Implementační dokumentace k 2. úloze do IPP 2020/2021
Jméno a příjmení: Pavel Bobčík
Login: xbobci03

1 Skript interpret.py

Narozdíl od parse.php je interpret.py řešen za využití objektově orientovaného programovaní, a tedy pomocí tříd. Vyskytují se zde například třídy pro vykonání instrukcí, hlavního běhu skriptu, třídy zajišťující určitou míru generalizace kódu, ale i třídy fungující jako switch a další.

1.1 Main class

Třída `Main` má za úkol hlavní běh programu. Stará se o kontrolu argumentů, vytvoření dočasného souboru (v přídadě standartního vstupu), načtení dat a další. Dále ve smyčce volá třídu `Switcher`, jenž volá požadovanou třídu zpracovávané instrukce.

1.2 Třídy instrukcí

Jednotlivé instrukce jsou reprezentovány vlastní třídou. Zde si uchovávají své informace jako je počet a druh neterminálů. Dále se zde provádí jejich funkcionalita. Více či méně, dle jejich podobnosti s funkcionalitou jiných instrukcí, jsou jejich úkoly zobecněny a vykonány v třídách, které je generalizují. Příkladem můžou být matematické operace, které jsou vykonávány ve společné třídě `MathOperation`. Dále lze uvést relační či logické operace a další.

1.3 Pomocné třídy

Mezi tyto třídy lze zařadit již zmíněné třídy sloužící ke generalizaci, ve kterých se odehrává většina funkcionality instrukcí. Jedná se o třídy jako jsou `MathOperation`, `LogicalOperation`, `Comparsion`, `NonTermController` a další, jako jsou třídy pro práci se zásobníky.

Dále samozřejmně pomocná třída na načtení a práci s XML souborem `XmlController` či na ukončení programu, včetně úklidu dočasných souborů a výpisu chybového hlášení, třída `CleanUp`.

Samozřejmně i třída `Switcher`. Tato třída obsahuje funkci `callOpcode`, jenž má za úkol dle obdrženého parametru rozhodnout, zda obsahuje odpovídající funkci. Pokud ne, ukončí se s chybou o pokus vykonání neplatné instrukce. V opačném případě zavolá odpovídající funkci, jenž volá třídu pro danou instrukci.

Nakonec lze ještě zmínit pomocné třídy sloužící jako objekt pole.

2 Skript test.php

Skript test.php se skládá z několika funkcí a jedné pomocné třídy sloužící jako objekt pole.

2.1 Funkce test

Funkce `test` by se dala označit jako hlavní funkce, jenž na základě uvedení či neuvední parametru `--recursive` inicializuje obsah pole testovaných souborů. Dále dle kombinace parametrů `--parse-only` a `int-only` vykoná odpovídající sekvenci testů.

2.1.1 Pouze –parse-only

Provede se funkce parse, jenž vykoná převod zdrojového souboru na formát XML a porovná se chybový kód. V případě rovnosti k nule se zkontroluje i výstupní XML soubor.

2.1.2 Pouze –int-only

V případě `--int-only` dojde k vykonání funkce interpret, jenž vstupní soubor zpracuje a porovná výstupní kód. V případě bezchybného výstupního kódu (nuly) porovná i výstup scriptu s očekávaným výstupem.

2.1.3 Oba skripty

V takovém případě se nejdříve zpracuje skript parse.php, jenž nám ze vstupního souboru vygeneruje vstup do skriptu interpret.py.

2.2 Pomocné funkce

Dále se zde nachází pomocné funkce pro počáteční, průběžnou i koncovou režii.

K počáteční režii lze zařadit funkci `checkArg` jenž zkontroluje správnost argumentů a uloží zadané hodnoty.

Jako průběžnou režii lze označit funkce `errorCode` sloužící k ukončení skriptu s chybou, funkci `rcFileOperation` na získání očekávaného chybového kódu, `compareExitCodes` na porovnání chybových kódů a `createMissingFiles` pro vytvoření chybějících souborů.

A mezi koncouvou režii lze zařadit funkce `removeTmpFiles` sloužící k úklidu dočasných souborů, potřebných pro běh skriptu, a `makeHtml` pro vytvoření HTML výstupu s výsledky testů.

2.3 Třída errorArray

Třída `errorArray` slouží jako objekt pole, který nese hodnotu názvu souboru pro který chyba nastala, typ chyby a detail o chybě.