

Implementační dokumentace k 2. úloze do IPP 2023/2024

Jméno a příjmení: Radek Libra

Login: xlibra02

Úvod

Tento dokument popisuje implementaci interpretu nestrukturovaného imperativního jazyka IPPcode24 v Jazyce PHP 8.3. Interpret je integrován do poskytnutého rámce ipp-core a umožňuje zpracování a vykonávání kódu zadaného ve formátu XML.

Struktura interpretu

Interpret je modularizovaný do několika základní tříd:

Třída Interpreter

Tato základní třída interpretu, která rozšiřuje abstraktní třídu AbstractInterpreter. Ve dvou cyklech zajišťuje zpracování vstupního XML dokumentu a sekvenční vykonávání instrukcí. V první cyklu jsou inicializovány jednotlivé instrukce pomocí třídy InstructionCreator. Při druhém cyklu zajišťuje vykonávání jednotlivých instrukcí a spravuje stavy programu, jako jsou například skoky. Tato třída uchovává instanci třídy ProgramState, Seznam všech instrukcí a zásobník pro volání funkcí.

Třída ProgramState

Tato třída je zodpovědná za správu všech stavů během vykonávání programu. Ukládá informace relevantních stavech v průběhu interpretace. Mezi její hlavní funkce patří:

- Správa paměťových rámců: Udržuje globální, lokální a dočasný paměťový rámec. Globální rámec je dostupný během celého běhu programu, zatímco lokální rámce jsou dynamicky spravovány na zásobníku rámců podle potřeby programu.
- Správa proměnných: Metody jako setGlobalVariable, getGlobalVariable, a další, umožňují přidávání, aktualizaci a získání proměnných v rámci jednotlivých paměťových rámců.
- Správa návěstí: Umožňuje přidání a získání indexu návěstí, což je klíčové pro implementaci skokových instrukcí.

Třída InstructionCreator

Statická třída, která na základě XML specifikace (operační kód instrukce a argumenty instrukce) dynamicky vytváří konkrétní instance instrukcí, které jsou poté znovu předány třídě Interpreter, která si je ukládá pro pozdější použití. Třída využívá jednoduchý „switch“, který rozhoduje, která instance instrukce má být vytvořena.

Abstraktní třída Instruction

Slouží jako abstraktní základ pro všechny instrukce, definující společné rozhraní a pomocné metody při zpracování argumentů jednotlivých instrukcí. Tato abstraktní třída implementuje dvě hlavní metody. Tyto metody jsou resolveOperand a resolveOperandType, které zajišťují získání informací o jednotlivých instancích třídy Variable. Každá instrukce zadaného jazyka má v této implementaci vlastní třídu, která implementuje specifické chování dané instrukce.

Třída Variable

Představuje základní strukturu pro uchovávání a správu proměnných. Každá instance Variable obsahuje informace o hodnotě a typu proměnné a implementuje základní metody pro práci s těmito proměnnými jako např. získávání a ukládání hodnoty nebo typu hodnoty proměnné.

Všechny tyto třídy jsou navrženy tak, aby spolupracovaly na zpracování a vykonávání programu definovaného v jazyce IPPcode24. Umožňují efektivní správu paměti, správnou interpretaci a vykonání instrukcí, a správu běhového prostředí, což všechno společně přispívá k modularitě a rozšiřitelnosti kódu.

Zpracování XML dokumentů

XML dokumenty jsou zpracovány pomocí PHP DOM parseru, který umožňuje efektivní manipulaci s XML strukturou. Instrukce jsou převedeny na odpovídající instance rozšiřující již zmíněnou třídu `Instruction`.

Chybové stavy

V případě detekce chyby je vyvolána jedna z implementovaných výjimek, kterou jsou implementovány jako rozšíření třídy `IPPEException`. Každá z těchto instancí má specifický návratový kód, který odpovídá předem určeným návratovým kódům z třídy `ReturnCodes`.

Problematické části v této implementaci

Tato implementace se setkala s řadou výzev, zejména v oblastech správy paměťových rámců, dynamického typování a řízení toku programu. Správa různých typů paměťových rámců vyžadovala pečlivé návrhy, aby bylo zajištěno správné manipulace s proměnnými v dynamickém prostředí IPPcode24. Dynamické typování v PHP ztěžovalo zajištění striktní typové kontroly vyžadované jazykem IPPcode24, což vyžadovalo robustní kontrolní a konverzní mechanismy pro každou datově manipulativní instrukci. Navíc, implementace skokových instrukcí vyžadovala precizní řízení toku programu, což bylo komplikováno potřebou správně zacházet s pozicemi instrukcí a volacími zásobníky. Tyto oblasti byly identifikovány jako klíčové problematické části, které vyžadovaly specifické řešení a často byly zdrojem chyb v průběžných testech interpretu.

Diagram Tříd

Z důvodu přehlednosti vynechány třídy výjimek. V diagramu jsou pouze znázorněny třídy instrukcí, jelikož má každá instrukce svoji třídu, tudíž by se snížila přehlednost diagramu.

