

Úkolem tohoto projektu bylo napsat skript, který načte soubor s popisem tříd jazyka C++, a vygeneruje strom dědičnosti, či detailní popis členů těchto tříd.

## Parametry

Pro načítání přepínačů jsem využil funkci `getopt()`, protože poskytuje vcelku jednoduché řešení pro krátké i dlouhé přepínače. Po načtení zjistím, jestli nesou chybné, a uložím jejich hodnoty do příslušných proměnných pro pozdější využití.

## Vstup

Skript dokáže zpracovávat standardní vstup, nebo soubory, jejichž cesta je zadána pomocí přepínače `--input`, respektive `-i`. Cesta může být jak relativní, tak absolutní. Soubor je celý načten do proměnné, a poté ihned uzavřen, a dále se nepoužívá.

## Zpracování vstupu

Pro zpracování vstupu jsem navrhnul několik regulárních výrazů. První z nich najde třídy, a uloží jejich jména, část s dědičností, a tělo třídy. Poté, se prochází nálezy, kontroluje se správnost zápisu, a třídy se ukládají do připravené struktury. Datová struktura je navržena jako asociativní pole tříd, kde jména načtených tříd jsou klíče. Pro zjednodušení jsou tyto jména také uloženy jako členy třídy. Po uložení jmen se prochází dědičností a kontroluje se existence rodičů. Po úspěšné kontrole se rodiče ukládají opět do asociativního pole, které je členem dané třídy. Do tohoto pole se ukládá přístupnost, a jako klíč je bráno opět jméno rodiče. Dalšími členy třídy struktury jsou asociativní pole (klíče jsou opět jména) atributů a pole metod. Jako první do těchto polí načteme členy tříd rodičů, v závislosti na přístupnosti těchto členů a dědičnosti. Po vyřešení dědičnosti se přistoupí ke zpracování samotného těla vstupní třídy. Tělo se pomocí regulárních výrazů rozdělí na bloky podle přístupnosti a ty dále na pole hlaviček tříd, a pole atributů. Tato pole se následně projdou, zkontroluje se syntaxe členů a uloží se do příslušných polí třídy struktury. Pro ukládání atributů a metod jsem opět využil asociativní pole tříd, kdy do třídy pro atributy se zapisuje jméno atributu, přístupnost, datový typ, jméno rodiče, pokud je zděděn a informace o tom, zda je atribut statický nebo virtuální. Třídy pro metody obsahují stejné informace, plus asociativní pole argumentů (jméno => datový typ) a informaci o čisté virtualitě metody. Celá tato část by mohla být ve vlastní funkci, ale jelikož by se použila jen jednou, nezabýval jsem se tím.

## Výstup

Pro sestavení výstupu jsem využil knihovnu XMLWriter, která je součástí PHP a je jednoduchá na používání. Výstup se postupně vypisuje buď na standardní výstup, nebo do souboru zadaného jako parametr přepínače `--output`, respektive `-o`. Tento projekt měl dvě podúlohy, a to a) tisk stromu dědičností nebo b) tisk podrobností třídy zadané parametrem přepínače `--details` nebo `-d`. pro obě podúlohy jsem vytvořil vlastní proceduru `outputA()` a `outputB()`. Při nezadání parametru přepínače `details` se procedura `outputB()` zavolá nad všemi třídami.

## Tisk stromu dědičnosti

Strom dědičnosti je struktura, na jejíž nejvyšší pozici jsou třídy bez rodičů, a pod nimi jsou jejich potomci. Procedura tedy projde všechny třídy ve struktuře, a pomocí procedury `outputAClass()` nechá vytisknout ty, které mají prázdné pole rodičů. Tato procedura vytiskne XML element třídy, a projde všechny třídy uložené v již zmiňované struktuře tříd. Pokud zjistí, že jméno tisknuté třídy se nachází v poli dědičnosti jiné třídy, zavolá se nad ní opět metoda `outputAClass()`.

## Tisk detailního popisu třídy

Detailní popis třídy začíná elementem `class` a informací o jejím jméně a druhu. Uvnitř najdeme element `inheritance`, ve kterém je seznam rodičů, což jsem řešil prostým průchodem pole.

Dále se uvnitř elementu `class` nachází elementy atributů a metod třídy rozříděné podle přístupnosti. Jelikož se ve výstupu podle zadání nesměli nacházet prázdné elementy přístupnosti ani elementy členů, musel jsem zjistit, zda třída má, a které druhy členů. Po zjištění už nic nebrání správnému tisku výsledku. K tisku jednotlivých členů jsem opět napsal procedury pro ulehčení, které nedělají nic jiného, než že tisknou data ze struktury.