

IMPLEMENTAČNÁ DOKUMENTÁCIA KU 1. PROJEKTU DO IPP (parse.py) 2023/2024

Meno a Priezvisko: **Boris Vícena**

Login: **xvicen10**

Úvod

Táto dokumentácia obsahuje stručný popis ku implementácii skriptu `parse.py`, ktorý slúži ako prekladač pre jednoduchý imperatívny jazyk IPPcode24. Skript spracováva zdrojový kód IPPcode24, vykonáva lexikálnu a syntaktickú analýzu a následne generuje formátovaný XML výstup reprezentujúci štruktúru analyzovaného programu.

Návrh a interná reprezentácia

Cieľom bolo vytvoriť skript (*kód*), ktorý bude schopný úspešne analyzovať a spracovávať kód jazyka IPPcode24. Kód v skripte je organizovaný do tried pre tokeny `class Token`, lexikálnu analýzu `class LexerFSM`, syntaktickú analýzu `class Parser`, a generovanie XML výstupu `class XMLGenerator`. Interná reprezentácia programu je udržiavaná pomocou tried a dátových štruktúr. Lexikálna analýza rozdeľuje kód jazyka IPPcode24 na základe vstupných dát a produkuje zoznam tokenov, ktoré sú následne spracované syntaktickou analýzou. Po úspešnej lexikálnej a syntaktickej analýze je program reprezentovaný štruktúrou dát vo forme slovníka `program`. Tento slovník obsahuje zoznam inštrukcií s príslušnými argumentami, pripravenými na konverziu do XML formátu.

Spôsob riešenia

Lexikálna analýza

Lexikálna analýza je vykonávaná pomocou triedy `LexerFSM`. Kód je rozdelený na tokeny pomocou funkcie `tokenize()`, ktorá odstráni z kódu komentáre a následne volá funkciu `tokenize_line()` na spracovanie každého tokenu pre jednotlivé riadky, kde každý token je klasifikovaný podľa typu za pomoci funkcie `classify_token()`. Klasifikácia je založená na pravidlách jazyka IPPcode24 a využíva pravidelné výrazy (*regular expressions*).

Syntaktická analýza

Syntaktická analýza prebieha v triede `Parser`. Táto časť kódu overuje prítomnosť a správnosť hlavičky programu, prechádza jednotlivé tokeny pomocou funkcie `parse()`, ktorá využíva funkciu `parse_instruction()` pre spracovanie inštrukcií a vytvára štruktúru programu na základe pravidiel jazyka IPPcode24. Kontrola a verifikácia typov a poradia argumentov inštrukcií prebieha na základe slovníka `INSTRUCTION_ARGUMENTS_TYPES`.

Generovanie XML výstupu

Na generovanie XML výstupu slúži trieda `XMLGenerator`. Tá prechádza štruktúrou programu a vytvára XML elementy pre každú inštrukciu s jej argumentami pomocou funkcie `generate_xml()`. Na formátovanie XML výstupu sa využíva knižnica `xml.minidom` vo funkcií `format_xml()`, ktorá má za úlohu daný XML výstup sprehládniť.

Chybové stavy

Pomocou konštánt `ERR_INVALID_HEADER`, `ERR_INVALID_OPCODE`, `ERR_LEXICAL_SYNTAX`, `ERR_ARGS` a `NO_ERR` s hodnotami 21, 22, 23, 10 a 0 v danom poradí je skript schopný detekovať a spracovať rôzne chybové stavy, ako sú chýbajúca alebo neplatná hlavička, neznáme inštrukcie alebo lexikálno-syntaktické chyby. Chybové kódy sú poskytované na identifikáciu problému a vhodné chybové správy sú zobrazené s príslušným chybovým kódom (*konštantou*).