

## ÚVOD

Cílem projektu bylo vytvořit skript v jazyce Python (`parse.py`), který má za úkol ze vstupního programu na standardním vstupu, který obsahuje program s instrukcemi jazyka IPPcode24, lexikálně a syntakticky zanalyzovat a vygenerovat .XML reprezentaci na standardní výstup.

## VYUŽITÉ KNIHOVNY

V projektu byla využita knihovna **xml.etree.ElementTree**, která usnadňuje práci s .XML formátem, jelikož umožňuje ukládat stromovou reprezentaci .XML souboru a vygenerování výsledku až ke konci běhu skriptu.

## NÁVRH

Projekt je rozčleněn do tří částí:

1. **InstructionProcessing.py** – obsahuje pseudo-třídu (zde pouze pseudo-třída pro lepší přehlednost a orientaci v kódu) **Instruction** věnující se lexikální analýze IPPcode24 kódu. **Instruction** obsahuje metody pro kontrolu správného lexikálního zápisu proměnné (*CheckVariable*), konstanty (*CheckConstant*), návěští (*CheckLabel*), typu konstanty (*CheckType*) dále obsahuje pomocnou metodu pro kontrolu řetězce (*CheckString*) – zda obsahuje pouze povolené znaky a metodu pro odstranění komentáře z řádku (*RemoveComments*).
2. **XML\_Processing.py** – obsahuje třídu **XML**, která nese jako atribut stromovou reprezentaci XML a čítač instrukcí, inkrementující se po každém přidání XML uzlu do stromové struktury. Třída se stará o převod IPPcode24 na .XML soubor. Třída **XML** obsahuje metody vykonávající převod: *PrintArithmetic* – převádějící aritmetické instrukce, *PrintRead* - pro instrukci READ, *PrintWrite* – instrukci WRITE, *PrintJump* – převádějící skokové instrukce, *PrintDeclaration* – instrukci DEFVAR, pro všechny ostatní instrukce je použita metoda *PrintInstruction* (kde není třeba specificky kontrolovat „parametry“ IPPcode24) a dále třída obsahuje pomocné metody: *Return Type*, vracející typ proměnné/konstanty (var, bool, nil, string, int) a metodu *Return Value*, vracející hodnotu proměnné/konstanty.
3. **Parse.py** – obsahuje výsledný spustitelný skript, který dostává na standardní vstup text s kódem v IPPcode24 anebo je volán s argumentem „—help“, v tomto případě skript vysází nápovědu na standardní výstup. Vstupní kód musí obsahovat hlavičku .IPPcode24 a instrukce IPPcode24, dále může volitelně obsahovat komentáře začínající znakem „#“, zpracovávání zdrojového kódu probíhá na bázi konečného automatu, kdy po kontrole validních argumentů skriptu, následuje hledání hlavičky .IPPcode24 a poté jsou postupně zpracovávány řádky zdrojového kódu za pomoci tříd z **InstructionProcessing.py** a **XML\_Processing.py**. Po zpracování validního zdrojového kódu dochází ke generování .XML do výstupního souboru a skript úspěšně končí. V případě, že během zpracování kódu dojde k chybě, program končí hodnotami, popsány v zadání.