

Implementačná dokumentácia k 1. úlohe do IPP 2020/2021

Meno a priezvisko: Martin Zaťovič

Login: xzatov00

Popis: Skript `parse.php` kontroluje lexikálnu a syntaktickú správnosť súboru načítaného zo štandardného vstupu podľa jazyka IPPcode21. Využíva knižnicu `DomDocument` na vytvorenie štruktúry a generáciu výstupného XML súboru. Skript najskôr skontroluje hlavičku vstupného súboru a v prípade, že je správna pokračuje načítaním ďalších riadkov. Na každom riadku vyhľadáva a odstraňuje komentáre označené znakom '#' a následne riadok rozdelí na podčasti obsahujúce operačný kód a argumenty. Vytvorí objekt inštrukcie a objekty argumentov podľa zadanej definície inštrukčnej sady. Podľa definovaného a teda očakávaného typu argumentu sa kontroluje lexikálna správnosť samotného argumentu. Na základe abstrakcie do objektov je možné jednoducho pristupovať k spracovaným informáciám o operačnom kóde a argumentoch funkcie, ktoré sú potrebné pre generovanie XML-dokumentu.

Parametre:

--help: Zobrazí krátku nápovedu k skriptu `parse.php`

Triedy:

Instruction: zaobaluje celú inštrukciu, volá jej spracovanie, priradzuje jej argumenty

Argument: abstraktná trieda

Triedy dediace of triedy Argument (tieto triedy obsahujú jedinú funkciu na skontrolovanie a uloženie hodnoty argumentu):

Nil - trieda pre špeciálny typ nil

Variable - trieda pre argumenty typu var

Symbol - trieda pre argumenty typu symbol (`nil@nil`, [LF/TF/GF@variable](#), `int@3...`)

Label - trieda pre argumenty typu label (`while`, `end`)

Type - trieda pre typy (`int`, `bool`, `string`)

Riadiace funkcie:

- trieda Instruction

`__construct` – z jedného riadku zo vstupného súboru vymaže komentáre a rozdelí ho na operačný kód a argumenty. Následne volá funkciu `opcode_set` pre uloženie operačného kódu do atribútu `opcode`, vďaka ktorej predpokladá aké argumenty by mali nasledovať. Ďalej volá funkciu `set_value` pre každý argument inštrukcie.

`opcode_set($opcode)` – volá funkciu `get_group` s parametrom `$opcode` pre zistenie čísla skupiny. Na základe skupiny vytvára objekty argumentov. Na záver aktualizuje atribút počtu argumentov objektu inštrukcie a taktiež jeho atribút operačného kódu.

`get_group($opcode)` – rozdeľuje operačné kódy do skupín podľa spoločných typov argumentov. Každá skupina má svoje číslo viz. príloha1. V tejto tabuľke operačných kódov je vyhladaný operačný kód z argumentu funkcie. Funkcia vracia -1 pri neznámom operačnom kóde, alebo číslo príslušnej skupiny.

- trieda Argument

`fetch_value($value)` – abstraktná funkcia, ktorá je definovaná každým typom argumentu zvlášť, čo umožňuje priradenie správneho typu a osobitú kontrolu lexikálnej a syntaktickej správnosti pre každý argument zvlášť.

príloha1:

Skupina č.0	OPCODE 'bez argumentov'	CREATEFRAME, PUSHFRAME, POPFRAME, RETURN, BREAK
Skupina č.1	OPCODE symbol	WRITE, DPRINT, PUSH, EXIT
Skupina č.2	OPCODE label	CALL, LABEL, JUMP
Skupina č.3	OPCODE label symbol symbol	JUMPIFEQ, JUMPIDNEQ
Skupina č.4	OPCODE variable	DEFVAR, POPS
Skupina č.5	OPCODE variable type	READ
Skupina č.6	OPCODE variable symbol	MOVE, INT2CHAR, STRLEN, NOT, TYPE
Skupina č.7	OPCODE variable symbol symbol	ADD, SUB, MUL, IDIV, LT, GT, EQ, AND, OR, STR2INT, CONCAT, STRLEN, GETCHAR, SETCHAR

