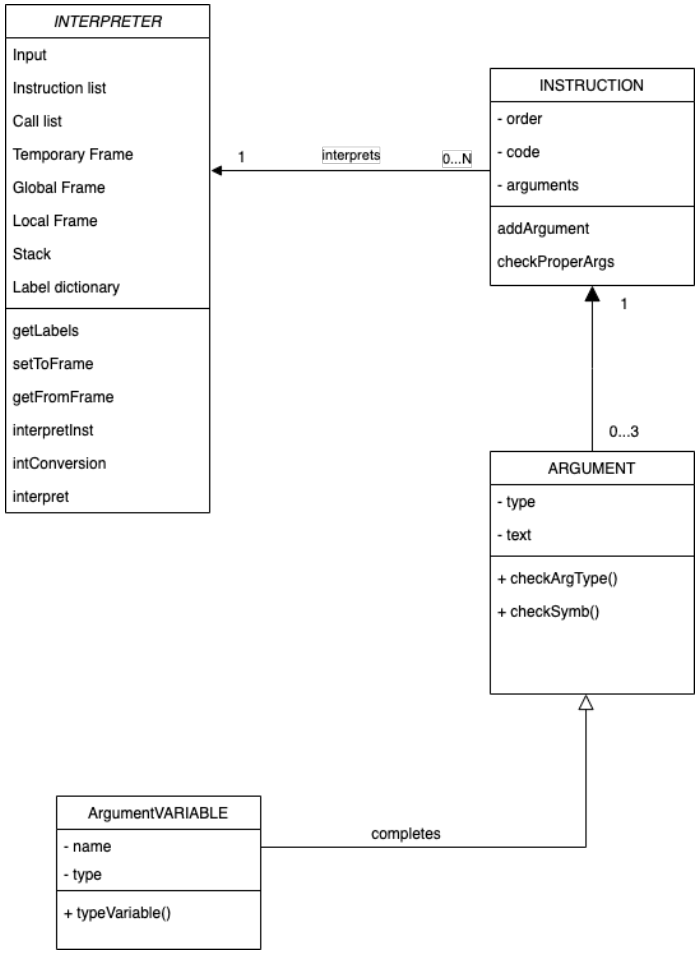
Implementační dokumentace ke 2. úloze do IPP 2022/2023
Jméno a příjmení: Martin Soukup
Login: xsouku15

Filozofie návrhu

Interpret získá potřebné argumenty, případně vypíše nápovědu. Při průchodu XML kontroluje správnost Instrukcí, argumentů a formátuje vstup pro ulehčení práce. Data se ukládají do třídních proměnných podle logiky návrhu a zapouzdří se. Třída interpretující instrukce tak bude mít všechna data vždy u sebe.

Logika tříd

Hlavní třída `Interpreter` interpretuje instrukce uložené v listu instrukcí, průchod tady tímto listem umožňuje kontrolovat na jakém indexu se interpretovaná instrukce nachází, skoky mezi instrukcemi se tímto zjednodušují. Třídní proměnné poté obsahují vstupní soubor, list volání, všechny rámce, zásobník a slovník návěští. Interpretace instrukcí je implementována jako tři různé funkce podle počtu argumentů, pro zjednodušení je v návrhu zaznamenána jen funkce `interpret`.
Třída `instruction` popisuje samotnou instrukci, její seřazení v rámci XML, název a argumenty. Argumenty instrukce jsou implementovány jako slovník s kódem argumentu a objektem třídy `Argument`, dochází tady k zapouzdření celého objektu jako třídní proměnnou. Funkce třídy umožňují přidat argument k instrukci a kontrolu počtu argumentů.
Třída `Argument` a `ArgumentVariable` popisují argumenty instrukce, jejich typ a hodnotu. Pokud je argument proměnnou tak třída ArgumentVariable doplňuje vlastnosti třídy Argument o název proměnné a funkci pro přidání typu vloženého do proměnné.



Interní reprezentace

Argumenty jsou získány pomocí `Argparse`, jsou podporovány jen dlouhé verze argumentů. Zdrojový soubor s XML reprezentací a vstupní soubor jsou v rámci funkce `main` parsovány. Je vytvořen objekt interpretu. Následuje kontrola správnosti struktury XML, kontrola hlavičky, jazyku. Program kontroluje, jestli dvě instrukce nemají stejné pořadí a jestli mají správné atributy. Po kontrole instrukce se vytváří objekt instrukce a pokračuje se v kontrole XML, zkontroluje se správnost názvu instrukce a poté se přechází na argumenty, v cyklu se kontroluje každý argument pro správnost atributů. Následně se vytváří objekt argumentu s formátovanou hodnotou. Každý argument se poté přidá k instrukci a celá instrukce se přidá do listu instrukcí interpretu.
Interpretace zdrojového kódu začíná seřazením instrukcí podle jejich pořadí a získáním všech návěští napříč zdrojovým kódem. Následně se cyklí skrz list instrukcí a uchovává se index instrukce, která se interpretuje. Po každé interpretaci instrukce, funkce pro interpretaci vrací pozici instrukce nebo pokud se jedná o skok na návěstí nebo volání, vrací pozici návěští nebo volání.
Proměnné jsou ukládány do jednotlivých rámců příslušnými funkcemi. Jsou ukládány do slovníku kde klíčem je název proměnné a hodnota je objekt `Argument`, takže u rámce je možný přístup k hodnotě i typu uložené proměnné. Výjimka je lokální rámec, který je implementován jako list, do kterého se ukládají dočasné rámce. Při přístupu k proměnné se poté pracuje s posledním přidaným rámcem do listu.

Rozšíření

Jediné rozšíření implementované je rozšíření STACK. Implementace zásobníkových funkcí je téměř totožná jako u klasických funkcí, až na to že se pracuje s posledními dvěma hodnotami na zásobníku místo argumentů z XML.