

# Dokumentace úlohy SYN: Zvýraznění syntaxe v PHP 5 do IPP 2014/2015

Jméno a příjmení: Štěpán Granát

Login: xgrana02

## Zadání

Cílem bylo vytvořit program v jazyce PHP 5, který aplikuje zvýraznění syntaxe (HTML značkami) na vstupní soubor podle formátovacího souboru. Je podporováno i rozšíření HTM, které validuje výsledný HTML kód a také umožňuje escapovat HTML značky již obsažené ve vstupním souboru.

## Návrh a realizace

Úkol jsem si rozdělil do několika tříd, které navzájem spolupracují.

### Zpracování argumentů (třída Arguments)

Argumenty jsou zkontrolovány a objektově uspořádány. Kontrolují se chyby vyplývající ze zadání. Pokud jsou argumenty v pořádku, zkontroluje také zadané soubory (zda existují a je možné otevřít).

### Zpracování formátovacího souboru (třída FormatFileParser)

Formátovací soubor rozdělí na jednotlivé řádky s regulárními výrazy a formátovacími značkami. Kontroluje validitu regulárního výrazu a převede jeho syntax na standardní syntax PCRE. Převedení je realizováno pomocí konečného automatu, který také kontroluje některé chyby. Výsledkem je pole formátovacích značek, které obsahuje objekty třídy FormatUnit (obsahují regulární výraz a HTML formátovací značky).

### Formátování textu (třída SourceFormatter a FormatMark)

Aplikuje formátování na vstupní soubor. Protože formátování musí být nezávislé na již obsažených HTML značkách (berou se jako normální text), musel jsem zahrnout variantu řešení s postupným vkládáním značek ve vlnách. Nakonec jsem navrhl následující řešení:

1. Nachystáme si prázdné pole Q objektů FormatMark, která představuje budoucí vložení jedné formátovací značky.
2. V cyklu projdeme všechny vyhovující řetězce a do pole Q přidáme záznam o budoucím vložení formátovacích značek na začátek a konec (do textu je zatím nevkládáme).
3. Pole Q seřadíme podle indexu a dále podle typu formátovací značky a vlny (vyplývající z formátovacího souboru) ve které byla vytvořena.
4. Procházíme pole Q a zároveň zdrojový soubor a výsledkem je výsledný naformátovaný soubor (značky tedy vložíme v jediném průchodu).

Pro nalezení řetězců vyhovujících regulárnímu výrazu je využita standardní funkce PHP5 preg\_match\_all.

### Rozšíření HTM (třídy HtmlParser a HtmlScanner)

Pro validaci výsledného souboru jsem použil algoritmus, který by měl odpovídat postupu ze zadání. Pro zjednodušení jsem implementoval jednoduchý scanner, který načítá HTML značky a rozlišuje, jestli jde o text, otevírací tag nebo uzavírací tag.

1. Připravíme zásobníky OPEN a SUSPEND a výsledný řetězec RET
2. Načteme HTML tag pomocí scanneru
3. Rozlišujeme, o který typ jde:
  - a. **otevírací tag:** Přidáme ho do zásobníku OPEN a přidáme do RET
  - b. **uzavírací tag:**
    - i. **Pokud je v OPEN:** Procházíme zásobník OPEN a všechny tagy, které se nachází před uzavíraným tagem uzavřeme a přesuneme do zásobníku SUSPEND. Uzavíraný tag odstraníme ze zásobníku OPEN
    - ii. **Pokud je v SUSPEND:** Tag zahodíme a odstraníme jej ze SUSPEND
  - c. **text:** Projdeme SUSPEND všechny přidáme do RET a do OPEN. Přidáme text do RET

- d. **konec soboru:** Končíme
4. Pokračujeme bodem 2.

## Příklad validace html

Vstupní řetězec: `<i>a<tt>b<b>c</i>def</tt></b>`

Vstup	RET	OPEN	SUSPEND
<code>&lt;i&gt;</code>	<code>&lt;i&gt;</code>	<code>&lt;i&gt;</code>	
<code>a</code>	<code>&lt;i&gt;a</code>	<code>&lt;i&gt;</code>	
<code>&lt;tt&gt;</code>	<code>&lt;i&gt;a&lt;tt&gt;</code>	<code>&lt;tt&gt;&lt;i&gt;</code>	
<code>b</code>	<code>&lt;i&gt;a&lt;tt&gt;b</code>	<code>&lt;tt&gt;&lt;i&gt;</code>	
<code>&lt;b&gt;</code>	<code>&lt;i&gt;a&lt;tt&gt;b&lt;b&gt;</code>	<code>&lt;b&gt;&lt;tt&gt;&lt;i&gt;</code>	
<code>c</code>	<code>&lt;i&gt;a&lt;tt&gt;b&lt;b&gt;c</code>	<code>&lt;b&gt;&lt;tt&gt;&lt;i&gt;</code>	
<code>&lt;/i&gt;</code>	<code>&lt;i&gt;a&lt;tt&gt;b&lt;b&gt;c&lt;/b&gt;&lt;/tt&gt;&lt;/i&gt;</code>		<code>&lt;tt&gt;&lt;b&gt;</code>
<code>def</code>	<code>&lt;i&gt;a&lt;tt&gt;b&lt;b&gt;c&lt;/b&gt;&lt;/tt&gt;&lt;/i&gt;&lt;tt&gt;&lt;b&gt;def</code>	<code>&lt;b&gt;&lt;tt&gt;</code>	
<code>&lt;/tt&gt;</code>	<code>&lt;i&gt;a&lt;tt&gt;b&lt;b&gt;c&lt;/b&gt;&lt;/tt&gt;&lt;/i&gt;&lt;tt&gt;&lt;b&gt;def&lt;/b&gt;&lt;/tt&gt;</code>		<code>&lt;b&gt;</code>
<code>&lt;/b&gt;</code>	<code>&lt;i&gt;a&lt;tt&gt;b&lt;b&gt;c&lt;/b&gt;&lt;/tt&gt;&lt;/i&gt;&lt;tt&gt;&lt;b&gt;def&lt;/b&gt;&lt;/tt&gt;</code>		

Výsledný řetězec: `<i>a<tt>b<b>c</b></tt></i><tt><b>def</b></tt>`

## Testování

Vytvořil jsem testovací sadu, kterou jsem dal veřejně k dispozici. Díky tomu jsem také opravil nemálo chyb, na které mě upozornili ostatní spolužáci.

## Zjednodušené schéma programu

