Dokumentace úlohy SYN: Zvýraznění syntaxe v PHP 5.6 do IPP 2016/2017
Jméno a přijmení: Jan Kubica
Login: xkubic39

1 Dekompozice úlohy

Zadaný projekt je rozdělen celkem do pěti souborů. Kromě stanoveného souboru `syn.php` se pracuje i se soubory v podadresáři `src`, kde se dále nachází `Arguments.php`, `RegexConvertor.php`, `FormatArray.php` a `Error.php`. Zvlášť je tedy řešeno zpracování argumentů při spuštění programu, konverze původních regulárních výrazů na regulární výrazy typu PCRE *(Perl Compatible Regular Expression)*, generování zformátovaného výstupního souboru s HTML značkami a ošetření chybových stavů.

2 Přístup k řešení

Objektově byly zpracovány vstupní argumenty a postupné formátování výstupního souboru, převážně strukturálně pak byl řešen převod regulárních výrazů a práce s chybovými výstupy.

2.1 Zpracování argumentů

Pro práci s argumenty je využita třída `Arguments`. Jejím základem je metoda `parseArg()`, která postupně zpracovává seznam hodnot zadaných při spuštění programu a nastavuje tak příslušné příznaky, včetně příznaku pro generování nápovědy, a ukládá cesty pro vstupní, výstupní a formátovací soubory. Je umožněno zadat každý povolený argument pouze jedenkrát, vícenásobné zadání vede k chybě na standardní chybový výstup. Dle zadání je možno používat taktéž zkrácené argumenty tvaru `-i=filename`, `-o=filename`, atd. Následně je kontrolováno, zdali je program spuštěn buď s přepínačem pro zobrazení nápovědy, nebo pro formátování HTML značek.

2.2 Zpracování formátovacího souboru

Pokud je zadána platná cesta k formátovacímu souboru, je daný soubor načten do paměti jako pole řádků pomocí funkce `parseInputFile()`, přičemž je po kontrole rozdělen zvlášť na část tvořenou regulárními výrazy, které pak přísluší část formátovacích příkazů. Prázdné řádky formátovacího souboru jsou ignorovány. Při neplatném vstupu zůstává proměnná prázdná.

V případě úspěchu je první část s regulárními výrazy je v dalším kroku zpracovaná funkcí `regexConv()`.

2.3 Kontrola a převod regulárních výrazů

Původní regulární výrazy z formátovacího souboru jsou nekompatibilní s regulárními výrazy formátu PCRE a funkcemi v jazyce PHP, proto je potřeba zajistit automatizovaný převod. Každý regulární výraz je převeden na pole znaků a postupně znak po znaku je vyhodnocen ekvivalentní výraz formátu PCRE. Samotná konverze by se dala přirovnat k funkci konečného automatu. Pokud se při přečtení celého vstupu znaků, jakožto jednoho regulárního výrazu, dospěje k platnému koncovému stavu, je regulární výraz přijat a výstupní výraz je tudíž použitelný pro funkce jazyka PHP. Zakázané kombinace operátorů, speciálních symbolů a závorek jsou vyhodnoceny chybovým stavem. V důsledku je tedy každý řádek formátovacího souboru v paměti tvořen dvojicí - platný regulární výraz a formátovací příkaz.

2.4 Kontrola formátovacích příkazů a vkládání HTML značek

V souboru `FormatArray.php` je v rámci funkce `loadFormatTags()` nejdříve ověřena validita formátovacích příkazů a jejich správné oddělení čárkami, mezerami či tabulátory. Pokud je mezi formátovacími příkazy neznámý příkaz nebo jsou formátovací příkazy nepřesně rozděleny, je program ukončen chybovým hlášením. Poté když je vše platně zadáno, je přistoupeno k zapisování HTML značek do proměnné určené pro výstupní soubor. Jeden po druhém jsou brány formátovací příkazy a pomocí vestavěné funkce `preg_match_all()` jsou nalezeny veškeré pozice příslušného regulárního výrazu v textu vstupního souboru. Tyto data jsou reprezentována ve formátu počáteční index - koncový index. Vstupní soubor je v rámci konstruktoru třídy `FormatArray` převeden na pole jednotlivých znaků, ke kterým se pak pomocí nalezených indexů přistupuje. K daným znakům je pak připsána HTML značka, zleva pro počáteční tag, zprava pro koncový. Případné vkládaní značek `<br />` je řešeno až v konečné fázi, a to prostým nahrazením řetězce „`\n`“ řetězcem „`<br />\n`“.

Výstupní proměnná je pak buď vložena do výstupního souboru pomocí funkce `makeOutputFile()`, nebo je vypsána na standardní výstup.