Dokumentace úlohy CSV: CSV2XML v Python 3 do IPP 2016/2017
Jméno a příjmení: Martin Bobčík
Login: xbobci00

Úkolem tohoto projektu bylo napsat skript, který načte soubor ve formátu csv, a podle zadaných pravidel jej konvertuje na soubor formátu xml.

Parametry

Pro načítání přepínačů a jejich parametrů byla využita knihovna argparse. Většina přepínačů byla implementována bez problémů. Problém nastal pouze u přepínače -h, který kolidoval s přepínačem pro nápovědu automaticky generovanou touto knihovnou. Proto byla automatická nápověda zakázána, a vyřešena explicitním přidáním vlastního přepínače a nápovědy. Po definování všech přepínačů se zavoláním metody `parse_args()` vytvoří objekt s atributy, které obsahují hodnoty parametrů. Knihovna argparse poskytuje ošetření jednoduchých chyb v zadání parametrů (jako například neznámé přepínače, přepínače bez parametrů apod.). Ošetření dalších chyb (např. přepínač nápovědy s dalšími přepínači, přepínače, které jsou bez jiných bezvýznamné apod.) muselo být implementováno. Kvůli jednoduššímu hledání těchto chyb byly výchozí hodnoty přepínačů nastaveny na `False`, proto po ošetření chyb byly hodnoty nezadaných přepínačů nahrazeny jejich, podle zadání, implicitními hodnotami. Nakonec byly ošetřeny potencionální chyby s kódem 30, tedy zadání nevalidní xml značky. Toho bylo docíleno porovnáním řetězce s regulárním výrazem. Pokud byl zadán přepínač -h, nahradí se v xml nevalidní znaky v prvním záznamu csv souboru znakem zadaným jako parametr přepínače. Problém zde dělal znak „\n“, který metoda `sub()` knihovny re pro regulární výrazy brala jako jeden znak, ale podle zadání (a testů) měl být brán jako CR LF, tedy dva znaky. Tohle bylo řešeno nahrazení znaku „\n“ dvěma znaky pro substituci. Po nahrazení těchto znaků byl opět proveden test na validitu xml značek. Pokud test selhal, končilo se tentokrát s chybou 31.

Vstup

Skript dokáže zpracovávat jak standartní vstup, tak soubor zadán pomocí přepínače --input s parametrem cesty k souboru. Nejprve bylo zpracování vstupu řešeno načtením do proměnné, a rozděleno pomocí regulárních výrazů na řádky, a poté sloupce. Tento postup ale neřešil buňky csv souboru, které mohou být díky uvozovkám na více řádků. Toto řešení je pořád patrné ve zdrojovém kódu, kde jsou zanechány jeho metody.
Nakonec bylo tedy zvoleno zpracování pomocí knihovny _csv, která už nenačítá soubor do zvláční proměnné, ale její metodě `reader()` se předá ukazatel na soubor a separátor csv buněk. Tato metoda vrací objekt, kterým jde iterovat csv soubor po řádcích. Tyto řádky jsou uloženy do seznamu, který je dále zpracováván. Pokud je zadán prázdný vstup, je vygenerován prázdný xml soubor, a to buď s hlavičkou, nebo bez ní, podle přepínače -n. Pokud nebyl zadán přepínač pro zotavení z chyb, zkontroluje se velikost všech csv záznamů vůči prvnímu záznamu, a při nestejnosti skript končí s chybou 32.

Sestavení výstupu

Pro sestavení xml stromu výstupu byl použit ElemenTree z knihovny xml.etree . Sestavování stromu je rozděleno podle toho, zda má probíhat se zotavením z chyb nebo bez. Oba přístupy jsou vesměs stejné, jen u zotavení z chyb se mění maximální hodnota počítadla iterátoru v závislosti na velikosti csv záznamu a zadaných parametrů.

