Implementační dokumentace k 2. úloze do IPP 2020/2021
Jméno a příjmení: Peter Zdravecký
Login: xzdrav00

1 interpret.py

Účel: Interpretácia kódu zapisaného v jazyku IPPcode21.

1.1 Základ

Spracovanie argumentov

K spracovaniu vstupných argumentov programu sa využíva knižnica `argparse`.

Inštrukcie

Základ programu tvorí premenná `instructions` typu `dictionary`, v ktorej sa uchovávajú informácie ohľadom existujúcich inštrukcií jazyka IPPcode21. Kľúčové hodnoty v tomto poli sú názvy inštrukcií. Uchovaná hodnota pri každom kľúči je pole argumentov, ktoré môže funkcia nadobúdať a zároveň ukazateľ na funckiu, ktorá bude volaná pri interpretácii kódu.

1.2 Načítanie a kontrola XML vstupu

Načítavanie a kontrola vstupu sa vykonáva vo funckii `checkXMLandSave`. Na načítanie `XML` dát sa využíva knižnica `xml.etree.ElementTree`. Funkcia vracia štruktúru v kotrej su uložené operačné kódy a argumenty inštrukcií.

Vizualizácia štruktúry uloženého programu IPPcode21

Tvar zápisu jednej inštrukcie:
`{0: {'instruction': 'DEFVAR','args': [{'type': 'var', 'value': 'GF@counter'}],'counter': 0,'order': '1'}}`

1.3 Interpretácia kódu

Rámce

Rámce ktoré uchovávajú premenné programu sú reprezentované ako slovníky, v ktorých sa dá vyhľadavať podľa názvu premmenej. Rámce (`TF|LF|GF`) su uložené v jednom poli pre jednoduchosť prístupu.

Pomocné funckie

Pre každú inštruckiu ktorá existuje v jayzku IPPcode21 je vytvorená pomocná funckia, ktorá sa zavolá pri jej interpretácii. Odkazy na tieto funckie su uložené v premmenej `instructions` a vstupy týchto funkcií sú argumenty pre jednotlivé inštrukcie.

Riadenie toku programu

Funckia `interpreteCode` vykonáva interpretácia kódu. Pre riadanie toku programu sa využíva while cyklus. Pomocou globálnej premmenej `currentInstIndex` sa určuje ktorá inštrukcia sa má vykonať. Interpretácia končí po vykonaní poslednej inštrukcie alebo pokiaľ sa nezavolá inštrukcia `EXIT`.

Skokové inštrukcie

Skokové inštrukcie menia globalnú premenú s názvom `currentInstIndex`, ktorá riadi tok programu. Nastavuje sa podľa návestia, na ktoré sa má skákať. Návestia sú uložené v slovníku. Ku každému návestiu je priradená hodnota, ktorá určuje kde sa návestie vo vykonávanom programe nachádza. Slovník návesti sa vytvára už počas spracovania `XML` vstupu.

1.4 Rozšírenia

Vypracované rožšírenia: `FLOAT | STACK | STATI`

2 test.php

Účel: Testovanie funkčnosti skriptov `parse.php` a `interpret.py`.
Výstup: Výsledky testov v podobe webovej stránky.

2.1 Základ

Spracovanie argumentov

K spracovaniu vstupných argumentov programu sa využíva knižnica `getopt`.

Dočasný súbor

Na začiatku programu sa vygeneruje dočasný súbor, do ktorého sa ukladajú výstupy spúštaných skriptov, ktoré slúžia na kontrolu.

2.2 Spúštanie testov

Testy sa vyhľadávaju v adresári podľa prípony `.src`. Súbory s príponami `.in | .out | .rc` v prípade, že neexistujú sa dogenerúju. Na každý test sa volá funkcia `doTest`. Návratová hodnota tejto funckie slúži na kontrolu, či test bol úspešný alebo nie. Výsledky testov sa zapísuju do poľa, ktoré neskôr služí na generovanie výstupu.

2.3 Funkcia `doTest`

Funckia na základe prepínačov ktoré, boli zadané na príkazovu riadku, pri spúštaní testovacieho skriptu vykonáva jednotlíve testy a porovnáva správnosť výstupov.

2.4 Generovanie webovej stránky

Na konci programu sa z poľa, kde sú zapísané výsledky jednotlivých testov, vygeneruje webová stránka. Na tejto stránke sa nachádza celkové vyhodnotenie testov a vyhodnotenie testov podľa jednotlivých priečinkov, v ktorých sa testy nachádzali.

2.5 Rozšírenia

Vypracované rožšírenia: `FILES`