IPP Projekt 1 2019

Parse.php / Analyzátor kódu v IPPcode19

Řešení /

Implementaci jsem pro vyšší přehlednost rozdělil do čtyř souborů:

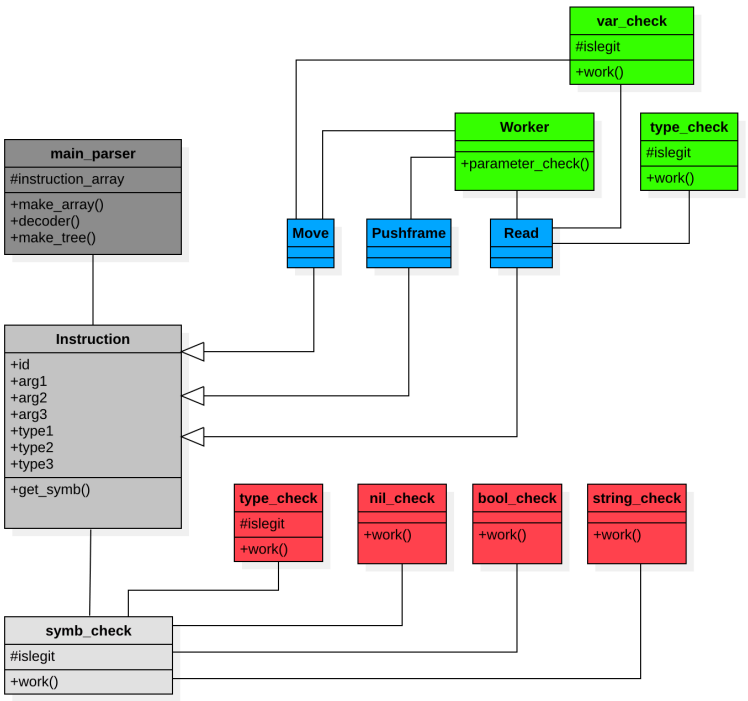
parse.php	hlavní tělo programu | nápověda | argumenty programu
aux.php	dekodér instrukcí | xml generator
instructions.php	generátor instrukcí
arguments.php	lexikální a syntaktická kontrola

Načítám řádek po řádku ze vstupu a zbavuji se bílých míst, prázdných řádků, komentářů a převádím všechny písmena na velká. Jako další krok kontroluji hlavičku zdrojového souboru funkcí Loader, pokud zde nedojde k chybě 21, navracím všechny ostatní řádky příkazem `yield` a vytvářím z nich pole které dekóduji zapomocí konstrukce `case`,která pro každou instrukci vytvoří instanci třídy podle typu instrukce, nebo vrací návratovou hodnotu 22 v případě, že se instrukci nepodaří dekódovat. Při psaní tříd instrukcí jsem využil dědičnosti, všechny instrukční třídy jsou potomky třídy `Instruction`, která je definována jako sedm proměnných pro uložení operandů instrukce a funkcí `get_symb` pro kontrolu případných symbolů. Každý z potomků pak obsahuje akce korespondující s počtem jeho operandů. Pokud jsou všechny instrukce vygenerovány bez chyb, dojde k jejich vygenerování do XML za pomocí třídy DOMDocument a rozšíření The Simple XML.

Rozšíření / STATP

Ke sběru statistik dochází jednak při odstraňování komentářů a dále při generování XML stromu, kde si značím počet vygenerovaných instrukcí, návěští a skoků.

Diagram tříd /



analogicky s třídou každé instrukce