

Implementační dokumentace k 2. úloze do IPP 2020/2021

Jméno a příjmení: Josef Krušina

Login: xkrusi01

1 Úvod

Dokumentace bude rozdělena na další dvě sekce, kde se podrobněji budu věnovat souborům `interpret.py` a `test.php`. Mezi importované moduly, které jsem využil patří: `argparse` pro kontrolu argumentů `sys` pro zachycení některých výjimek, `Enum` pro výčet chyb, `ElementTree` pro procházení XML a vytvoření stromu a `re` pro regulární výrazy.

2 Interpret

V rámci tohoto projektu jsem se chtěl více sblížit s paradigmatickým OOP. Možnost využití třídových atributů a metod mi znatelně zjednodušila práci. Interpret je řízený na konci skriptu, třídy mají vlastní `main` metody, které volají metody v potřebném pořadí. Nyní provedu krátký rozbor jednotlivých tříd využitých v `interpret.py`:

ErrorTypes – Třída zastupující výčtový typ, obsahující pojmenování typů chyb a jejich návratové hodnoty. Na ni se odkazuje funkce `errorHandling`, která se stará o výpis chybové hlášky a vrácení návratového kódu.

ParseArguments – Třída nejprve kontroluje argumenty a jejich kombinace za pomoci modulu `argparse`, poté pomocí `ElementTree` kontroluje formát XML.

ParseXML – Slouží k strukturní kontrole XML, provádí lexikální a syntaktickou analýzu a zároveň se zde tvoří seznam knihoven obsahující jednotlivé instrukce. Seřazení pořadí instrukcí se děje mimo třídu.

Labels – Obsahuje dvě základní metody, umožňující práci nad labely.

InstructionStack – Využíván pro ukládání a odebrání instrukcí ze zásobníku.

Frames – Poslední pomocná třída slouží k funkcionalitě rámců, ukládání hodnot a získávání hodnot z rámců.

ParseInstructions – Využívá instance pomocných tříd k zpracování jednotlivých instrukcí. Slouží jako hlavní interpretační funkce. Nejdříve se načítají rámce, poté se prochází jednotlivé instrukce. Obsahuje mnoho metod, většina tvoří implementaci instrukcí.

Mezi nedostatky interpretu patří nedostatečná granularita při určování chyb a občasné nesprávné fungování interpretu při užívání zásobníků volání a rámců.

3 Testovací soubor

Testovací soubor třídy narozdíl od interpretu nemá. Slouží k testování `interpret.py` a `parse.php`. Využívá globálních proměnných, které nesou implicitní hodnoty pro argumenty. Nejdříve pomocí `getopt` se získají a zpracují argumenty, proběhne kontrola validních kombinací. Dál proběhne průchod složky/složek, podle toho jestli je zvolená rekurze. Funkce `sortFound()` roztřídí do různých polí soubory podle přípon, vytvoří nové soubory, pokud pro zdrojový soubor chybí. Hlavní `for` cyklus probíhá pro počet prvků v poli pro zdrojové soubory. Zde se dále rozhoduje, podle argumentů, o způsobu testování testovaných souborů. Poté funkce `printhtml` tiskne výsledné html.