Kořenový element

Před iterací se musí vytvořit kořenový element xml stromu, bez kterého by knihovna xml.etree nepracovala správně. Jméno tohoto elementu je odvozeno z parametru přepínače -r. Pokud tento přepínač zadán není, je element po sestavení stromu smazán smazáním příslušných řádků výstupního řetězce.

Záznamy

Iteruje se dvourozměrným seznamem buněk po řádcích. Na začátku každé iterace se sestaví, na základě přepínačů, jméno elementu pro každý záznam. Jméno je zadáno parametrem přepínače -l. Pokud není zadán, je implicitně „row“. Pokud byl zadán přepínač -i, má tento element i atribut „index“ s pořadovým číslem záznamu. Začátek počítadla lze nastavit parametrem přepínače --start. Zde se také ošetřuje parametr --padding (viz níže).

Buňky

Po vytvoření elementu se iteruje skrz záznam po buňkách. Pro xml element každé buňky je opět sestaveno jeho jméno, které tentokrát sestává z prefixu nastaveného parametrem přepínače -c (implicitně „col“), a sufixu, který představuje pořadové číslo buňky. Zde se také aplikuje přepínač --padding. Pokud je zadán přepínač -h, je jméno

odvozeno z prvního záznamu csv souboru. Tento záznam složí jako hlavička, tedy jedna buňka odpovídá jednomu jménu elementu buňky.
Nakonec je přidána samotná buňka jako text elementu, a element je přiřazen do stromu.

Výstup

Po sestavení výstupního xml stromu by se již dalo tisknout, je ale třeba ještě ošetřit některé přepínače, a zformátovat vstup.
V následujících částech byl problém s výstupem některých metod, které ze záhadných důvodů někdy vracely řetězec ve formě pole bytů, a někdy jako pole znaků, proto je každá uzavřena v try-exception bloku, kdy v try je volání metody s dekódováním pole bytů do pole znaků, a v exception je volání bez této konverze. Tato metoda řešení ale nejspíš není úplně zdařilá, a šlo by najít elegantnější řešení.

Formátování výstupu

Jelikož knihovna xml.etree převádí xml strom na řetězec bez jakéhokoliv formátování (tj. celý soubor by byl na jednom řádku), je třeba ho třeba zformátovat. To se provádí metodou `prettify()`, která jako vstup bere strukturu xml stromu knihovny xml.etree. Strom je převeden na řetězec pomocí metody `tostring()` knihovny xml.etree. Tato metoda také řeší převod znaků nepřípustných v xml textu na jejich xml validní ekvivalenty. Řetězec je poté rozparsován do xml stromu knihovny xml.dom a pomocí metody `toprettyxml()` převeden na zformátovaný řetězec.

Přepínač -r

Přepínač -r říká, jestli má výsledný soubor obsahovat xml root element. Ten byl přidán při tvoření xml stromu. Root element se nachází na druhém a posledním řádku výstupního řetězce. Ten se proto převede po řádcích na seznam, a pomocí klíčového slova `del` je smazán jeho druhý a poslední záznam. Zde bylo využito možnosti jazyka Python iterovat seznamem i do záporných hodnot. Seznam je poté opět převeden na řetězec.

Přepínač -n

Přepínač -n říká, zda se na výstupu bude nacházet xml hlavička. Postup mazání hlavičky je stejný jako u přepínače -r, s tím rozdílem, že se smaže pouze první záznam seznamu.

Rozšíření Padding

O rozšíření padding se stará stejnojmenná metoda, která bere jako parametry maximální velikost iterátoru a jeho aktuální hodnotu a vrací řetězec reprezentující číslo doplněný o požadovaný počet formátovacích nul.
Metoda porovná délku obou čísel v textové podobě. Pokud se délky rovnají, vrátí rovnou výsledný řetězec. Jinak se textová podoba aktuální hodnoty vloží do pomocné proměnné a iteruje se pomocí cyklu `while` dokud nejsou délky stejné a v každé iteraci se pomocná proměnná konkatenuje se znakem nuly.