubor do proměnné `$textNaFormatovanie`. Následně se formátovací soubor prochází po řádcích, kde se na začátku řádku nachází regulární výraz, který je ale v jiném formátu jako regulární výraz používaný při funkci `preg_match_all()` a proto je třeba ho překonvertovat. Po překonvertovaní regulárního výrazu už může pomocí funkce `preg_match_all()` vyhledat všechny části textu, které vyhovují danému regulárnímu výrazu.

Následně procházíme tyto nalezené části textu, ale protože nevíme, na jaké pozici se přesně nachází, je nutné, abychom začali hledat od konce naposledy nalezeného řetězce. I když už víme kde se nachází začátek a konec, stále nemůžeme vložit dané elementy přímo na nalezenu pozici, protože by to mohlo bránit při hledání následujících regulárních výrazů. Proto si uložíme pozici a typ formátovaní do polí `$poleNajdenychPozicia` a `$poleNajdenychTyp`.

Když projdeme všechny řádky ve formátovacím souboru, nic nám už nebrání a konečně můžeme vložit elementy do proměnné `$textNaFormatovanie`. Vkládaní ale musíme realizovat od konce, což nám zaručí, že všechny elementy se vloží na své správné místo a žádný element se nevloží do jiného elementu.

Nakonec už jen musíme podle stavu přepínače `$BR` vložit element `<br />` na konce řádků a můžeme výsledný text poslat na výstup nebo uložit do souboru.

Regulární výraz

Samotná iplementace překonvertováni regulárního výrazu bylo asi tou nejsložitější částí. Regulární výraz, se prochází znak po znaku a postupně se do proměnné `$regularExpression` přidává ekvivalent daného znaku. Při tom je třeba dávat pozor na to, zda v něm není chyba jako například špatný počet závorek, negace na konci výrazu a či není v negaci více jako jeden znak.

Při vícenásobné negaci je třeba zjišťovat počet negací. Při lichém počtu negací se vkládá začátek negace: `[^` a při sudém počtu se vkládá konec negace: `]`. Když se mezi negacemi neobjevil žádný znak kromě závorek, tak se závorky odstranily.

V regulárním výrazu se mohl vyskytnut libovolný znak v negaci: !%a. Když se takovýto znak vyskytne, přepne se přepínač `$lubovolnyZnakVNegacii`, který když je zapnutý, tak se nezačne vyhledávaní.

Rozšíření NQS

Implementace tohoto rozšíření byla celkem jednoduchá. Stačilo zkontrolovat, či se za znaménkem `+` nachází znaménko `*`. Když bylo nalezeno, tak se do výsledného regulárního výrazu vložilo `*`. Při znaménku `*` nebylo třeba kontrolovat jestli se za ním nachází `+` nebo `*`, protože se rovnou mohlo vložit `*`.