Dokumentace úlohy SYN: Zvýraznění syntaxe v Python 3.6 do IPP 2016/2017

Jméno a příjmení: Václav Martinka

Login: xmarti76

1. Obecné řešení zadání

Zadání programu je spuštění regulárních výrazů z jazyka IFJ nad zadaným souborem a obalení výsledků HTML tagy. Proto je řešení postaveno na maximálním využití knihovny `re` (regulární výrazy) jazyka Python.

O správné načtení argumentů se stará funkce `reag_args` ze souboru `functios.py`. Protože jazyk Python nezná příkaz `switch`, je místo něj použita kombinace `if` a `elif`. Načtené informace se ukládají do objektu `options` stejnojmenné třídy (viz níže) a zároveň se provádí jejich validace.

2. Struktura skriptu

Celý skript je postaven na dvou rozsáhlejších třídách, a to `Options` ze souboru *functions.py* a `Format` ze souboru *format.py*. Všechny soubory, kromě hlavního souboru *syn.py*, jsou umístěny ve složce *lib*.

- **Třída `Options`**

  Tato třída slouží pro uchování hodnot načtených z příkazové řádky, jedné se vlastně jen o společný „obal“ pro jednotlivé proměnné. K atributům se přistupuje přímo, což sice není čisté objektové řešení, ale kód je přehlednější. Třída obsahuje jen jednu metodu, a to pro otevření jednotlivých souborů, popř. `stdin`/`stdout`.

- **Třída `Format`**

  Tato třída reprezentuje vstupní formátovací soubor. Funkce `parse_format_file( options )` se stará o naplnění daty a zároveň provádí validaci formátovacího souboru.

  Součástí třídy jsou i podtřída `FormatRecord`, která obsahuje ještě podtřídu `HtmlTag`.

  Kromě toho je `Format` tvořen jen jedním atributem, a to seznamem objektů podtřídy `FormatRecord`, pro uchování jednotlivých formátovacích záznamů (jednotlivé řádky ve formátovacím souboru). K tomuto atributu se vážou dvě metody `add` (kromě přidání provádí i validaci) a `get`.

  - **Podtřída `FormatRecord`**

    Tuto podtřídu tvoří dva atributy. Regulární výraz uložený ve stringu a seznam formátů. K těmto atributům jsou zde patřičné metody pro načtení/uložení dat z/do atributu, které provádí i validaci vkládaných dat.

    Jednotlivé formáty jsou reprezentovány objekty podtřídy `HtmlTag`, která obsahuje 2 atributy:
    - HTML tag (uchován formou `enum`)
    - jeho hodnotu (např. barvu), která může být i `None`.

    K těmto atributům se přistupuje metodou `tag`, která tyto informace převede na validní HTML tag.

- **Soubor `functions.py`**

  Tento soubor obsahuje všechny potřebné funkce. Zejména funkce pro postupný převod z regulárního výrazu předmětu IFJ do regulárního výrazu jazyka Python.

  Dále je zde funkce `formating_text`, která volá převedené regulární výrazy nad vstupním souborem. Z výsledků se vytváří seznam jednotlivých pozic v souboru a formátu, který se má na tyto bloky aplikovat. To je nutné provést až nakonec, jinak by mohlo dojít k formátování již vložených HTML tagů. Skript to řeší vytvořením seznamu ve tvaru [*tag před znakem, znak, tag za znakem*], díky tomu nedochází ke změně indexů v důsledku postupného vkládání. Nakonec je z tohoto seznamu opět vytvořen string, který je uložen do výstupního souboru.

3. Rozšíření

Skript implementuje obě rozšíření.

- **HTM**

  Z tohoto rozšíření je implementována pouze druhá část a to `escape`. Escapování znaků je provedeno až na závěr, těsně před tím, než je výstup doplněn o tagy vytvořené skriptem, takže nehrozí, že by došlo k poškození skriptem vytvořených tagů.

- **NQS**

  Rozšíření je implementováno pomocí několika regulárních výrazů, které nahradí vícenásobné kvantifikátory za jeden.