

Implementační dokumentace k 1. úloze do IPP 2021/2022

Jméno a příjmení: Tadeáš Kachyňa

Login: xkachy00

V rámci řešení mého projektu jsem se rozhodl pro částečné použití objektově orientovaného programování, což znamená, že jsem si při řešení pomohl vytvořením třídy, ale nedodržoval jsem striktně všechna paradigmatu a zásady OOP.

Stručný popis programu

Celý můj kod zastřešuje jeden velký cyklus while, který postupně načítá ze standardního vstupu jednotlivé instrukce a ty následně zpracovává. V tomto cyklu se nejprve odstraní z aktuálně načteného řádku komentáře, případně se načte další řádek, pokud je prázdný. Tenhle proces pokračuje dokud parser nenarazí na hlavičku "IPPCODE22". V případě nalezení čehokoliv jiného je toto považováno za chybnou hlavičku a je vyvolána chyba s návratovou hodnotou 21. Po nalezení hlavičky je vydáno povolení k parsování kódu. Nejprve se řádek rozdělí pomocí mezer a načte se do pole. Protože jsou instrukce case-insensitive, tak jsou převedeny na upper-case. Následně se zkontroluje, jestli sedí správný počet argumentů pro aktuálně zpracovávanou instrukci. Jsou převedeny problémové znaky, které by mohly v budoucnu představovat nebezpečí v XML souboru, kde jsou vyhrazené jako speciální znaky. Po kontrole je vytvořena nová instance třídy Instruction do které je nahrán operační kód instrukce a její argumenty. Nyní už zbývá jen zkontrolovat, zda-li každá instrukce obsahuje správné typy argumentů. O to se stará rozsáhlý switch-case. Instrukční sadu instrukcí jsem si rozdělil do 9 polí. Instrukce v každém poli spojuje stejný počet a typ argumentů. Podle těchto skupin se switch-case rozhoduje, který "case" bude vykonán. Zde již používám metody vytvořené v již dříve zmíněné třídě Instruction. K tisknutí XML nepoužívám žádnou knihovnu. Postupně se vystikne na standardní výstup začátek instrukce, její operační kód a pořadí, následně její argumenty a kód znáčí konec instrukce. Pokud se kdekoliv v tomto úseku vyskytne chyba, je vyvolána chyba s hodnotou 23. Jediný případ kdy je vyvolána chyba s hodnotou 22, pokud není operační kód instrukce znám.

Popis třídy a jednotlivých funkcí

Třída Instruction

Metody: print_instruction, print_arg[1-3], get_type, print_instruction_end

Tato třída je jediným náznakem OOP v mém programu. Vytvořil jsem si ji hlavně z důvodu lepší manipulace s instrukcí a jejími argumenty. Při vytvoření nové instance se zavolá konstruktor, který nahraje operační kód instrukce a její argumenty do proměnných. Třída obsahuje 6 metod - pro tisk začátku a konce instrukce, tisk argumentů instrukce a pro zjištění typu symbolu v argumentech.

Funkce cut_string

Vstupní parametry: \$splitted

Vracejí se hodnota: \$splitted_new

Tato funkce se stará o odstranění komentářů z jednotlivě načtených řádků. Vstupním parametrem je aktuálně načtený řádek a výstupem je již řádek zbavených komentářů.

Funkce check_inst_args

Vstupní parametry: \$splitted

Funkce se stará o kontrolu správného počtu argumentů instrukce. Vstupem je aktuálně načtený řádek. Pokud je počet argumentů chybný, hlásí program chybu a následně se ukončí. V opačném případě se nic neděje.

Funkce regex_match

Vstupní parametry: \$type

Vracejí se hodnota: regulární výraz

Funkce se stará o výběr správného regulárního výrazu, který následně vrací a předává knihovní funkci preg_match.