Implementační dokumentace k 2. úloze do IPP 2020/2021
Jméno a příjmení: Vojtěch Fiala
Login: xfiala61

1 interpret.py

1.1 Zpracování argumentů

Skript `interpret.py` začíná svoji činnost zpracováním argumentů. Nejsou implementována žádná rozšíření, tedy skript pracuje pouze s nebonusovými parametry. Narazí-li skript na nevalidní či špatně zapsaný parametr, zahlásí chybu a skončí. Totéž se stane v případě neplatné kombinace parametrů.

Zpracování parametrů probíhá skrz regulární výrazy. V případě, že regulární výraz vyhodnotí parametr jako validní, jedná-li se o argument `--input` nebo `--source`, je dále zpracována cesta k souboru. Jestliže daný soubor není dostupný (neexistuje nebo k němu interpret nemá přístup), skript opět zahlásí chybu a končí.

1.2 Lexikální a syntaktická analýza

Jestliže jsou argumenty úspěšně zpracovány, skript načte obsah zadaných souborů, při kterém dojde k základní kontrole kostry XML souboru. Pokud byl soubor se vstupy předán jako argument, je otevřen a přesměrován na standardní vstup. Poté dochází k lexikální a syntaktické analýze. Jednotlivé instrukce jsou kontrolovány podle očekávaného typu a počtu parametrů. V průběhu analýzy jednotlivých instrukcí také dojde k případnému seřazení argumentů instrukcí. Jestliže jsou všechny instrukce validní, jsou dále seřazeny podle jejich pořadí.

1.3 Interpretace instrukcí

V případě úspěšné lexikální a syntaktické analýzy dochází k samotné interpretaci. V rámci ní projde interpret načtený zdrojový kód dvakrát. Při prvním průchodu hledá návěští, u kterých si uloží jejich pozici a z načteného kódu je smaže. Při druhém průchodu jsou zpracovávány jednotlivé instrukce. V případě, že interpret nalezne instrukci pro skok, pokusí se dané návěští najít v seznamu návěští a v případě, že je úspěšně nalezeno, získá jeho pozici a další zpracování probíhá od této pozice.

Seznam návěští, stejně jako jednotlivé rámce, jsou řešeny přes globální slovníky. Ve slovníku je uložen název a hodnota proměnné. Pro práci s rámci se dále využívají ještě pomocné slovníky sloužící k uchování typu jednotlivých proměnných. Zásobníky (rámců i proměnných) jsou řešeny přes seznamy, ke kterým je přistupováno jako k zásobníkům.

Pokud je argument instrukce typu `var`, tedy proměnná, je (mimo instrukci pro její definici, kdy je naopak do slovníku přidána) zkontrolována její existence vyhledáním ve slovníku s proměnnými a poté načtením její hodnoty a typu pro sémantické kontroly.

2 test.php

2.1 Zpracování argumentů

Po spuštění skriptu je prvním krokem zpracování argumentů, které probíhá podobně jako u skriptu `interpret.py`. Jednotlivé parametry jsou zpracovávány přes regulární výrazy. Žádná rozšíření nebyla implementována a skript tedy pracuje pouze se základními argumenty. Nalezne-li neplatný či špatně zapsaný argument, zahlásí chybu a konči svůj průběh. Totéž nastane v případě nevalidní kombinace parametrů. V případě, že některý ze souborů předaných argumentem není dostupný, skript taktéž zahlásí chybu a končí.

2.2 Načtení jednotlivých testů

V případě správného zadání parametrů skript dále načte dostupné soubory v závislosti na zadaných parametrech. Jestliže byl zadán argument `--recursive`, je vytvořený rekurzivní iterátor typu `RecursiveIteratorIterator`, jinak je iterátor typu `DirectoryIterator`. Jednotlivé iterátorem nalezené soubory jsou dále porovnány a jestliže mají koncovku `.src`, jsou vyhodnoceny jako soubory s testovými vstupy a přidány do pole testových souborů. Celé pole testových souborů je poté ještě seřazeno podle abecedy, nehledě na velikost písmen.

2.3 Testování

Před započetím testování je na standardní výstup vypsána hlavička HTML souboru. Poté jsou testy zpracovávány v cyklu, který projde všechny testové soubory. Pro každý test dojde ke kontrole, jestli existují pomocné soubory k němu (vstup, výstup a návratová hodnota), které jsou případně vytvořeny. V závislosti na zadaných parametrech jsou poté spouštěný kombinace jednotlivých testů. V rámci každého testu je načten obsah souboru s koncovkou `.rc` obsahující očekávanou návratovou hodnotu, která je porovnána s výsledkem testu. Po každém testu jsou zaznamenány výsledky porovnání očekávané návratové hodnoty a získané návratové hodnoty (příp. také výsledky porovnání výstupu), které jsou vypsány na standardní výstup v HTML formátu. V případě, že je test úspěšný, je také navýšeno počítadlo úspěšných testů. Po zpracování všech testových souborů je vypsán celkový výsledek zahrnující počet úspěšných testů, neúspěšných testů a testů celkem. Výsledky se nacházejí na konci vygenerovaného HTML výstupu, ale lze k nim přistoupit i rychleji za pomoci hypertextového odkazu nacházejícího se na začátku HTML výstupu. Nakonec je také na standardní výstup vypsáno ukončení HTML souboru.

Průběh skriptu je ukončen kontrolou, jestli zůstaly nějaké pomocné soubory, které jsou případně smazány.