

Analýza zadání

Cílem tohoto projektu je napsat skript, který provede analýzu dědičnosti mezi třídami definovanými ve vstupním souboru. Výstupem skriptu je potom strom dědičnosti, nebo vypsané detaily o některé z definovaných tříd.

Zpracování argumentů

Všechny argumenty jsou zpracovány a jejich hodnoty uloženy do asociativního pole `params`. Kontrola validnosti spočívá v ověření unikátnosti každého parametru, pro parametr vypisující náповědu i v kontrole počtu parametrů a v případě zadání přepínače *conflicts* je třeba ověřit, že byl zadán i přepínač *details*.

Zpracování vstupního souboru

Nejprve se projde vstupní soubor znak po znaku a vstup se rozdělí na pole trojrozměrných polí, která obsahují název třídy, údaje o děděných třídách a členy této třídy. Dále jsou zpracovány členy a rozděleny na jednotlivé atributy a metody. U každého členu jsou uloženy vlastnosti jako název, typ, zda se jedná o atribut, či metodu, modifikátor viditelnosti a další užitečné informace.

Kontrola validnosti vstupního souboru

K tomuto účelu slouží funkce `check_if_correct`, která kontroluje, zda nedochází k dědění od nedeklarované třídy, zda má každá deklarovaná třída i svou definici a také jestli nedochází k násobnému dědění od jedné třídy jinou třídou.

Dědění členů

Pro každou třídu, která dědí od jiné třídy, je provedena kopie atributů ze třídy děděné. Pokud se dědí atributy s modifikátory viditelnosti *private*, ve zděděné třídě nebudou přístupné, viditelnost zděděného atributu je tedy přepsána na *NA*. Kontroluje se, zda se nejedná o *pure virtual* metodu, v takovém případě zůstává modifikátor viditelnosti stejný. Po zdědění členů se provede kontrola přetěžování metod a atributů a redeklarace členů. Pokud dochází k redeklaraci, je zděděný člen smazán.

Nastavení abstraktnosti

Ve funkci `set_if_abstract` je projit záznam o každé třídě a všechny její členy, pokud obsahuje alespoň jednu *pure virtual* metodu, je abstraktní, jinak je konkrétní.

Kontrola konfliktů

Nejprve je pro všechna použití operátoru *using* zkontrolováno, zda používaný člen existuje a pokud ano, jsou odstraněny záznamy se všemi členy stejného jména, které jsou zděděny z jiných tříd a následně i záznam s operátorem *using*. Dále je provedena kontrola konfliktů. Pokud je zadán přepínač

conflicts, jsou konfliktní členy uloženy do pole `found_conflicts` a smazány jejich záznamy ze třídy, ve které ke konfliktu došlo. Pokud přepínač nebyl zadán a dojde ke konfliktu, skript je ukončen s návratovou hodnotou 21.

Vypsání výstupu

Existují dva přístupy k vypsání výsledného výstupu na základě toho, zda byl zadán přepínač *details*. Pokud byl zadán a požadují se údaje o všech třídách, volá se funkce `print_class_details` pro výpis jedné třídy opakovaně pro každou z nich. Pokud se požadují detaily pouze o jedné třídě, zavolá se tato funkce pouze jednou. Z této funkce se dále volají funkce na výpis konfliktních členů a funkce na výpis detailů o členech s určitou viditelností, přičemž se vždy požadovaná viditelnost předává jako parametr.

V případě výpisu bez přepínače *details* je po vypsání hlavičky pro každou třídu volána funkce `check_if_root`, která vrací, zda daná třída od nikoho nedědí a zda od této třídy dědí třída jiná. Pokud jsou obě podmínky splněny, volá se rekurzivně funkce `print_inheriting_classes`, která vypisuje údaje o třídách, které dědí ze třídy zadané na vstupu. Invariantní proměnné jsou řešeny pomocí globálních proměnných.

Odsazování ve výstupních datech zařizuje funkce `indent_line`, která přidá požadovaný počet mezer na konec dosud vypsáných dat.

Vyhledání pomocí Xpath

Pokud byl zadán parametr *search*, provede se pomocí funkcí z knihoven *lxml* a *minidom* vyhledání zadaného výrazu. Pokud je dosavadní výstup prázdný, případně obsahuje pouze xml hlavičku, vrací se nezměněný vstup. Jinak je odříznuta hlavička a vstup je převeden na stromovou reprezentaci a následně je provedeno vyhledání. Na základě typu vrácených dat je provedeno jejich vypsání a přidání k xml hlavičce.

Závěr

Jako poslední operace je provedeno otevření výstupního souboru, případně navázání na standardní výstup, výsledná data jsou vypsána na výstup a celý skript končí s návratovou hodnotou 0.