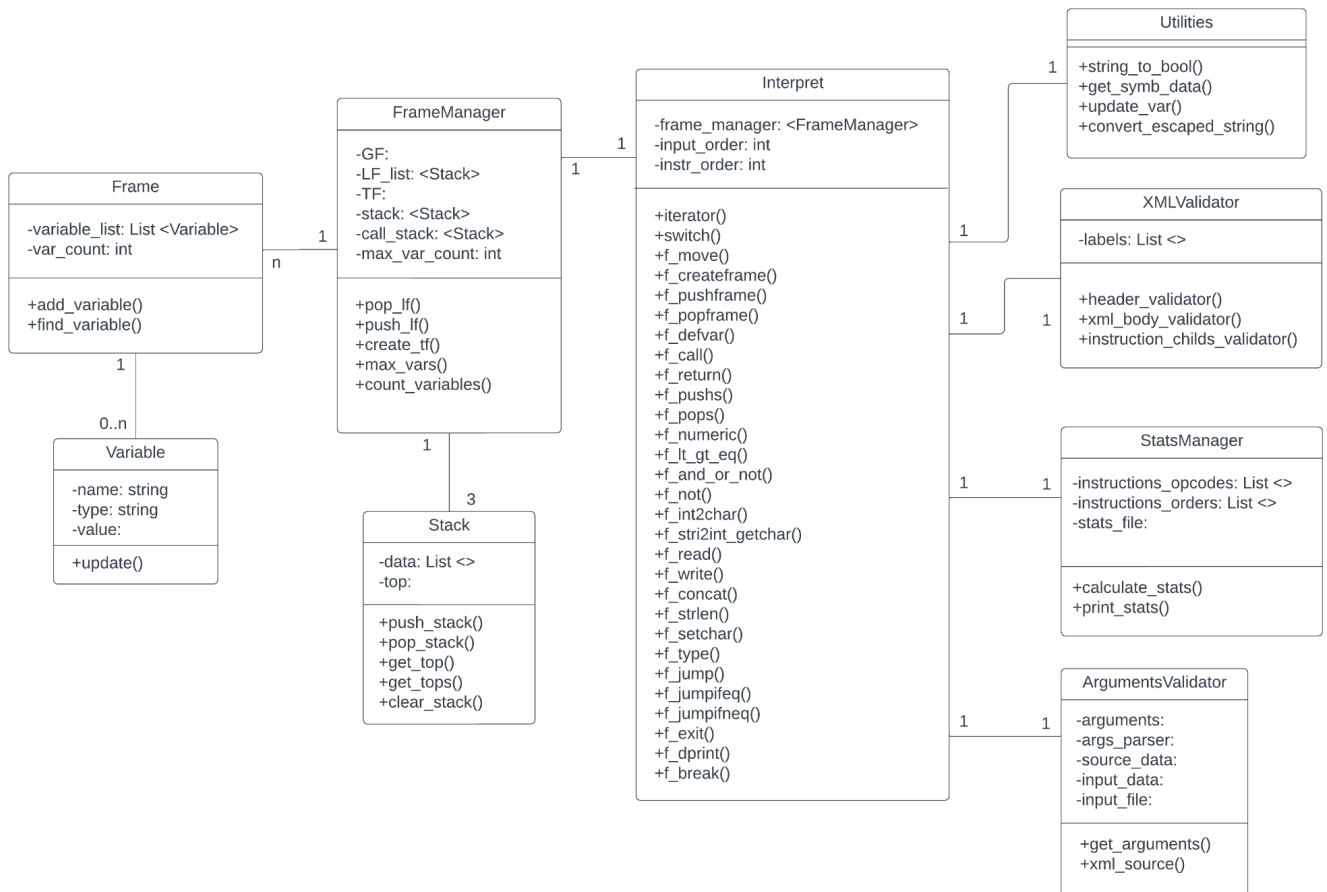


1. Implementace interpret.py

Interpret byl vytvořen podle objektového návrhu. Jednotlivé třídy popisuje níže zobrazený diagram tříd.



2. Validator

Před použitím řídicí metody celého programu, jsou použity metody dvou validačních tříd.

Tou první třídou je **ArgumentValidator**, která v atributech uchovává data ze souborů zadaných při spuštění programu. Metoda `get_aguments()` podle zadaných argumentů provede validaci, uložení zadaných argumentů, případně výpis nápovědy. Následuje metoda `load_data()`, která načte zadané soubory a seřadí instrukce XML podle atributu `order`.

