

Cíl / zadání

Zadáním projektu bylo vytvořit skript v jazyce PHP v5.3.3, který analyzuje vstupní soubor obsahující výšeč kódu jazyka C++ se zjednodušenou syntaxí a na výstupní soubor vypíše jednotlivé třídy a jejich details – proměnné, metody, druh. Cílem projektu bylo vyzkoušet si základy objektově orientovaného programování v jazyce PHP – návrh, implementace.

Návrh

Objektový návrh projektu zahrnuje 6 tříd a 3 globální funkce, které jsou postupně využity k chodu skriptu v těchto fázích:

Zpracování parametrů

Vstupní parametry skriptu jsou zpracovány třídou `/params/`, která metodou `/get_params/` načte parametry z terminálu a metodou `/check_params/` zkontroluje jejich správnost – nedovolené kombinace, počet argumentů, popřípadě vypíše nápovědu funkcí `/printHelp/`, popřípadě je skript kdykoliv při zjištění chyby ukončen voláním funkce `/printError/`, která na standardní chybový výstup vypíše chybu a ukončí program s odpovídající návratovou hodnotou.

Kontrola vstupního a výstupního souboru

Kontrola vstupního a výstupního souboru je zajištěna třídou `/fileHandling/` a jejími metodami `/read_input_file/` a `/check_output_file/`. V případě zjištění chyby – např. omezení práv přístupu k souboru – se ukončí program s příslušnou návratovou hodnotou již zmíněnou funkcí `/printError/`. Pokud vše proběhne bez problému, je vstupní soubor načten jako string do proměnné, se kterou se dále pracuje.

Zpracování vstupních dat

Vstupní soubor je rozdělen podle jednotlivých tříd, jsou vytvořeny instance třídy `/classData/` pro každou třídu a do těchto instancí je uložena příslušná část zdrojového souboru odpovídající jejich definici. Tyto data jsou poté statickou třídní funkcí pro každou instanci třídy `/classData/` rozparsována a získají se informace o třídě – jméno, předci/rodiče, public/private/protected části definice. Dále je pak nad každou instancí této třídy volána třídní funkce `/get_class_kind/` pro získání typu abstract/concrete a `/get_class_children/` pro získání potomků třídy. Tyto data jsou později využita při výpisu elementů do XML souboru.

Data proměnných a metod

Data měla být zpracována a uložena do polí ke každé třídě, každá proměnná reprezentovaná instancí třídy `/variables/` a každá metoda instancí třídy `/methods/`. Bohužel k této implementaci jsem se již dostal pouze pro částečně pro zpracování proměnných a již ne metod. Objevil jsem totiž chybu v objektovém návrhu a celý skript už jsem nestíhal přepsat tak, aby bylo možné s těmito objekty vhodně pracovat.

Výstupní data

Jsou vypsána ve formátu XML a odsazena dle parametru `-pretty=xml=k`. Je využita rekurzivní funkce `/write_element/`, která načítá data z instancí objektů tříd a vypisuje do výstupního XML souboru.

Poznámky k implementaci

Bohužel jsem na počátku neměl vhodný objektový návrh a musel jsem ho předělávat později, a tedy už mi nezbyl čas na úplnou implementaci veškeré funkcionality. Ve skriptu chybí výpis detailů všech tříd a vyhledávání podle XPathu.

Závěr/zhodnocení

Implementace projektu bohužel nebyla plně dokončena, zde je vidět, jak důležitý je správný objektový návrh na počátku projektu. Paradoxně cíl projektu byl splněn výborně – naučil jsem se programovat v jazyce PHP. Nejtěžší část projektu nebyla samotná práce v jazyce PHP, ale díky mé vstupní neznalosti jazyka C++ to bylo právě hledání pravidel definice tříd a jejich vlastností v jazyce C++.