interpret.py

Popis: Skript `interpret.py` parsuje vstupný XML súbor, kontroluje jeho syntaktickú správnosť a štruktúru a vykonáva ním definovaný kód jazyka IPPcode21. Pre parsovanie využíva knižnicu `ElementTree`, ktorá kontroluje formátovanie XML súboru a načíta jeho elementy do dátových štruktúr. Pri spustení skriptu sa spúšťa cyklus, ktorý načítava a spracováva jednotlivé inštrukcie. V prípade, že zadaný XML súbor skutočne reprezentuje kód jazyka IPPcode21 a celý beh prebehol úspešne, vracia `interpret` hodnotu 0. V prípade neúspechu skript vracia špecifický návratový kód.

Parametre:

- help: Zobrazí krátku nápovedu k skriptu `interpret.py`
- source: vstupný XML súbor
- input: súbor obsahujúci vstupy pre kód IPPcode21, v prípade, že tento parameter nie je zadaný a v kóde sa nachádza inštrukcia `READ`, načíta vstup zo štandardného vstupu

Triedy:

- Arguments – trieda pre spracovanie parametrov skriptu
- Error – spracovanie chybových hlásení
- FrameProcessor – využíva vlastný zásobník pre definíciu a prácu s rámcami
- InstructionHolder – implementuje dátovú štruktúru pre jednotlivé inštrukcie a taktiež zásobník pre volania funkcií
- Instruction – trieda pre zaobalenie inštrukcií a ich argumentov
- ParserXML – trieda pre parsovanie a kontrolu vstupného XML súboru
- Stack – poskytuje dátovú štruktúru - zásobník pre prácu s premennými

Jednotlivé triedy sú rozdelené do celkovo 5 modulov – jedného hlavného (`interpret.py`), ktorý obsahuje zároveň funkciu `Main()` a 4 submodulov, ktoré sa nachádzajú v zložke `moduls`.

test.php

Popis: Skript test.php poskytuje testovacie rozhranie pre skripty parse.php a interpret.py. Testovacie dáta je možné načítať rekurzívne z všetkých podzložiek, alebo nerekurzívne len zo zadanej zložky. Pre nájdenie všetkých testovacích súborov skript využíva príkaz zo shellu – find. Za použitia parametrov je možné špecifikovať výhradné testovanie parseru, interpretu, alebo testovanie oboch. Pri výhradnom testovaní parseru sa implicitne používa nástroj “A7soft JExamXML”, v inom prípade skript využíva príkaz zo shellu – diff. Skript na záver generuje výstupný HTML súbor, obsahujúci prehľadnú tabuľku testov a ich úspešností.

Parametre:

- help: zobrazí krátku nápovedu k skriptu interpret.py
- directory: zložka obsahujúca testy
- recursive: skript bude prehľadávať aj podadresáre
- parse-script: umiestnenie parsovacieho skriptu
- int-script: umiestnenie interpretovacieho skriptu
- parse-only: skript bude testovať len parser
- int-only: skript bude testovať len interpret
- jexamxml: umiestnenie JAR balíčku softvéru “A7soft JexamXML”
- jexamcfg: umiestnenie konfiguračného súboru pre “A7soft JexamXML”

Triedy:

- Arguments – trieda pre spracovanie parametrov skriptu
- Tester – trieda pre prehľadávanie adresárov, spúšťanie a vyhodnocovanie testov
- HTMLGenerator – trieda pre generovanie výstupného HTML súboru

HTML: Generovanie HTML kódu prebieha v poradí v akom sú spúšťané jednotlivé testy.

Výstupom je HTML stránka obsahujúca počty úspešných a neúspešných testov a tabuľku, ktorá popisuje jednotlivé testy. Prvý stĺpec tabuľky obsahuje názov testu, druhý stĺpec popisuje výsledok a očakávanú hodnotu parseru, tretí výsledok a očakávanú hodnotu interpretu a štvrtý, farebne odlíšený, popisuje celkový výsledok testu. Zelená farba indikuje úspešný test, červená neúspešný, pričom pole obsahuje dôvod neúspechu.