

Implementační dokumentace k první úloze do IPP 2023/2024

[Andrii Bondarenko]

Login: [xbonda06]

9. března 2024

Úvod

Tento dokument popisuje implementaci skriptu `parse.py`, který jsem vytvořil v rámci první úlohy předmětu IPP na VUT FIT v akademickém roce 2023/2024. Tento skript zajišťuje lexikální a sémantickou analýzu jazyka IPPcode24 a generaci XML souboru, který reprezentuje tento kód.

Popis skriptu `parse.py`

Skript `parse.py` ze standardního vstupu postupně po řádcích čte zdrojový kód v jazyce IPPcode24. Dostává instrukce, které analyzuje a přidává do XML souboru.

Zdrojový kód je analyzován ve smyčce, která je součástí hlavní třídy `Parser`. Smyčka odstraňuje komentáře, mezery a bílé znaky, které ignoruje. Každou instrukci tato třída očísluje pomocí atributu `order`, uchovávající informaci o tom, jaký řádek právě analyzuje.

Na začátku smyčka zkontroluje, jestli je první řádek `".IPPcode24"`. Pokud ano, proběhne inicializace XML souboru. Každá následující instrukce je postupně zpracována, ověřena její existence, typ, správnost argumentů a správně přidána do XML souboru. Pro práci s XML souborem jsem použil moduly `The ElementTree XML API` a `Minimal DOM`.

Pro zjednodušení analýzy jsou vytvořeny seznamy instrukcí, které IPPcode24 podporuje, a speciální seznamy různých typů instrukcí podle počtu a typu argumentů. Vytvořil jsem také seznam povolených typů argumentů a paměťových rámců. Seznamy jsou využívány jednotlivými třídami pro kontrolu správného počtu a typů argumentů.

Skript je vytvořen podle principů OOP a obsahuje několik tříd, které mají definované role a komunikují mezi sebou. Celá funkcionalita programu je zajištěna metodami těchto tříd:

- `ProgramIOController`: Zpracovává vstupní parametry skriptu a poskytuje nápovědu nebo vypisuje chybové zprávy.
- `Instruction`: Reprezentuje jednu instrukci IPPcode24, včetně jejího pořadí, operačního kódu a argumentů.
- `Argument`: Uchovává informace o jednom argumentu instrukce, včetně jeho typu a hodnoty.
- `InstructionValidator`: Provádí syntaktickou a lexikální analýzu instrukcí a ověřuje správnost argumentů.
- `Parser`: Zajišťuje hlavní logiku zpracování vstupního kódu a jeho převodu do XML.
- `XMLGenerator`: Třída pro generování výstupního XML dokumentu.