

Laboratorul 4.

Analiză Sintactică II.

Generarea de AST.

AST - Abstract Syntax Tree

AST-ul este o structură de date construită în memorie, de către compilator, ca o rafinare a arborelui de derivare (obținut doar prin rularea parser-ului).

Această structură nu are o formă standardizată (depinde de implementarea și cerințele fiecărui compilator și limbaj de programare), dar în general are următoarele caracteristici:

- Toate nodurile, chiar dacă corespund unor tipuri diferite de instrucțiuni, sunt implementate ca fiind clase ce extind aceeași clasă de bază.
- Nu conține noduri "dummy", sau "intermediare" (ex: pentru a parsa o constantă numerică, inițial se generează un arbore de forma `expr - test - sum - term - integer`).
- Dacă unele structuri de sintaxă sunt echivalente (ex: reprezentări diferite ale constantelor numerice, `if...then...endif` vs. `if...then...else NOP endif`), ele pot fi reprezentate în AST sub o formă unificată.
- Dacă unele structuri de sintaxă există doar ca indicații pentru parser (ex: parantezele din expresiile matematice, pentru stabilirea ordinii corecte a operațiilor), nodurile lor pot fi eliminate din AST, atâta timp cât ordinea corectă a operațiilor apare corect în arbore.

Exerciții:

Ne propunem ca în laboratorul de astăzi să construim și să afișăm AST-ul unui program CPLang. **Salvați fișierele laboratorului anterior sau copiați-le într-un proiect nou pentru a nu pierde rezolvarea!** Copiați fișierele din arhivă în proiectul laboratorului anterior și urmăriți TODO-urile din fișierele `ASTNode.java`, `ASTVisitor.java` și `Test.java`.

Definirea ierarhiei de clase pentru AST

Pornind de la clasa-părinte (`ASTNode`), extindeți cu clase specializate pentru fiecare tip de nod ce poate apărea în AST (pentru blocuri de instrucțiuni, pentru `if`, operații aritmetice, etc.).

- Asigurați-vă că salvați corespunzător copiii unui nod, luând în considerare numărul acestora
- Pentru definiții și apeluri de funcții puteți avea câmpuri de forma `LinkedList<ASTNode>` în care să salvați argumentele apărute între parantezele rotunde

Generarea AST-ului și afișarea sa

Completați în fișierul `Test.java` cele 2 instanțe de visitor:

- `astConstructionVisitor` = se va ocupa de construcția AST-ului, folosindu-se de arborele de derivare obținut în urma parsării (laboratorul trecut)
- `printVisitor` = va printa conținutul fiecărui nod, afișarea fiind făcută folosind indentarea cu tab-uri în funcție de nivelul pe care nodul este prezent

Afișând nodul-părinte al AST-ului, recursiv se va declanșa afișarea întreg arborelui.