

## **Dokumentace k projektu do předmětu IPP, zadání CSV**

### **Zadání**

Mým úkolem bylo vytvořit skript v jazyce Python, který vstupní CSV soubor převede do formátu XML.

### **Zpracování parametrů**

První věc, kterou skript provede je kontrola zadaných parametrů. K tomuto účelu jsem použil knihovní funkci `argparse`. V případě vícenásobného zadání stejného parametru, zadání neznámého parametru nebo nedovolené kombinace parametrů, ukončím skript s příslušným chybovým kódem. Pokud jsou zadané parametry validní, nastavím příznaky, podle kterých provedu požadovanou operaci.

### **Zpracování vstupního souboru**

Pokud nebyl zadán žádný vstupní soubor jako parametr, čtu data ze standardního vstupu. Celý vstup načtu do paměti, protože potřebuji soubor projít vícekrát kvůli rozšíření `PAD` a v případě souboru na standardním vstupu nemohu číst vícekrát.

Ke zpracování vstupního CSV souboru využívám knihovní funkci `CSV : :Reader`, která mi jednotlivé záznamy v CSV souboru uloží do dvourozměrného pole.

### **Generování XML souboru**

Samotné generování XML souboru probíhá ve dvou cyklech, protože vstup je uložen v dvourozměrném poli. Pokud není aktivní přepínač `-n`, tak jako první se vygeneruje hlavička. V případě aktivního přepínače `-h`, musím první řádek souboru projít cyklem a jednotlivé prvky uložit do pole, protože budu podle nich generovat názvy sloupců. Pokud je navíc aktivní přepínač `--all-columns`, názvy přebývajících sloupců jsou odvozeny podle hodnoty parametru `-l`. Dále u přepínačů `-r`, `-l` a `-c` musím kontrolovat, zda zadaný název elementu je validní.

Dalším problémem bylo vyřešit zotavení z chybného počtu sloupců na neprvním řádku. V tomto případě je důležité, zda je aktivní přepínač `--all-columns`. Pokud ano, tak všechny přebývající řádky jsou také vytištěny. Pokud není, tak pokud pomocný counter řádku je roven nebo větší než počet sloupců na prvním řádku, smyčka je přerušena a tisk dalšího sloupce neproběhne. Pokud je naopak počet sloupců menší než počet sloupců na prvním řádku, chybějící buňka je doplněna prázdným polem. Pokud je aktivní přepínač `--missing-field`, buňka je vyplněna hodnotou přepínače.

Před tiskem každé buňky je zavolána funkce, která konvertuje problematické znaky s UTF kódem menším než 128 na odpovídající zápisy v XML pomocí metaznaku `&`.

### **Zpracování výstupního souboru**

Pokud nebyl zadán příslušný parametr s cestou výstupního souboru, výstup je přesměrován na standardní výstup. Pokud výstupní soubor neexistuje, tak je vytvořen. Pokud již existuje, tak je bez varování přepsán. Výstupem je odpovídající XML soubor.

## Rozšíření

Rozhodl jsem se implementovat rozšíření PAD.

Rozšíření spočívá v doplnění u použitých čítačů o takový počet nul zleva, aby všechna čísla dané sekvence měla stejný počet číslic. Implementace byla přímočará. Prošel jsem vstupní soubor a spočítal počet řádků a sloupců. Poté jsem ve smyčce doplnil odpovídající počet nul podle délky čísla. Jediná komplikace nastala při aktivním přepínači `--all columns`, kdy jsem musel pro každý řádek zvlášť vypočítat počet sloupců.