

## Dokumentace k projektu do předmětu IPP, XML2DDL v Pythonu 3

### Analýza problému

Mým úkolem bylo vytvořit skript v jazyce Python 3, který bude provádět analýzu vstupního XML souboru obsahující informace o tabulkách a následně generování sady SQL příkazů, které tyto tabulky v databázi vytvoří. Tvar výsledného souboru obsahující tuto sadu příkazů lze parametrizovat předáním parametrů při spuštění tohoto skriptu.

### Postup řešení

Prvním krokem při spuštění skriptu je kontrola a zpracování parametrů, jež byly skriptu předány z příkazové řádky. V dalším kroku se skript pokusí otevřít vstupní soubor obsahující XML definice tabulek a při úspěšném otevření načte jeho obsah do paměti. Po načtení vstupních dat, jež skript parsuje pomocí modulu `xml.etree.ElementTree`, se tato data rekurzivně prochází a analyzuje. Takto analyzovaná data se zapíše do výstupního souboru ve formátu dle předaných parametrů při spuštění skriptu. V případě úspěšné konverze se skript ukončí a vrátí nulovou návratovou hodnotu a v případě chyby se na standardní chybový výstup vypíše chybové hlášení a skript vrátí odpovídající chybovou návratovou hodnotu dle zadání.

### Zpracování parametrů

Pro zpracování parametrů se využívá modul `argparse`, který obstarává jak samotné zpracování zadaných parametrů při spuštění skriptu tak dle definovaných pravidel i následné generování textu nápovědy (parametr `--help`). Tento modul dle zadaných parametrů naplní vnitřní proměnné a na základě hodnot těchto proměnných se řídí další činnost skriptu. Pokud byl zadán neplatný parametr nebo neplatná kombinace parametrů program se odpovídajícím způsobem ukončí.

### Zpracování vstupního (XML) souboru

Pro zpracování vstupních dat ve formátu XML využívám modul `xml.etree.ElementTree`, pomocí kterého načítám a zpracovávám vstupní data z XML souboru a který také zprostředkovává rozhraní pro práci s takto načtenými daty. Takto načtená data dále rekurzivně procházím a analyzuji pomocí funkce `processXML`, která má za úkol naplnit vnitřní datovou strukturu typu `hash` (slovník) pro zjednodušení další práci s daty.

### Generování výsledné sady SQL příkazů

Pro vytvoření výsledné sady SQL příkazů je implementována funkce `printDDL`, jež vygeneruje sadu SQL příkazů z dříve naplněné vnitřní datové struktury `hash` data a zapíše ji do výstupního souboru. Pro generování XML dokumentu obsahujícího informace o relacích mezi tabulkami (přepínač `-g`) se využívá funkce `printXML`, která analyzuje obsah vnitřní datové struktury `hash` `relations` a následně vygenerované vazby pomocí modulu `xml.dom.minidom` zapisuje do výstupního souboru.

### Zpracování chyb

Pro zpracování chyb v implementovaném skriptu využívám funkce `printError`, které předávám jako první parametr chybové hlášení a jako druhý parametr požadovaný návratový kód při ukončení skriptu.

### Závěr

Při testování výsledného skriptu jsem využil všech veřejně přístupných testů (včetně těch, které byly přiloženy k zadání) a všemi těmito testy skript prošel. Testování jsem prováděl na operačním systému GNU/Linux Mint a také na referenčním školním serveru Merlin.