Implementační dokumentace k 2. úloze do IPP 2020/2021

Jméno a příjmení: Vasil Poposki

Login: xpopos00

K 2. úloze byl implementován skript interpret.py, který provádí interpretaci programu v jazyce IPPcode21 na základě jeho XML reprezentace. Během implementace jsem se nejvíc zaměřil na objektově-orientovaný přístup. Byly vytvořené dvě třídy – třída instrukce(`class Instruction`) a třída operandu instrukce(`class Argument`). Třída `Argument` obsahuje atributy pro typ a hodnotu jednotlivého operandu instrukce a dvě metody:

- `check_type()` – Metoda pro kontrolu typu operandu, která vrací hodnotu True, pokud hodnota argumentu odpovídá danému typu. Pro kontrolu byli použity regulární výrazy.
- `conv_string()` - Konverze escape sekvencí v řetězci na odpovídající ASCII hodnoty.

Třída `Instruction` obsahuje následující atributy: operační kód, pořadí instrukce a seznam operandů - instancí třídy `Argument` a dvě metody:

- `instr_argument(arg_type, arg_value)` - Přidá načtený operand do seznamu operandů - vytvoří novou instanci třídy `Argument`.
- `instr_valid()` - Provádí lexikální a syntaktickou kontrolu elementů a atributů ve vstupním XML souboru.

Popis toku programu:

Pro parsování vstupního XML souboru jsem použil modul `xml.etree.ElementTree`, který umožňuje vytváření stromové struktury z XML souboru. Po parsování se provede kontrola.

- `xml_control(root)` – Kontrola elementů, textových atributů, pořadí instrukcí apod.

Pro každou nově načtenou instrukci vytvoří se nová instance třídy `Instruction`. Pokud kontrola proběhla v pořádku, instance se přidá do seznamu `instructions`. Následně se provede samotná interpretace instrukcí (funkce `interpret()`). Pomocné funkce pro interpret:

- `input_getdata(pos, arg_type)` - Pomocná funkce k interpretaci instrukce READ. Nalezne v seznamu vstupů pro interpretaci hodnotu na aktuální pozici a zkontroluje, zda typ odpovídá hodnotě.
- `label_order(label_name)` - Pomocná funkce k nalezení pozici v programu pro skokové instrukce.

Interpret používá čítač pro sledování aktuální pozici v programu, čímž se zajistí řízení toku programu. Během while cyklu se podle operačního kódu interpretuje jednotlivá instrukce a čítač se po každé iteraci inkrementuje (pokud nedošlo ke skoku na navěští). Pro ukládání hodnot proměnných jsem použil slovník, kde klíč byl název proměnné a hodnota aktuální hodnota proměnné. Slovník umožnil při práci s proměnnou kontrolovat, zda proměnná byla definovaná/inicializovaná. Jelikož je jazyk IPPcode21 dynamický typovaný, bylo také potřeba sledovat aktuální typ proměnné a kvůli tomu byl použit další slovník, který obsahoval typy proměnných uložené jako řetěz. Funkce `interpret()` vrací při úspěšné interpretaci programu hodnotu 0, nebo odpovídající chybový kód při selhání.