Dokumentace úlohy SYN: Zvýraznění syntaxe v Python 3 do IPP 2011/2012
Jméno a přijmení: Pavol Loffay
Login: xloffa00

1.1 Analýza úlohy

V této úloze jsem na začátku identifikoval tyto významné úlohy. Zpracování parametrů skriptu a formátovacího souboru, kontrola regulárních výrazů a úprava pro jazyk *Python 3* a následná aplikace regulárních výrazů a vkládání formátovacích značek do výsledného textu.

1.2 Návrh řešení problému

Úlohu jsem rozdělil do dvou částí. První část zahrnuje zpracování formátovacího souboru s kontrolou a úpravou regulárních výrazů. Druhá část se stará o nalezení indexu, na který se mají vložit formátovací řetězce a jejich následné vkládání.

Další kapitoly popisují jednotlivé problémy. Jejich pořadí je pro korektní běh skriptu nutné dodržet.

1.2.1 Zpracování parametrů

Skript je možné ovládat různými parametry, proto jsem se rozhodl použít modul `getopt`, který jejich zpracování výrazně ulehčuje.

1.3 Popis řešení

1.3.1 Zpracování formátovacího souboru

Po otevření souboru se jeho obsah načítá po řádcích do proměnné pomocí metody `readlines()`. Tato metoda vrací seznam, kde každou položkou je jeden řádek. Pro odlišení regulárního výrazu od seznamu formátovacích parametrů jsem použil metodu `split()`, která rozdělí řetězec na podřetězce podle zadaného znaku. Tato metoda znova vrací seznam, kdy v tomto seznamu se regulární výraz vždy nachází v prvním prvku. Zpracování formátovacích příkazů sa provádí analogickým způsobem.

Formátovací příkazy jsou ihned transformovány na ekvivalentní HTML značky, přičemž se vytvářejí začáteční a koncové, takže jsou vždy vytvořeny dva řetězce. Jeden, který obsahuje počáteční značky a druhý koncové. Tímto jsem docílil, že ke každému regulárnímu výrazu jsou připravené řetězce, které se vloží na počáteční a koncovou pozici.

1.3.2 Kontrola regulárního výrazu

Z důvodu, že uživatel může zadat nesprávný regulární výraz, je nutné jej kontrolovat. Samotné ověření výrazu spočívá v kontrole, zda-li jsou speciální symboly `.|!*+()%`, zadané správným způsobem. Kontrola se provádí, vyhledáváním zakázaných kombinací výše uvedených znaků. Některé se také nesmí nacházet na začátku, nebo na konci výrazu. Dále bylo nutné ověřit, zda-li jsou všechny závorky ukončeny.

1.3.3 Úprava regulárního výrazu

Po kontrole správnosti regulárního výrazu jej bylo nutné převést na ekvivalentní výraz pro jazyk *Python 3*. Nejprve byl celý escapován metodou `escape()`. Následně se zpětně od-escapovali znaky `|*+()`, protože mají shodnou sémantiku. Úprava ostatních částí regulárního výrazu spočívala v záměně některých znaků. Například `!A`, za `[^A]` nebo odstranění všech výskytů znaku '`.`'.

Kontrola a úprava regulárních výrazů probíhá v části, kde se zpracovává formátovací soubor.

1.3.4 Aplikace regulárních výrazů

Upravené regulární výrazy, byly následně použity pro vyhledávání částí vstupního textu, které mají být označeny. Pro vyhledávání jsem použil metodu `search()`, která vrátí objekt, na který se dále aplikují následující metody. Pomocí metod `start()` a `end()` se zjistí pozice, na které se mají vložit formátovací řetězce. Nalezené pozice se ukladají do tabulky s rozptýlenými položkami, kde klíč je index v textu, který je označován a hodnota je řetězec, který se na daný index vloží. Počáteční HTML značka se pak vloží na počáteční index a koncová na koncový index.

1.3.5 Vkládaní formátovacích značek

Formátovací značky s indexy, na které mají být vloženy jsou uloženy v tabulce s rozptýlenými položkami. HTML značky je nutné vkládat do textu od posledního indexu k prvnímu. Proto se vytvoří pomocný seznam, který obsahuje pouze klíče tabulky. Ten se následně seřadí a může dojít k vkládaní formátovacích řetězců do výsledného textu.

1.3.6 Parameter `--br`

Vkládaní značky `<br />` na konec řádku probíhá jako poslední fáze úpravy výsledného textu.