

Dokumentácia k projektu

Analýza zadania

Cieľom projektu bolo vytvoriť skript pre analýzu dedičnosti medzi triedami zapísanými zjednodušenou syntaxou programovacieho jazyka C++11. Skript zozbiera všetky informácie zo vstupného súboru a formou XML výpisov odpovedá na požiadavky používateľa skriptu.

Postup riešenia

Po spustení skriptu sa ako prvé spracujú používateľské argumenty. Pomocou získaných dát od používateľa sa nastaví dôležité premenné, ktoré ovplyvňujú výstup skriptu.

V prípade korektného zadania argumentov program ďalej pokračuje analýzou vstupného súboru a napĺňaním dátových štruktúr záujmovými dátami.

Ďalšou fázou skriptu je spracovanie nazbieraných dát. V tejto časti sa buduje strom dedičnosti a dopĺňajú informácie o triedach, ktoré vyplývajú z jej vzťahov medzi inými triedami.

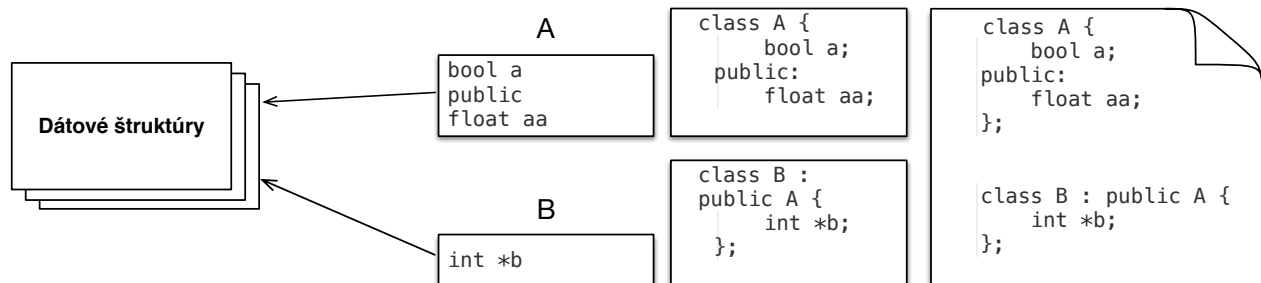
Poslednou fázou programu je vytvorenie výstupného XML výpisu. Táto časť sa venuje správne formátovaniu XML tak, ako si to vyžaduje zadanie.

Spracovanie argumentov

Na spracovanie používateľských argumentov je použitá knižnica *argparse*. Jej funkčnosť je rozšírená o rôzne nadštandardné funkcie, ktoré vo svojom základe neponúka. Napríklad kontrola opakovaného zadania rovnakého argumentu, alebo úprava hodnoty návratových kódov pri chybách.

Analýza vstupného súboru

Pred samotnou analýzou súboru je C++ vstup rozdelený na zoznam tried. Následne prebieha analýza nad každou triedou. Z hlavičky je vyčítaný názov a zoznam tried s ktorými je vo vzťahu. Potom skript rozdelí telo každej triedy na zoznam príkazov na základe regulárneho výrazu. Následne skript analyzuje každý príkaz a dáta ukladá do pripravených štruktúr. (popis nižšie)



Dátové štruktúry

Dáta ktoré program počas analýzy vstupného súboru nazbiera, sú uložené do objektov *ClassObject* a *Method*. Každý z týchto objektov má atribúty a metódy, ktoré definujú množinu uchovávaných dát a operácie nad nimi. Na uloženie objektov, ktoré nepotrebujú definovať operácie nad jej dátami, skript používa technológiu *nametuples*.

Spracovanie dát

Po načítaní a správnom uložení dát musí program vykonať množstvo operácií nad nimi. Najprv skript vykoná všetky formuly *using* pre každú triedu.

Následne je spustený rekurzívny algoritmus, ktorý preskúma všetky triedy z ktorých dedí (priame aj nepriame dedičnosti). Na každej triede sa zavolá funkcia, ktorá fyzicky skopíruje atribúty a metódy, ktoré sú implementované v skúmanej triede. Táto funkcia má tiež na starosti dôkladné triedenie podľa spôsobu dedenia a "možnosti" a "nemožnosti" dedenia z danej triedy. Výsledkom je doplnený zoznam každej triedy o zdedené metódy a atribúty.

Ďalšou operáciou je skontrolovanie každej triedy, či neobsahuje *pure virtual* metódy. Ak áno je celá trieda nastavená ako virtuálna.

Ďalšou dôležitou úlohou je detekcia konfliktov, ktoré počas dedenia mohli nastať. Kontroluje sa rôznosť mien medzi metódami a atribútami. Zvyšné druhy konfliktov sú odhalené v priebehu vkladania elementov.

Generovanie XML

Pre generáciu XML program používa *minidom* z balíčka *xml.dom*. Pre zaistenie funkčnosti XPATH program využíva aj knižnicu *lxml*.

Rozšírenie CFL

Program implementuje aj rozšírenie CFL. Spôsob riešenia spočíva vo vytvorení zoznamu konfliktných členov v každej triede. V prípade, že sa vyskytne konflikt, je tento konfliktný člen uložený do tejto separátnej množiny a vymazaný z tried, ktoré boli zúčastnené na konflikte.