

Laboratorul 3

Analiză Sintactică I

CPL

0 Setup

- Descărcați arhiva ce conține scheletul laboratorului.
- Scheletul are suport pentru Eclipse și IntelliJ sau poate fi compilat și rulat în linie de comandă folosind Makefile. Alegeți arhiva corespunzătoare setup-ului pe care ați decis să îl folosiți.

1 Intro

- Obiectivul laboratorului este ca să implementăm un analizor sintactic pentru limbajul **CPLang** introdus la laboratorul anterior.
- Înainte de parcurgerea exercițiilor, citiți cu atenție scheletul. Pentru fiecare exercițiu, găsiți în cod explicații și sugestii de implementare. Urmărește **TODOx** unde x este nr-ul exercițiului. Sursele de interes sunt **CPLANGParser.g4** și **Test**.

2 Parser pentru limbajul CPLANG

- Completați gramatica din **CPLangParser.g**, astfel încât să puteți parsă programe CPLang urmărind **TODO1**-urile din fișier.
- Pentru rezolvare, urmăriți **TODO1**-urile din **Test.java**.
- În metoda **main** se apelează metodă corespunzătoare task-ului în funcție de parametrul primit din linia de comandă. În cazul de față, parametrul este 1. Fișierul de intrare cu cod CPLang este manual.txt.

3 Extragerea de statistici din cod

- Implementați folosind **listeners** și urmărind **TODO2**-urile din **Test.java**, un pas de analiză ce afișează
 - numărul de definiții de variabile (atât la "global scope" cât și ca parametri formali ai unor funcții)
 - numărul de funcții definite.

- Pentru testare, rulați Test cu parametrul 2. Fisierul de intrare cu cod CPlang este manual.txt. Ca referință puteți folosi reference1-2.txt

Pentru următorul cod:

```
1 Int a;
2 Int b = 1;
3 Bool c = a == b;
4 Int increment(Int a) {a + 1};
5 print_out(increment(a));
```

Se va afișa:

```
1 Definitii de variabile: 4
2 Definitii de functii: 1
```

4 Evaluador de expresii matematice

- Implementați folosind **visitors**, un interpretor de expresii matematice simple. Expresiile vor fi separate de ";" și vor conține doar literalii numerici întregi pozitivi, paranteze, și operatorii "+-*/*".

!!! Va trebui să țineți cont de ordinea obișnuită a operațiilor.

Exemplu de input:

```
1 (3*2+6)/2;
2 8+8*2;
3 1+1/2;
```

Exemplu de output:

```
1 6
2 24
3 1
```

- Pentru acest task, urmăriți **TODO3**-urile din **CPLangParser.g4** și **Test.java**
- Pentru testare, rulați Test cu parametrul 3. Fisierul de input este input3.txt, iar ca referință puteți folosi reference3.txt.