Dokumentace úlohy SYN: Zvýraznění syntaxe v PHP 5.6 do IPP 2016/2017
Jméno a přijmení: Juraj Kubiš
Login: xkubis15

1.1 Analýza úlohy

Na začátku jsem si zadaný úkol rozdělil na několik částí. Konkrétně to bylo zpracovaní argumentů, zpracovaní formátovacího souboru, validace a převod regulárních výrazů do formátu PCRE *(Perl Compatible Regular Expression)*, vkládání HTML značek na správná místa a případná implementace rozšíření.

1.2 Řešení problému

Až na zpracování argumentů jsem tenhle projekt implementoval objektově.

1.2.1 Zpracování argumentů

Tahle část projektu je implementována ve funkci `processArguments()`. Základem je použití funkce `getopt()`, která vrací tabulku povolených přepínačů a jejich hodnot. Pokud se počet záznamů v tabulce liší od počtu argumentů, byl zadán buď neznámy přepínač, nebo byl některý přepínač zadán vícekrát. V obou případech se ale jedná o chybu. Následně testuji nesprávnou kombinaci přepínačů, konkrétně použití přepínače `--help` s jinými přepínači.

1.2.2 Zpracování formátovacího souboru

Formátovací soubor je načten do paměti, kde je reprezentován instancí třídy `FormatList`. Tahle třída je inicializována pomocí metody `initFromFile()`, která jako parametr dostane cestu k formátovacímu souboru.

Soubor je zpracováván po řádcích, které jsou nejdřív kontrolovány regulárním výrazem, jestli odpovídají předepsanému formátu. Prázdné řádky jsou ignorovány.

Následně je řádek rozdělen na regulární výraz a formátovací příkazy. Ty jsou vkládaný do tabulky, kde klíčem je regulární výraz a položkami jsou seznamy příkazů.

Pokud je cesta neplatná, zůstane tabulka po inicializaci prázdna.

1.2.3 Validace a převod regulárních výrazů

Každý regulární výraz je reprezentován instancí třídy `Regex`. Parametr konstruktoru je regulární výraz ve formátů ze zadání. Po invokaci metody `convert()` objekt obsahuje původní regex a jemu odpovídající PCRE regex.

Konverze je založená na konečném automatu, který čte regulární výraz znak po znaku. Pokud se automat dostane do stavu, že přečtený sled znaků odpovídá elementárnímu regulárnímu výrazu reprezentújíci jeden znak v přijímaném řetězci a nebo operátoru, je do pomocného seznamu přidán ekvivalentní elementární regulární výraz ve formátu PCRE.

Následně je pomocný seznam zkontrolován, jestli se v něm nenachází zakázaná kombinace operátorů. Pro každou dvojici po sobě jdoucích elementárních výrazů je vyhodnocena validita jejich vzájemného postavení na základe jednoduché tabulky.

Výraz je pak převeden na řetězec a naposled validován – je vložen do funkce `preg_match()` a na základě návratové hodnoty je rozhodnuta validita výrazu.

1.2.4 Vkládání HTML značek

Dokument je reprezentován instancí třídy `document`, inicializace probíhá pomocí metody `initFromFile()`.

V každé instanci je kromě reprezentace samotného textu dokumentu i tabulka výskytů regulárních výrazů v dokumentu. Záznam do téhle tabulky je přidán pomoci metody `findRegexMatchPositions()`, kde parametrem je regulární výraz. Klíčem do tabulky je regulární výraz a záznam je seznam dvojic (začátek a konec shody s regulárním výrazem).

Vkládaní je založeno na dvojitém průchodu dokumentu. Při prvním průchodu je nejprve naplněna tabulka výskytů všech regulárních výrazů. Následně, při druhém průchodu, jsou na pozice vkládaný otevírací a zavírací značky. Po každém vložení značky do textu jsou pozice v tabulce aktualizovány (pozice za vloženou značkou jsou inkrementovány o délku téhle značky).

Případné vkládaní značek `<br />` je řešeno až nakonec a to prostým nahrazením řetězce `\n` řetězcem `<br />\n`.

1.2.5 Rozšíření NQS

Rozšíření je implementováno prostým nahrazením sledu libovolného počtu operátorů `+` a `*` pouze jedním operátorem. Pokud sled obsahuje jenom `+`, je nahrazen jedním `+`, jinak je tenhle sled nahrazen jednou `*`.