

Implementační dokumentace k 2. úloze do IPP 2023/2024

Jméno a příjmení: Tomáš Dolák

Login: xdolak09

ÚVOD

Cílem projektu bylo vytvořit interpret v jazyce **php8.3**, který má za úkol zpracovat a vykonat instrukce jazyka **IPPcode24**. Instrukce jsou předávány interpretu ve formátu **XML** (Extensible Markup Language) a jejich zpracování je provedeno podle parametru order, obsaženého v instrukci.

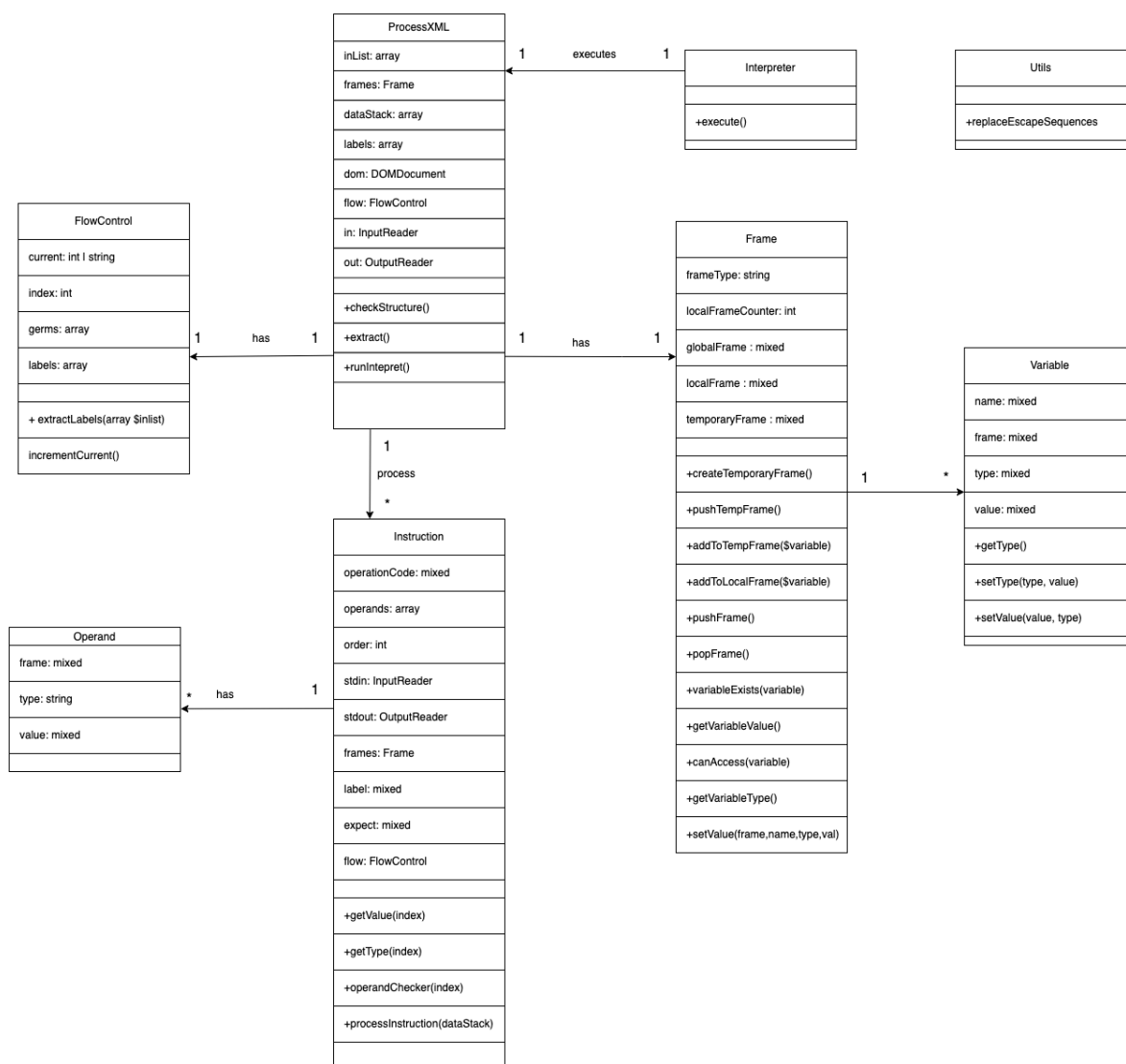
VYUŽITÉ KNIHOVNY A NÁSTROJE

Pro implementaci interpretu **IPPcode24** nebyly využity žádné externí knihovny. V projektu jsou použity pouze vestavěné třídy nabízené jazykem php, jako je například **DOMDocument** nebo **DOMXPath**. Dále bylo využito **ipp-core** pro práci se vstupními/výstupními operaci, výjimek a spouštění skriptu.

POUŽITÉ TECHNOLOGIE

Pro implementaci tohoto projektu byly využity technologie php8.3, Docker, composer a PHP Stan.

UML DIAGRAM



POPIS TŘÍD

Variable: Abstrakce proměnné, obsahuje v sobě datový typ, identifikátor, rámec, do kterého patří a svoji hodnotu. Nabízí operace **setType()** – nastavující typ proměnné a **setValue()** – nastavující hodnotu.

Frame: Frame představuje rámce používané během běhu interpretovaného programu. Obsahuje globální rámec (**globalFrame**), lokální rámce (**localFrame**), dočasný rámec (**temporaryFrame**) a čítač lokálních rámců (**localFrameCounter**) obsahující aktuální počet lokálních rámců.

Operand: Představuje operandy instrukce. Obsahuje svůj typ, hodnotu a rámec.

Instruction: Třída Instruction, představuje instrukci. Každá instrukce jako atributy obsahuje svůj operační kód (**operationCode**), operandy (**operands**), své pořadí (**order**). Dále instrukce obsahuje odkazy na standardní vstup (**stdin**) a standardní výstup (**stdout**), aby v případě potřeby mohla provádět vstupní a výstupní operace, odkaz na rámce (**frames**) pro dohledání platnosti, hodnoty a typů proměnných. Dále instrukce nutně obsahuje odkaz na návěští (label), které