

1 Interpretácia

1.1 Analýza vstupnej XML reprezentácie

Interpretácia XML reprezentácie kódu začína v súbore `interpret.py`. Po načítaní a spracovaní argumentov je vytvorená instancia triedy `CodeParser`, definovanej v súbore `codeparser.py`. Zavolá sa jej metóda `readInput`, ktorá sa postará o načítanie vstupu. Následne je volaná jej metóda `parseCode`, zabezpečujúca analýzu vstupnej XML reprezentácie kódu.

Na analýzu inštrukcií sú využívané triedne atribúty `opcodes`, `expand_types` a `value_patterns`. Atribút `opcodes` je slovník, ktorého kľúčmi sú operačné kódy inštrukcií a príslušnou hodnotou je zoznam neterminálov, predstavujúcich operandy. Neterminály slúžia ako kľúče do slovníka `expand_types`. Danému kľúču je priradená n-tica, obsahujúca kľúče do slovníka `value_patterns`, kde sú uložené regulárne výrazy pre kontrolu hodnôt neterminálov.

Objekt triedy `CodeParser` má atribút `interpret`, odkazujúci na instanciu triedy `CodeInterpret`, do ktorej sú postupne ukladané jednotlivé inštrukcie. Počas analýzy sa volajú metódy tejto triedy `newInstruction(name)` a `addArgument(value, num)` na vytvorenie nového záznamu o inštrukcii a postupné pridávanie jej operandov. Po dokončení kontroly je zavolaná metóda `finishInstruction`, ktorá prevedie inštrukciu do internej reprezentácie.

1.2 Interpretácia

Po dokončení analýzy sa zavolá metóda triedy `CodeInterpret` `run`. Tá prechádza zoznam inštrukcií a volá príslušné metódy. Trieda `CodeInterpret` definuje metódu, modifikujúcu stav objektu tejto triedy, pre každú inštrukciu jazyka IPPcode21.

Medzi instančné atribúty patrí napríklad počítadlo inštrukcií, odkaz na dočasnú a globálnu tabuľku rámcov, zoznam lokálnych tabuliek rámcov, zásobník volaní alebo zásobník premenných pre rozšírenie `STACK`.

Dalším atribútom je slovník `label_dict`, ktorý pre každé návstie definuje hodnotu čítača inštrukcií. Tento slovník je naplnený už pri postupnom pridávaní inštrukcií ešte pred samotnou interpretáciou a to kvôli dopredným skokom. Následne sú jeho hodnoty upravené tak, aby nedošlo k chybám pri nekompletnej postupnosti atribútov `order` jednotlivých inštrukcií.

1.3 Práca s rámcami a premennými

Rámce sú reprezentované objektami triedy `Frame`, definovanej v súbore `codeinterpret.py`. Tieto objekty si pamätajú premenné, ktoré v danom rámci boli definované a to pomocou slovníka `vars`. Kľúčom je meno premennej a odpovedajúcou hodnotou je instancia triedy `Variable`, definovanej v súbore `variable.py`.

Objekty typu `Variable` si pamätajú svoj typ a hodnotu, ponúkajú metódy pre kontrolu typov, či zmenu svojej hodnoty a typu. Ďalej implementujú metódy pre preťaženie aritmetických, relačných a logických operátorov, ktoré sa starajú aj o kontrolu typovej kompatibility a inicializácie premenných.

Implementácia tried `Frame` a `Variable`, tak umožňuje zapísať kód metód, predstavujúcich jednotlivé inštrukcie v triede `CodeInterpret` pomerne jednoduchým spôsobom.

2 Testovací rámec

2.1 Spracovanie argumentov a získanie požadovaných súborov

Spracovávanie argumentov sa odohráva vo funkcii `parseArguments` definovanej v súbore `parse_arguments.php`. Využíva funkciu `getopts`, poprípade kontroluje zakázané kombinácie parametrov. Výsledné pole `$options` je následne modifikované pomocou funkcie `setDefaultParams`, ktorá do neho doplní implicitné hodnoty parametrov zo zadania.

Pomocou objektov tried `RecursiveDirectoryIterator`, `RecursiveIteratorIterator` alebo `DirectoryFilter` (definovanej v súbore `DirectoryFilter.php`) a `RegexIterator` je získaný iterátor obsahujúci všetky požadované `.src` súbory.

2.2 Vykonávanie testov

Pred samotným vykonávaním testov je vytvorené pole `$tables`, ktorého indexmi sú mená jednotlivých adresárov a hodnotami sú instance triedy `Table`, popísanej v súbore `Table.php`.

Objekty triedy `Table` predstavujú tabuľku s výsledkami testov pre každý adresár. Ponúkajú metódy `createTable`, `addTest`, `getTitle`, `getTable` a `getSummaryRow`. `createTable` slúži na vytvorenie HTML elementu predstavujúceho tabuľku. Metódou `addTest` je možné pridávať záznamy o vykonaných testoch. Metódy `getTitle`, `getTable` vracajú HTML elementy predstavujúce nadpis tabuľky a výslednú tabuľku. Pomocou metódy `getSummaryRow` je možné vytvoriť riadok pre aktuálny adresár do tabuľky so súhrnom výsledkov pre jednotlivé adresáre.

Počas vykonávania testov sa prechádzajú získané súbory, vytvárajú sa objekty typu `Table` pre jednotlivé adresáre a pre každý testovací súbor sa vytvorí instance triedy `Test` alebo od nej odvodených tried `ParseTest` a `IntTest`, podľa spôsobu vykonávania testov.

Objekty týchto tried poznajú názvy testovaných skriptov, testovaného súboru a taktiež instance triedy `Table`, do ktorej bude uložený výsledok testu. Trieda `Test` definuje metódu `setup`, ktorá pomocou ďalších metód získa očakávaný návratový kód a výstup. Hlavným rozdielom tried `Test`, `IntTest` a `ParseTest` je spôsob implementácie metódy `run`, ktorá slúži na samotné spustenie a vyhodnotenie testu.

V metóde `run` triedy `Test` je najprv spustený PHP skript analyzátora a jeho výsledok je uložený do pomocného súboru. V prípade bezchybného prekladu do XML reprezentácie kódu je potom spustený skript určený na interpretáciu. Jeho vstupom je výstupný súbor z analyzátora a výstup interpretácie je uložený do ďalšieho pomocného súboru. Tento výsledok je porovnaný s referenčným `.out` súborom pomocou nástroja `diff`. Následne je zavolaná metóda `addTest` instance triedy `Table`, ktorej je predaný výsledok testu, získaný návratový kód, získaný výstup a taktiež referenčný návratový kód a výstup testu. Tieto údaje sú využité pri vytváraní vyskakovacieho okienka s detailným popisom jednotlivých testov na výslednej stránke. Nakoniec sú odstránené všetky vytvorené pomocné súbory.

Metóda `run` v triedach `IntTest` a `ParseTest` je implementovaná podobným spôsobom, rozdiel je len v spúšťaní jednotlivých skriptov a porovnávaní výsledkov.

2.3 Výstup

Výstupné HTML je vytvorené v instancii triedy `DOMDocument`. Postupne sú do neho pridané metainformácie, hlavička, názov stránky a nadpis. Ďalej sa pripojí element `style`, ktorého obsah je uvedený v súbore `html_elements.php` spolu s elementom `script`. Tabuľky sú generované pomocou funkcií definovaných v súbore `html_tables.php`, ktoré prechádzajú zoznam `$tables` a postupne pridávajú tabuľky s výsledkami testov do výstupného `DOMDocument` objektu.