Implementační dokumentace k 1. úloze do IPP 2023/2024

Jméno a příjmení: Radek Libra

Login: xlibra02

Celková filozofie návrhu

Při řešení této úlohy byl především kladen důraz na modularitu, implementováním nezávislých tříd/komponentů, které byly vytvořeny tak, aby se dali používat i nezávisle v jiný projektech nebo kontextu. Toto je umožněno díky tomu, že každý tento komponent má jasně danou funkci a je oddělen od zbytku skriptu. Další klíčovou vlastností, která byla značnou podmínkou při řešení tohoto scriptu, je rozšiřitelnost. Kdy například implementace validace instrukce je navržena tak, aby přidání nové instrukce vyžadovala minimální úsilí. Ve většině případů pouze přidání vzoru dané instrukce do již existujících struktur. Implementace je také zaměřena na celkovou čistotu a srozumitelnost kódu. Všechny komponenty a funkce mají, jak již bylo zmíněno, jasně daný účel, který lze snadno odvodit od názvu těchto jednotlivých částí. Například třída `Instructions` se zaměřuje na výhradně na reprezentaci a validaci instrukcí IPPcode24, zatímco třída `Statistics` spravuje logiku zajišťující zpracování statistik. Identifikátory jsou pojmenovány tak, aby co nejpřesněji odráželi jejich účel nebo obsah. V neposlední řadě byl kladen důraz na Udržitelnost a Testovatelnost. Tyto dvě vlastnosti jsou přínosné především při samotné implementaci a testování scriptu. Kdy například každá třída mohla být testována nezávisle.

Interní reprezentace

Třída `Instruction`

Reprezentuje jednotlivé instrukce IPPcode24. Jejímž atributy jsou `opcode` (řetězec reprezentující operační kód instrukce), `operands` (seznam operandů řetězce), `operands_types` (seznam typů operandů odvozených z jejich syntaktické analýzy). Tato třída dále umožňuje validaci operandů, na základě regulárních výrazů a kontrolu validity operačního kódu.

Třída `Statistics`

Uchovává informace o počtu jednotlivých statistik analýzy kódu IPPcode24. Mezi její atributy patři slovník `statistics`, který mapuje názvy statistik a jejich počet v průběhu analýzy a slovník `opcode_counts`, který mapuje jednotlivé instrukce a jejich počet v kódu. Tato třída poskytuje metody pro inkrementaci jednotlivých statistik, procházení všech instrukcí pro vyhodnocený statistik skoků na základě kontextu použití a vypsání skupin statistik do předem daných souborů.

Třída `XMLGenerator`

Generuje XML reprezentaci zpracovaného kódu na základě seznamu instrukcí. Hlavní metoda tohoto objektu je metoda `xml_generate`, která zajišťuje generování výsledného výstupu. Dále je zde implementována metoda `xml_print`, která zajišťuje výpis výsledku na standardní výstup

Hlavní logika je pak implementována v hlavním souboru parse.py, který využívá knihovnu `argparse` pro zpracování a validaci vstupních argumentů. Tato logika dále obsahuje smyčku pro čtení a analýzu vstupního kódu, používá `Instruction` pro reprezentaci a validaci instrukcí a `XMLGenerator` pro vytvoření výstupu. Současně sbírá statistiky o zpracovaném kódu pomocí `Statistics`.

Postup řešení

Skript parse.py implementuje proces analýzy zdrojového kódu IPPcode24 od zpracování vstupních argumentů, přes ověření hlavičky a lexikální a syntaktickou analýzu instrukcí, až po generování XML reprezentace a sběr statistik o zpracovaném kódu pomocí třídy `Statistics`. Analýza se soustředí na validaci instrukcí a operandů podle specifikace IPPcode24 s použitím tříd `Instruction`. Po validaci je finální XML výstup, společně se statistikami, pokud byly požadovány, vypsán pomocí `XMLGenerator`. Celý proces končí úspěšně s návratovým kódem 0, nebo signalizuje chybu příslušným chybovým kódem. Jediný řešený problém, který nebyl přesně specifikován v zadání je problematika řešení validního operačního kódu instrukce, kdy bylo potřeba jasně určit, kdy se jedná o chybu operačního kódu a kdy o syntaktickou chybu. Toto bylo zajištěno porovnáním s regulárním výrazem vytvořeným specificky pro operační kód.