

Dokumentace úlohy CLS: C++ classes v PHP do IPP 2016/2017
Jméno a příjmení: Lukáš Drahník
Login: xdrahn00

Dokumentace úlohy CLS: C++ classes v PHP do IPP 2016/2017
Jméno a příjmení: Lukáš Drahník
Login: xdrahn00

1. Úvod do projektu

Cílem projektu bylo vytvořit skript pro analýzu dědičnosti mezi třídami popsanými zjednodušenou syntaxí pro soubory programovacího jazyka C++11.

1.1. Obsah repozitáře

Repozitář obsahuje zadání, zdrojový kód, testy, dokumentaci, soubor rozšíření s uvedenými zapnutými rozšířeními a skripty pro vytvoření a otestování potřebného archivu k odevzdání do informačního systému na FIT v Brně - WIS.

1.2. Požadavky na program

Projekt vyžaduje k běhu PHP verzi 5.6. Kompatibilita s jinými verzemi nebyla vyžadována a tudíž není zajištěna.

1.3. Readme

Obsahuje základní příkazy včetně potřebných příkazů pro zprovoznění repozitáře do spustitelného stavu.

1.4. Vývoj projektu

Vývoj projektu byl provedený postupných navrhováním a implementováním konkrétních modulů dle zadání, které bylo nutné testovat odděleně. K vývoji byl použit distribuovaný systém správy verzí, Git.

2. Rozdělení na moduly

Logickými celky v programu jsou CLS argument parser sloužící pro zpracování vstupních argumentů, CPP parser + CPP struktury pro převod zdrojového C++ kódu do sebnavazujících objektů, CLS parser pro zpracování vytvořených objektů a převod do XML a XML element objekt sloužící jako základní struktura pro generování XML.

2.1. CLS argument parser

Modul starající se o načítání vstupních parametrů pomocí metody getopt, parametry jsou v instanci uloženy po spuštění metody run a jsou zde implementovány metody isOptionSet a getOptionValue pro usnadnění. Na modul navazuje zpracování vstupních tříd pomocí CPP parseru, který sám o sobě potřebuje znát mód (standard nebo details) kvůli optimalizacím (dobrým příkladem je hledání konfliktů, kdy se kupříkladu nakládání s nimi bez zapnutého option --conflicts ignoruje).

2.2 CPP struktury a CPP parser

Modul řeší zpracování C++ kódu ze vstupu a jeho následné převedení na struktury provázaných objektů. Výstupem modulu jsou C++ objekty například typu třída, metoda, atributy u metody. V rámci modulu se také dohlíží na patřičnou sémantickou a syntaktickou správnost zápisu programu v jazyce C++ a vyvozování patřičných výjimek a návratových hodnot.

2.3. CLS parser

Základním kamenem modulu je používání CPP parseru a tedy navázání a práce s objekty z jazyka C++. V modulu je několikrát použitý rekurzivní sestup, při výpisu na patřičný výstup tak i při generování struktury z objektů typu XMLElement o kterém bude více zmíněno níže.

2.4. XML:

Pro zápis do souborů byl použit objekt XMLWriter, ke kterému byl doprogramován objekt XMLElement zajišťující jasné předávání potřebných dat k výpisu v XML mezi metodami bez ztráty informací. Objekt zajišťuje název a párovost elementu, instance potomků a potřebné zachování struktury, tedy pole instancí pod elementů – instance jsou taktéž typu XMLElementů.

3. Propojení modulů

Moduly jsou navrženy jako jednotlivé celky. Tato struktura byla zvolena vzhledem k vývoji a možnosti otestovat již hotové moduly. Provázanost kupříkladu rozdělení na jmené bloky řeší balíčkovací systém pro php, Composer. Po stažení projektu je nutné provést příkaz composer dump-autoload, který vytvoří linky na soubory specifikované pomocí složek v souboru composer.json a vytvoří složku vendor uvedenou v .gitignore, kde se nachází i veškeré soubory potřebné pro autoload. Ve výsledku to tedy znamená vyhnout se volání require, require_once.

4. Otestování modulů

K otestování byly použity testy CLS supplementary tests distribuované společně se zadáním rozšířené o obtížnější testy. Také je v repozitáři umístěn skript Is it ok, který skrze bash script testuje validitu vygenerovaného archivu k odevzdání.