**Dokumentace úlohy SYN: Zvýraznění syntaxe**

**Jméno a příjmení: Dominik Skála**

**Login: xskala11**

Blabla

Celý skript pracuje na základě jednoduché kostry, která se skládá z několika částí: zpracování vstupu, kontrola přítomnosti a zpracování formátování, načtení vstupu, formátování vstupu, tisk vstupu.

Každá z těchto částí je zapsána jednou či více funkcemi, které jsou volány uvnitř jednotlivých souborů, případně jsou volány přímo kostrou skriptu, níže bude každá z nich blíže popsána.

Jak již byl zmíněno výše, skript má určitou kostru, ta je nastíněna v hlavním souboru skriptu s názvem: „syn.php“. Prvním krokem je zahrnutí tří pomocných souborů obsahujících externí funkce, potřebné pro práci skriptu. Následně je provedena kontrola vstupních argumentů skriptu, pokud vše proběhne v pořádku, dojde k načtení formátovacího souboru, následně dojde také k jeho rozboru. Provede se kontrola validity jednotlivých regulárních výrazů a formátovacích značek HTML. Po rozboru se načte všechen potřebný vstup, ať už ze standardního vstupu terminálu, či z vstupního souboru. Provede se formátování vstupu podle rozboru, následně se tento text uloží do zadaného výstupního souboru, či se vytiskne na standardní výstup terminálu.

Prvním bodem je kontrola argumentů, to obstarává funkce checkArguments() ze třídy Common. Tato funkce obdrží na vstupu pole argumentů, které jazyk PHP samo vytváří při spuštění každého skriptu. Následně se ověří počet těchto argumentů, zda vyhovuje zadání, za využití funkce preg\_match se regulárním výrazem zajistí kontrola vstupních argumentů – které byly předány a které ne. Pokud některý chybí či byl zadán vícekrát, skript se zachová adekvátně vzhledem k zadání. Následně se provede za účelem zrychlení skriptu kontrola na přítomnost vstupního a výstupního souboru. Pokud je skriptu předán formátovací soubor, který neexistuje, skript toto nezajímá.

Dalším krokem je následné načtení formátovacího souboru a jeho rozbor. To řeší funkce parseFormatFile() obsažena ve třídě Parser. Jejími vstupními parametry je samotná třída a zadaný formátovací soubor. Pokud k zadání formátovacího souboru nedojde, formátování se přeskočí a funkce vrací prázdné pole formátovacích prvků.

Pokud je však formátovací soubor zadán, provede se za pomocí operace foreach() průchod souboru po řádcích, kdy dochází podle regulárního výrazu: „([\w\d"\.\|\!\%\\*\+\(\)]+[\t]+[\w\d,: \t]+)$“ ke kontrole validity jednotlivých řádků, zda obsahují znaky, které obsahovat mají. Následně se provede rozdělení do pole polí $formats[$i] podle všech mezer (tabulátory, mezery). Jakmile je toto pole naplněno, dojde k průchodu tohoto pole a každý řádek je následně zpracován příslušnými funkcemi. Jednotlivé regulární výrazy jsou zpracovány funkcí parseRegex(), HTML tagy parseTag().

Samotný rozbor HTML tagů je triviální, pouze dojde ke kontrole každého prvku pomocí funkce preg\_match(), zda obsahuje tag, který je specifikací dovolen. Pokud obsahuje jiný, či stejný tag, ale v jiném formátu, než dovoluje specifikace, dojde k chybě a ukončí se skript.

Rozbor regulárních výrazů je již zábavnější ovšem při použití funkce preg\_match() a preg\_replace() jde opět a pouze jen o triviální automat. Zároveň však stojí za zmínku, že funkce preg\_match() a preg\_preplace() vcelku jednoduše umožňují implementaci rozšíření NQS, které má zajistit zkrácení zápisu regulárních výrazů. V tomto případě se opravdu jednalo pouze o 6 řádků podmínek s nahrazením textu.

Ještě před tiskem formátovaného vstupu je třeba ověřit, zda nebyl zadán na vstup I argument “—br”. Pokud byl zadán, na konec každého řádku se vloží string: “<br />”. Toto je řešeno jednoduchým rozdělením formátovaného vstupu po nových řádcích, tedy „\n“, a následným „slepením“ všech řádků za pomocí „<br />\n“.

Tisk do souboru je řešen funkcí writeToFile(). Její alternativou je writeToStdout(). Obě funkce dělají prakticky to samé, funkce writeToFile() pouze navíc otevírá pointer na soubor zadaný. Krátkou kontrolou by obě funkce šly spojit do jedné, o něco větší. Rozhodl jsem se však obě funkce oddělit, aby každá sloužila svému účelu.