Dokumentace úlohy CLS: C++ Classes do IPP 2015/2016
Jméno a příjmení: Radovan Sroka
Login: xsroka00

## Úvod

Tato dokumentace pojednává o řešení prvního projektu do předmětu IPP. Jedná se o skript v jazyce PHP ve verzi 5.3.3. Účelem skriptu je analýza C++ hlavičkových súborov. Skript analyzuje triedy jazyka C++, ich dedičnost, prípadne konflikty.

## Zpracování parametrů

K zpracování parametrů jsem využil standardní funkci jazyka PHP, `getopt()`. Tato funkce dokáže zpracovat zkrácené i dlouhé parametry programu. Tyto parametry se následně uloží do asociativního pole, kde klíčem je zadaný parametr a hodnotou prvku je zadaná hodnota parametru. Problémem také bylo načítání krátkých parametrů, protože funkce `getopt()` dokáže kombinovat krátké parametry. Načtené parametry následně vrátím v poli $options.

## Struktura

Skript jako takový je jedna velká třída, která se stará téměř o všechno. Jmenuje se `CLSParser`. Okrem toho se v skriptu nachází ježte jiné třídy, `CPPClass`, `ClassMethod`, `ClassAttr`…
Hned jako se `CLSParser` vytvoří konstruktorem, který se stará o parametry začne se vykonávat postupnost metod jako `readInput()` (přečte vstup se souboru nebo standardního vstupu), `parse()` (rozparsuje vstup a naplní objekty informacemi)a nakonec `gen_xml()`, který bude z vyzbíraných dat generovat výstupní xml.

## Parsování

Parsování je zabezpečeno rekuzívnim sestupem z hora dolu jednoduchou mnou definovanou gramatikou. Zadáni zcela vylučuje různé komentáře, makra nebo jiný kod okrem třídy, teda tato fakta velkým způsobem zjednodušují gramatiku avšak aj bez toho je gramatika celkem komplikována. Počas parsování se parsovaná data postupne ukladají do struktur které jich popisují. Parser v každém případu očekáva validní vstup, pokud tak není, vrátí chybu.

## Dědění

Mechanizmus dědění je zcela rekurzívní. Hlavním principem je to že se přehledává pole tříd a hledá se taková, která dědí aspoň od jednoho, když se taková najde můžeme započat rekurzi. Teda ak máme nějakou třídu hledáme z množiny všech tříd od kterých dědí a tento postup aplikujeme na všechny třídy co najdeme. Vždy cestou spět kopíruji všechny atributy a metody podle daných pravidel.

## Výstup

Výstup skriptu je ve formátu xml. Na generování xml struktury jsem použil knihovnu `xmlwriter`, která je podle mne velmi jednoduchá, dostačující a lehce se používá. Navyše je velmi vhodná pro rekurzívní algoritmus. V jediném případu jsem použil aj knihovnu `simplexml` a aj to kvůli metode `xpath()` a to tak, že jsem výstup `xmlwriter` načítal do `simplexml` objektu aby se v něm dalo vyhledávat.

## Závěr

Projekt jsem vyvíjel na referenčním školním serveru `merlin`, takže jsem měl zaručenou funkčnost mého řešení a nemusel jsem hledat jak nainstalovat starší verzi interpretu PHP na svůj počítač. K testování jsem využil zveřejněné ukázkové testy, které jsem upravil na automatické porovnání XML souborů a doplnil jsem je o další testy. A okrem toho jsem zavedl continues integration testovaní na vzdáleném virtuálnem serveru.