Document Readme pentru aplicatia Interpreter

Pentru a aduce aplicatia in starea care este ea acum, am folosit fisierele de la ultimul laborator de compilatoare, unde aveam deja implementat aflarea rezultatului si afisarea token-ilor, cu niste mici erori pe care le-am rezolvat. Urmatorul pas a fost sa caut cum pot implementa fereastra turtle in interiorul aplicatiei, la apasarea butonului de desenare, nedeschizandu-se o alta fereastra. Acest pas a fost usor, in documentatia oficiala pentru Turtle am gasit toate informatiile de care aveam nevoie.

Apoi am decis sa tratez crearea arborelui de derivatie. Cea mai usoara metoda parea sa fie folosirea unei stive si a unui arbore binar. Pentru a avea acces la implementari ale acestor structure de date am folosit libraria Pythonds. Am incercat sa creez arborele de derivatie in parser, acolo unde aveam acces la toti tokenii, dar din pacate, singurul caz care merge tratat si pentru care afiseaza un arbore bun este cel in care arborele de derivatie rezultat este balansat, iar expresia initiala contine paranteze care delimiteaza operatiile. De asemenea, am incercat sa folosesc deja variabilele si functiile care se aflau deja in fisierul interpreter.py pentru a creea arborele, dar nu m-au ajutat, in final folosind ideea originala de a crea un arbore binar nou. Desenul se realizeaza in maniera postordine, folosind o functie recursiva de desenare, in care desenarea cercurilor si a caracterelor token se face la iesirea din fiecare nivel de recursivitate, deci postordine, libraria Pythonds oferind suport pentru afisarea parcurgerilor in maniera text, deci a fost usor sa verific daca rezultatul este bun.

La final, am implementat afisarea pe ecran si in fisier a productiilor rezultate in urma rezolvarii operatiilor facute de catre program, lucru ce a fost trivial, deoarece a trebuit numai sa salvez numai tokenii parcursi de clasa interpreter pe randuri si sa ii afisez pe ecran/in fisier.