potřeba pro skokové instrukce. Instrukce obsahuje také odkaz na tok programu (**flow**) a pole obsahující aktuální očekávané operandy. Instrukce nabízí svým protokolem operace pro vykonání instrukce – **processInstruction()**, kontrolu operandů – **operandChecker()** a operace pro získání hodnoty a typu operandu **getValue()**, **getType()**.

FlowControl: FlowControl je jedna z nejdůležitějších tříd v projektu, představuje tok programu. Tato třída v sobě obsahuje aktuální index vykonávané instrukce (**current**), zarážky (**germs**), které udávají kde má program pokračovat po návratu ze skoku a pole návěští (**labels**) obsahující odkazy na seznam návěští, na která je potenciálně možno v kódu skočit. Obsahuje metodu pro extrakci návěští ze seznamu instrukcí (poznámka: tato operace je prováděna pokaždé po převedení vstupního XML do interní reprezentace) a dále obsahuje metodu pro posun na další instrukci (tedy inkrementaci **current**).

ProcessXML: Tato třída spojuje převod vstupního XML do interní reprezentace a vykonání interpretace kódu. Obsahuje pole instrukcí (**inList**), pole rámců (**frames**), datový zásobník pro instrukce pracující s datovým zásobníkem (dataStack), pole návěští (**labels**), XML dokument (**dom**), tok programu (**flow**) a odkazy na standardní vstup a standardní výstup (**stdin**, **stdout**). Třída nabízí operace pro kontrolu vstupního XML **checkStructure()**, extrakci dat z XML **extract()** a vykonání programu **runInterpret()**.

Interpret: Třída dědící z abstraktní třídy **AbstractInterpret**, poskytuje originální **InputReader** a **OutputWriter** pro práci se standardním vstupem a standardním výstupem a nabízí operaci **execute()**, které zajistí celý chod interpretu.

TESTOVÁNÍ PROJEKTU

Testování projektu interpretu bylo prováděno poskytnutých referenčních testech **supplementary-tests** na kterých se neprojevíly možné nedostatky implementace. Absence nedostatků, které mohou nastat však není vyloučena.