

# IPP Projekt

## 2. část – skripty v jazyce PHP a python

### interpret.py

Skript interpret.py načítá XML reprezentaci zdrojového kódu v jazyce IPPcode20. Provádí následně jeho lexikální a syntaktickou analýzu a dále provádí interpretaci programu společně s běhovou sémantickou analýzou. Výstup programu tiskne na standardní výstup a ukončuje činnost a příslušným návratovým kódem.

#### 1 Přehled funkce skriptu

Skriptu je možno volitelnými parametry specifikovat soubory, ve kterých se nachází buďto interpretovaný program, či vstup do běhu programu. Pokud jeden z těchto parametrů chybí, bere se jako vstup implicitně standardní vstup. Alespoň jeden z těchto parametrů však musí být specifikován.

Skript zpracovává XML reprezentaci programu pomocí knihovny ElementTree, která disponuje spoustou funkcí pro zpracovávání XML kódu. Touto cestou si skript celý XML kód zpracuje do stromového objektu, který potom postupně kontroluje na správnost jeho struktury, názvů tagů, atributů, atd.. Dále provádí interpretaci programu postupně instrukci po instrukci. Při tom probíhá kontrola správnosti operandů všech instrukcí.

#### 2 Pomocné funkce

##### 2.1 zpracujInstrukce(root)

Tato funkce přijme argument „root“, což je kořenový element stromového objektu XML kódu. Následně kontroluje kořenový element na jeho syntax a ukládá všechny jeho subelementy do seznamu. Potom kontroluje správnost zápisu pořadí instrukcí přes argument „order“ a pomocí funkce najdiLabely koná první průběh všemi instrukcemi podle pořadí, při kterém si uloží do seznamu všechny návěští programu a jejich pozici v kódu, přičemž kontroluje i jejich duplicitu. Nakonec volá funkci checkSyntax a vykonejInstrukce.

##### 2.2 checkSyntax(instrList)

Tato funkce přijímá jako argument seznam všech elementů instrukcí programu. Poté vykonává průběh celým seznamem, při kterém kontroluje správnost zápisu operačního kódu instrukcí a počet jejich operandů ve formě elementů arg1, arg2,...

##### 2.3 vykonejInstrukce(instrList, orderList, slovníkLabelu)

Funkce vykonejInstrukce je pro chod interpretu nejtěžejnější. Na začátku si vytvoří všechny potřebné proměnné, čímž jsou slovníky pro uchovávání proměnných rámců, counter pro cyklus napříč instrukcemi, seznamy, které fungují jako zásobníky pro zásobníkové instrukce, zásobník volání a zásobník lokálních rámců, a potřebné flagy. Dále provádí cyklus všemi instrukcemi, které po jedné vykonává. Po instrukcích v seznamu předaným atributem „instrList“ se pohybuje pomocí seřazeného seznamu atributů „order“, které dostane pomocí atributu „orderlist“. Proměnná „counter“ slouží jako index v seznamu „order“ atributů. Díky tomu lze tuto proměnnou v cyklu modifikovat pro dosažení funkce skokových instrukcí.

##### 2.4 zpracujArgumenty()

Zpracovává argumenty předané skriptu. Modifikuje hlavně způsob vstupu dat do činnosti skriptu.

##### 2.5 precitiHodnotu(argument, slovníkTF, slovníkLF, slovníkGF, TframeCreated...

Tato funkce je často volaná funkcí vykonejInstrukce. Navrací hodnotu dat přečtených z argumentu instrukce.

## test.php

Skript test.php slouží pro automatické testování skriptů parse.php a interpret.py. Po zpracování všech argumentů skript provede všechny testy v zadané složce a na standardní výstup vytiskne HTML reprezentaci výsledků všech testů.

### 1 Přehled funkce skriptu

Na začátku skript začne prohledávat argumentem zadanou složku s testy. Za každý nalezený test provede porovnávání výstupu skriptů parse.php a interpret.py při zadaném vstupu se vzorovým výstupem. Všechny výsledky uchovává a následně tiskne na standardní výstup HTML reprezentaci výstupu, která obsahuje přehledné tabulky s podrobnými výsledky jednotlivých testů a celkovou úspěšnost testování.

### 2 Pomocné funkce

#### 2.1 scan(\$slozkaTestu)

Tato rekurzivní funkce prohledá obsah složky zadané argumentem. Pokud najde správně pojmenované soubory s testy, provede na každém test pomocí funkce „test()“. Pokud byl skriptu zadán parametr –recursive, tak stejným způsobem provede tato funkce prohledávání každé podsložky, kterou najde. Docíleno je tak rekurzivním voláním funkce. Informace o všech nalezených složkách se ukládají do velkého seznamu seznamů.

#### 2.2 test(\$nazev, \$slozkaTestu, \$IDSlozky)

Tato funkce provede test podle názvu souborů zadaného parametrem „\$nazev“. Prvně zkontroluje úplnost všech testových souborů, popřípadě dogeneruje soubory pro vstup a očekávaný výstup a návratový kód.

Následně v závislosti na zadání argumentů –parse-only a –int-only otestuje skripty parse.php a interpret.py nebo pouze jeden z nich. Testování provede spuštěním daných skriptů přes příkaz exec() a následným porovnáváním výstupů skriptů se vzorovým výstupem a návratovým kódem v souborech testu. Výsledky testu jsou poté uloženy do strukturovaného seznamu seznamů s výsledky, který je použit při generaci HTML výstupu testovacího skriptu.

#### 2.3 error(\$hodnota, \$zprava)

Tato funkce se volá při chybné práci s testovacím skriptem. Vypíše na standardní chybový výstup chybovou hlášku a ukončí skript s daným návratovým kódem.