

## 1 Zadání

Úkolem tohoto projektu bylo vytvořit skript v jazyce Python 3 pro automatické zvýrazňování různých částí textu. Skript pracuje s tabulkou regulárních výrazů, ke kterým je přiřazeno požadované výstupní formátování.

## 2 Popis řešení úlohy

Skript je rozdělen na jednodušší programové úseky, které dokáží fungovat samostatně, ale ve výsledku aby byla dodržena funkcionálnost skriptu navzájem spolupracují.

### 2.1 Zpracování argumentů

Argumenty příkazové řádky jsou zpracovány pomocí modulu `argparse`. Skript zanalyzuje argumenty příkazové řádky hned po spuštění a kontroluje, zda nejsou zadány argumenty duplicitně, nebo zda nedošlo ke kolizi argumentů.

### 2.2 Vstup skriptu

Pokud je zadán přepínač `--input`, tak skript zpracovává soubor daný tímto přepínačem, jinak skript zpracovává text zadáný na standardní vstup. V případě chyby otevření datového proudu, nebo souboru, skript vyvolá příslušnou chybovou hlášku a ukončí se příslušným chybovým kódem.

### 2.3 Formátovací soubor

Vstup skriptu je modifikovaný na základě formátovacího souboru daného přepínačem `--format`. V případě, že přepínač `--format` chybí, nebo soubor daný tímto přepínačem chybí popřípadě z nějakého důvodu nejde otevřít, tak skript pouze kopíruje neformátovaný vstup na výstup. Formátovací soubor obsahuje na každém řádku regulární výraz, podle kterého se aplikují tagy, které jsou od regulárního výrazu odděleny znakem `\t`.

### 2.4 Analýza formátovacího souboru

Formátovací soubor se analyzuje v souboru `parser.py`. Skript prochází formátovací soubor po řádcích a vždy text před prvním tabulátorem je identifikován jako regulární výraz a další věci za tímto regulárním výrazem jsou identifikovány jako tagy. Jednotlivé tagy musí být odděleny znakem `,` a libovolným počtem mezer nebo tabulátorů. Pro každý tag je ověřena jeho validita pomocí funkce `validnyTag()` a pro každý regex je ověřena jeho validita a je uskutečněn převod tohoto regexu na regex, kterému rozumí python ve funkci `validnyRegex()`. Následně se každý tag uloží se svým regulárním výrazem do pomocného slovníku. V případě nevalidity formátovacího souboru je skript ukončen.

### 2.5 Aplikace formátovacího souboru

Aplikace formátovacího souboru je zpracována v souboru `styler.py`, kde se dále používá pomocný slovník, získaný analýzou formátovacího souboru. Pro každou položku v pomocném slovníku jsou pomocí regulárního výrazu dané položky vypočteny umístění otevíracího a ukončovacího tagu a tyto informace jsou uloženy do dalšího pomocného slovníku spolu s příslušným tagem. Pokud jsou vypočítány umístění pro všechny tagy, tak se tagy nakonec aplikují na dané vypočítané místo.

## **2.7 Výstup skriptu**

Výstup skriptu je vytištěn na standardní výstup, nebo v případě zadaného přepínače --output do souboru specifikovaného tímto přepínačem.

## **3 Závěr**

Skript byl řádně otestován na systému Linux Deepin 2014.2 a na školním systému CentOS. Pro testování byla použita oficiální testovací sada a mnou vytvořená testovací sada. Všechny testy prošly úspěšně. Skript byl tvořen ve vývojovém prostředí PyCharm od firmy JetBrains.