Druhá validační třída je **XMLValidator**, kde jsou provedeny postupně metody `header_validator()`, `xml_body_validator` a z ní `instruction_chilids_validator` pro kontrolu správnosti XML souboru. Vyskytuje se tu také argument `labels`, což je list pro ukládání všech návěští ze souboru.

3. Interpret + Utilities

Hlavní třídou je **Interpret**, kde se nachází metoda `iterator()`, která řídí celý běh programu.

Jednotlivě prochází instrukce, ukládá jejich argumenty a volá metodu `switch()` s jménem instrukce, ta podle zadané instrukce zavolá metodu tomu odpovídající a předá argumenty podle jejich

typu. Tyto metody (vypsáné v UML diagramu) svou funkcí odpovídají zadání, navíc je zde kontrolován také typ a validita hodnoty. Výjimku tvoří `f_numeric(var, symb, symb2, oper)`, která podle hodnoty uložené v `oper` provede ADD, SUB, MUL nebo IDIV. Stejně tak i instrukce LT/GT/EQ, AND/OR a STR2INT/GETCHAR jsou sdruženy do společných metod, kvůli velké podobnosti.

Ke třídě `Interpret` je vytvořena také třída pomocná nazvána `Utilities`. Ta obsahuje 4 funkce, které jsou zpravidla volány v metodách ve tvaru `f_name`. Dvě méně rozsáhlejší `string_to_bool()` - převede string na hodnotu bool a `convert_escaped_string()` - převede string na string bez escape sekvencí. Dále je tu `get_symb_data()`, která pracuje s třídou `FrameManager` a podle toho, zda jde o konstantu nebo proměnnou (v tomto případě otestuje zda existuje v daném rámci), vrátí dvojici hodnota a typ. Poslední pomocná metoda `update_var()` provede kontrolu, zda může ukládat do rámce a dané proměnné, poté zadanou hodnotu i s typem uloží.

4. FrameManager + Frame + Variable

Místem, kde se nachází tzv. paměť programu je třída `FrameManager`, zde jsou napojeny třídy `Frame` a `Stack` k různým účelům. Rámce GF a TF jsou třídou `Frame`, LF je vytvořen jako zásobník rámců. `Stack` se používá k zásobníkovým instrukcím pod atributem `stack` a do `call_stack` se ukládají instrukce CALL. Dále následují metody pro vložení a vyjmutí rámce LF, vytvoření dočasného rámce TF.

Každý `Frame` obsahuje list `Variables`, jejich počet a metody pro přidání nebo hledání konkrétní proměnné. A každá `Variable` obsahuje jméno, typ, hodnotu a metodu `update()`.

5. Stack

Třída `Stack` je tradiční zásobník, obsahuje data, ukazatel na vrchol zásobníku, metody `top/push/pop/clear`. Navíc také `get_tops()`, která vrátí dvě vrchní hodnoty (pro účely rozšíření).

6. Rozšíření

STATI - je implementováno

Pro rozšíření STATI byla navržena třída `StatsManager`, ve které dochází ke kalkulaci všech statistik vyjma maximálního počtu proměnných, které je ukládáno ve třídě `FrameManager` a počítáno vždy při vytvoření nové proměnné metodou `max_vars()`.

Atribut `instruction_opcodes` se plní během validace XML souboru, zatímco `instruction_orders` se plní v průběhu programu. Po spočítání statistik dojde ke kontrole, zda je zadán přepínač `stats` a případnému vypsání vybraných statistik do souboru.

STACK - je implementováno

Zásobníkové instrukce jsou prováděny obdobně jako klasické, jen při volání metod ve formátu `f_name` se namísto `var` použije string "stack", abychom při ukládání výsledku instrukce věděli, že se jedná o zásobníkovou instrukci a do `symb` nahrajeme pomocí metody `get_top()` nebo `get_tops()` hodnoty ze zásobníku.

FLOAT - není implementováno

Rozšíření FLOAT bylo plánováno řešit přímo v metodách určených k manipulaci s čísly (ADD/SUB/MUL a další), kde by byla provedena validace dekadického i hexadecimálního zápisu čísla s desetinnou čárkou a provedení instrukce.