

Implementačná dokumentácia k projektu z IPP 2021/2022

Meno: Pavel Kratochvíl
Login: xkrato61

Zadanie:

Cieľom projektu bolo vytvorenie troch skriptov (parse.php, interpret.py, test.php). Tie dokopy interpretujú a testujú správnosť interpretácie neštruktúrovaného imperatívneho trojadresného jazyka IPPcode22.

Skript parse.php

Prvou časťou projektu je skript na syntaktickú a lexikálnu analýzu jazyka IPPcode22. Zdrojový kód je spracovávaný z užívateľského vstupu. V prípade korektnosti je z neho vygenerovaný XML dokument na výstupe. K zobrazeniu nápovedy dochádza pri spustení s flag-om `--help`. Pri použití iných flag-ov alebo parametrov je vrátený chybový návratový kód 10.

Popis implementácie:

Pri načítaní zdrojového kódu je najprv nájdená povinná hlavička .IPPcode22. Pri jej hľadaní musíme zohľadniť možnú prítomnosť prázdnych riadkov alebo komentárov. Na preskočenie prázdnych riadkov je v celom skripte využívaná kontrolná funkcia `lineIsEmpty()` a na odstránenie komentárov funkcia `lineStripComment()` a následne vstavaná funkcia `trim()` na odstránenie prebytočných medzier a koncov riadkov. Na samotné porovnanie správnosti hlavičky je využitý regex výraz s kontrolou pomocou vstavanej funkcie `preg_match()`.

Následuje sekvencia riadkov, z ktorých každý začína inštrukciou a jej parametrami (v prípade ak nejaké má). Na určenie použitej inštrukcie som použil implementáciu pomocou štruktúry `switch`, v ktorej sú inštrukcie zoskupené do skupín s rovnakou sekvenciou argumentov. Napríklad inštrukcie `LABEL`, `CALL` a `JUMP` očakávajú práve jeden argument – `<label>`. Vďaka tomu sa nám uľahčí ich kontrola, keďže pri každej skupine potom stačí zavolať funkciu `checkArgumentCount()` s počtom špecifickým pre danú skupinu. Inštrukcia je potom uložená do XML "stromu" funkciou `xmlWriteCommand()`. Tá im priradí aj poradové číslo podľa globálnej premennej. Argumenty sú vo výstupnom XML kóde "potomkami" danej inštrukcie, k ich zápisu dochádza v jednej z funkcií: `instructionParseVar()`, `instructionParseLabel()`, `instructionParseConst()`, `instructionParseSymb()`. V nich sa regex-om kontroluje sémantická a syntaktická správnosť a zapíšu sa do XML stromu k svojej inštrukcii.

Spracovávanie chyby počas behu programu:

Chcel som v projekte implementovať spôsob pre čo najlepší výpis chybovej hlášky pre užívateľa. Pri akejkolvek chybe je zavolaná funkcia `handleError()` s parametrom chybového kódu určeného miestom výskytu. Vďaka tomu je užívateľ informovaný chybovou hláškou na `STDERR`. Napríklad o chybnnej hlavičke alebo chybe v argumente konštanty. Návratový kód sa zo skriptu vracia podľa špecifikácie.