

Dokumentace k druhé úloze IPP

Jméno: Alexej Beňuš

Login: xbenus01

Účel programu

Účel programu je interpretace jazyka IPPcode23.

Třídy

Arg

- Pro reprezentaci argumentů
- instanční proměnné: argType, value, number

Label

- Pro reprezentaci labelů a jejich umístění v kódu
- instanční proměnné: Name, number

LabelList

- Pro jednodušší implementaci labelů
- instanční proměnné: Labels

Var

- Pro reprezentaci proměnných
- instanční proměnné: Name, type, value

Frame

- Pro reprezentaci framů
- instanční proměnné: Variables, defined

FrameList

- Pro jednodušší implementaci LF
- instanční proměnné: Frames

StackData

- Pro reprezentaci dat na zásobníku
- instanční proměnné: Value, type

Instruction

- Pro reprezentaci instrukcí
- Kontrola a provádění
- třídí proměnné: GF, TF, LF, stack
- instanční proměnné: name, number, args

Interpret

- hlavní tělo programu
- třídí proměnné: instructions, numOfInstructions, labels
- instanční proměnné: xmlCode, inputs, instructions_counter

Funkce programu

První věc, co program udělá je porovnání argumentů a podle toho nastavení příslušných proměnných.

Poté vyhodnotí správnost XML kódu.

Nyní přečte všechny instrukce, aby si mohl zapsat návěští do tabulky, kvůli řízení programu.

Po všech zvyše potřebných přípravách začne samotné vykonávání instrukcí.

Na každou instrukci je zavolána funkce *execute()*. Po zavolání *execute()* si jednotlivé instrukce volají své vlastní funkce.

Všechny funkce fungují na stejném principu:

- Zjištění existence proměnných
- Typová kontrola
- Provedení příkazu

V případě zjištění chyby je program ukončen náležitou návratovou hodnotou.

V případě úspěchu pokračuje program dále ve čtení instrukcí.

Pro řízení programu je použita návratová hodnota funkce *execute()*, ve které se v případě skoku posílá číslo instrukce na kterou chce program skočit. Tato hodnota se zpracuje v Interpretu, kde se s ní přepíše čítač instrukcí.

