

Program slúži na interpretáciu vstupného XML súboru. Implementovaný je podľa vzoru „Strategy“ určeného pre objektovo orientované programovanie. Tento vzor je kvôli sprehl'adneniu súborov a kvôli lepšej orientácii implementovaný iba abstraktne pomocou `switch` v jednej triede `Interpreter`.

Postup programu

1. Načítanie vstupných súborov a zmena XML súbora na `DOMDocument` pomocou kontajnera.
2. Prvý prechod `DOMDocument`-om (uloženie čísel inštrukcií, uloženie pozície a hodnoty inštrukcie `LABEL` do premennej `array $labels`).
3. Druhý prechod `DOMDocument`-om pomocou `while` cyklu, počas ktorého prebieha interpretácia. V tomto kroku je implementovaný vzor „Strategy“, ktorý pre každú inštrukciu vytvorí samostatnú triedu `Instruction` a v samotnej triede už prebehne algoritmus určený pre konkrétny operačný kód (`OPCODE`). Nasledujúca hodnota inštrukcie je určená buď inkrementáciou premennej, ktorá ukazuje na aktuálnu inštrukciu alebo pri skokových inštrukciách pomocou hodnoty premennej `array $labels`.

Triedy

`Interpreter` – hlavná trieda, prebieha v nej prvý aj druhý prechod `DOMDocument`-om, riadi skoky medzi inštrukciami, obsluhuje výnimky vyvolané chybou počas chodu programu.

`Instruction` – obsahuje abstraktný „Strategy“ vzor, ktorý sa od pôvodného vzoru líši tým, že všetky „podtriedy“ sú implementované priamo v triede `Instruction` a určenie, ktorý algoritmus kedy použiť vykonáva `switch`. Trieda obsahuje taktiež overenie syntaktickej správnosti inštrukcií.

`Argument` – trieda určená pre uchovanie argumentov a overenie ich syntaktickej správnosti. Táto trieda obsahuje okrem iného aj funkcie `getType()` a `getDeepType()`. Rozdiel medzi nimi je v tom, že zatiaľ čo `getType()` vráti aktuálny typ terajšieho argumentu (takže aj hodnotu `var`), `getDeepType()` iteruje medzi všetkými ďalšími premennými na ktoré ukazuje a hľadá posledný typ, ktorý je iný ako `var`. Ak nič nenájde, vráti buď typ `nil` Alebo chybu podľa konkrétneho prípadu.

`ObjectsContainer/Frame` – implementácia rámca, použité pre `Global frame`, `Temporal frame` a `Local frame`.

`ObjectsContainer/Stack` – implementácia zásobníka

Rozšírenia

`STACK` – rozšírenie je implementované priamo v triede `Instruction`, kde sú do `switch`-u pridané jednotlivé operandy (napríklad `CLEAR`, `ADD`, `SUB`,...). Pre fungovanie rozšírenia sú vytvorené v triede

Interpreter dve nové premenné `$dataStack` (obsahuje hodnoty) a `$dataStackTypes` (obsahuje typy hodnôt v `$dataStack`).

Testovanie

Na testovanie som použil vlastné testy, prispôsobené testy z minulého roku a taktiež test PHPStan na leveli 6 aj leveli 0. Všetky testy boli spustené na serveri Merlin.

Diagram tried

V tomto diagrame nie sú vypísané chybové triedy (Exceptions), taktiež ani všetky parametre a metódy použité v implementácii. Do diagramu som vložil iba tie dôležité pre pochopenie zvolenej architektúry.

