2023

Csv-json-converter

ZÁVĚREČNÝ PROJEKT DO PŘEDMĚTU 7PRCC

CARBOL ZDENĚK, R22430

Analýza problému

Pro vytvoření aplikace v C++ jsem si zvolil problém převodu CSV souborů do JSON souborů a naopak.

Vstupy aplikace

Aby taková aplikace mohla fungovat, tak musí nějakým způsobem přijímat vstupy od uživatele a vracet výstupy.

Když se zamyslíme nad vstupy aplikace, tak by se mělo jednat o:

- **cestu ke vstupnímu souboru,**
- **cestu k výstupnímu souboru,**
- **zadání způsobu převodu.**

Vzhledem k počtu vstupů zde bude vhodné zadávat tyto parametry jako vstupní parametry aplikace, takže se budou zadávat v příkazovém řádku.

Validace

Pro správný převod je potřeba, aby uživatel zadal korektní vstupy, tudíž bude potřeba před samotným převodem tyto vstupy validovat.

Je potřeba validovat:

- **způsob převodu** -> dejme tomu, že bude přípustné zadat **csv-to-json** nebo **json-to-csv**,
- **vstupní a výstupní soubor** -> validní budou zadané soubory se správnými koncovkami pro daný převod, pro jednoduchost půjde soubory zadat s libovolnou cestou najednou v jednom parametru,
- **počet zadaných parametrů**.

Výstup aplikace

V případě, kdy uživatel zadá validní parametry, nebude problém s otevíráním souborů ani s převodem pak bude výstupem aplikace uložení výstupního souboru (tak jak to uživatel zadal).

V opačném případě aplikace selže a vrátí ideálně chybovou hlášku s tím, co se nepovedlo.

Podoba souborů

V případě převodu souboru CSV do JSON budeme brát v potaz různé druhy oddělovačů pro data:

- čárku, středník a tabulátor.

Vstupní CSV soubor musí obsahovat hlavičku pro korektní převod. Pokud stringové hodnoty budou obsahovat znak oddělovače, tak by měly být v uvozovkách, ale v ideálním případě by všechny hodnoty měly být v uvozovkách.

V případě převodu JSON do CSV bude možné pro zjednodušení zpracovat pouze JSON soubory s nevnořenými bloky, ale JSON bez vnoření bloků s jednoprvkovými/více prvkovými hodnotami zpracovat půjde.

Návrh aplikace

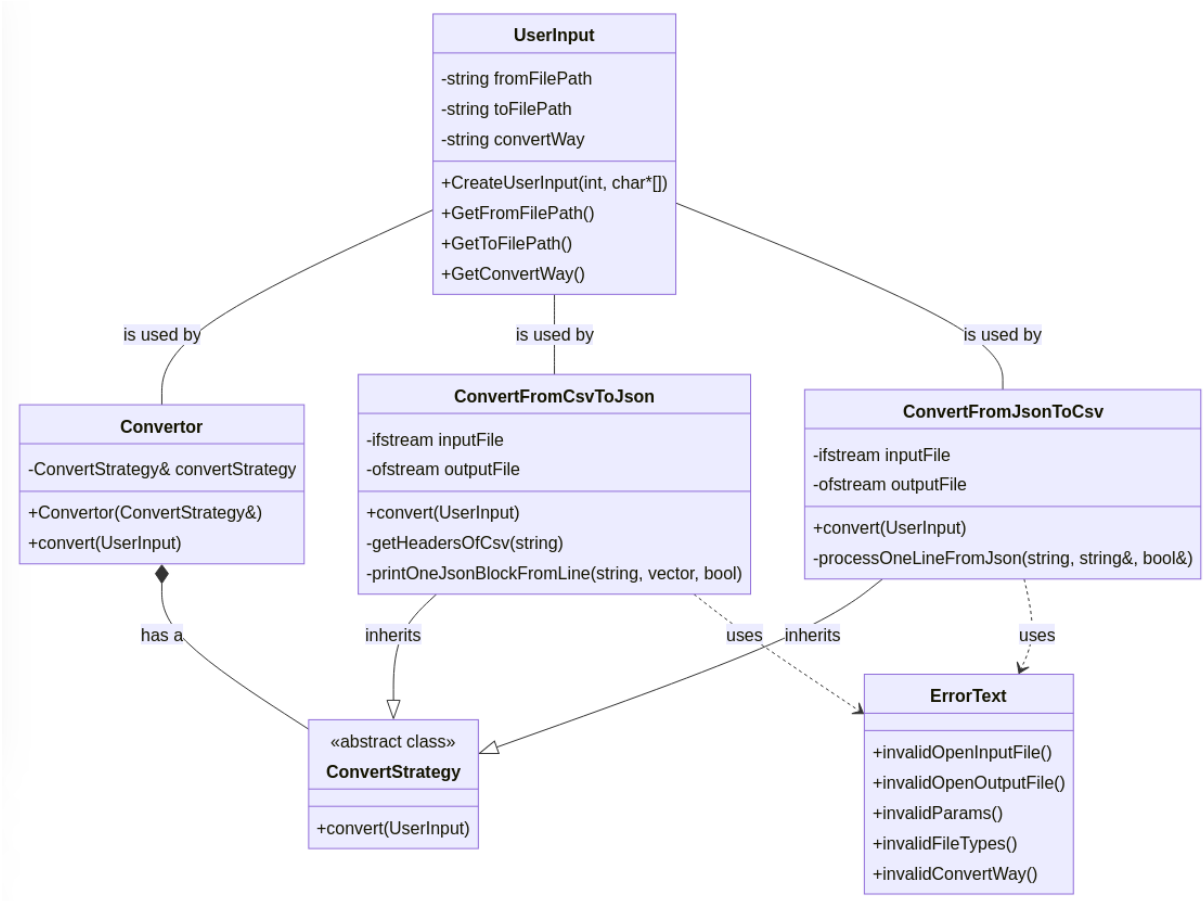
Ve funkci **main** bude minimum funkcionalit.

Pro validaci vstupů vytvořím třídu **UserInput**, která zároveň bude držet vstupy uživatele, takže to bude v podstatě taková přepravka.

Vzhledem k tomu, že mám dva způsoby převodů souborů a do budoucna by se jich mohlo objevit více, tak u převodu použiji **návrhový vzor Strategie**. Takže si vytvořím **abstraktní třídu ConvertStrategy s metodou convert**, která bude přijímat **UserInput** v parametru metody. Třídy jednotlivých strategií budou **ConvertFromCsvToJson** a **ConvertFromJsonToCsv**. Kontextem pro tyto strategie bude třída **Convertor**, u jejíž objektů bude možné v konstruktoru volit jednotlivé strategie.

Na závěr ještě implementuji třídu **ErrorText**, která na jednom místě bude držet všechny chybové textové výstupy, tudíž je bude snadno možné měnit na jednom místě v metodách, případně přidávat nové. Metody této třídy budou statické, protože není důvod vytvářet objekty.

Při hlubším rozboru tříd a zamyšlení budou třídy aplikace vypadat takhle:



Použité technologie

- C++,
- CMake,
- GitHub (zde bude i README.md s návodem na používání aplikace).

Závěr

Aplikaci se mi povedlo vytvořit a nahrát na tento můj veřejný repositář na GitHubu: https://github.com/Sidney711/csv-json-converter.