Implementační dokumentace k 1. úloze do IPP 2018/2019
Jméno a příjmení: Jakub Gajdošík
Login: xgajdo24

Popis implementace programu

Program může být spuštěn s flagem –help, kdy se na STDOUT oběví popis a nápověda k programu, ale samotný program se nespustí.

Abych zajistil, že program bude převádět kód z jazyka IPPcode2019 efektivně, rozdělil jsem si program na 2 hlavní funkce. Funkce Input() vždy první zkontroluje hlavičku a postupně naformátuje všechny řádky kódu IPPcode2019, aby byli bez zbytečných prázdných znaků. Poté je jednu po druhé pomocí příkazu yield předává funkci to_XML(). Tato funkce převede řádky na instrukční objekty, jež postupně připojuje do XML stromu, který bude obsahovat finální verzi kódu.

Samotné instrukce jsem rozdělil do 4 tříd které se liší počtem argumentů. Během vytváření této třídy je zapotřebí volat funkce ověřující správnost instrukce, zda instrukce existuje, nebo má správný počet argumentů a nakonec se kontroluje, jestli odpovídají typy argumentů. Existenci instrukce a jestli má správný počet argumentů řeším pomocí stavových automatů, a to jestli odpovídají typy argumentů k dané instrukce, kontroluji pomocí regulárních výrazů. Pro přehlednost mám rozdělený typ arugemntů a samotnou hodnotu argumentu, kde v některých případech musím hodnotu argumentu oseknout aby to odpovídalo zadání a to také provádím pomocí regulárních výrazů.

Instrukce musely být očíslovány podle pořadí v jakém byly zadány a nejefektivnější řešení mi přišlo, vytvořit pomocnou funkci ve, které se nachází static proměnná, která si při každém zavolání pamatuje hodnutu, kdy byla posledně zavolána. Poté co mám objekt s instrukcí a argumenty zkontrolovaný přidám ho do stromu ve formátu XML. Toto opakuju dokud jsou načítány další řádky IPPcode2019 a poté se celý strom instrukcí uloží a vypíše na STDOUT.