



FUT Language

Crie Simulações de Partidas de Futebol com a FUT Language

Introdução

A FUT Language é uma linguagem de programação que permite criar simulações de partidas de futebol. Ela foi desenvolvida seguindo todos os padrões, desde a criação de sua EBNF até o desenvolvimento de um compilador em python, que realiza as análises léxicas, sintáticas e semânticas da entrada do programador, interpretando-a e exibindo o resultado ou dando erro.



Motivação

Paixão pelo Futebol

Desde pequeno sempre fui apaixonado por futebol e vários outros esportes. Assim, quando nos foi proposto a criação de uma nova linguagem de programação, decidi aliar o projeto com uma paixão antiga.

Simulação e Estatística

Sempre consumi muito conteúdo relacionado a análises de futebol as estatísticas de partidas. Dessa forma, senti a necessidade de criar uma linguagem que possibilitasse profissionais e amadores simular partidas.

Características

1

Sintaxe

A sintaxe da linguagem é inspirada em python e julia, com algumas alterações.

2

Estruturas

A linguagem possui declaração de variáveis, atribuição de valores, while loop, operadores aritméticos e lógicos, condicionais, funções e print.

3

Tipagem

A linguagem possui os tipos INTEGER e STRING e é necessário dizer o tipo ao declarar uma variável

Curiosidades da Linguagem



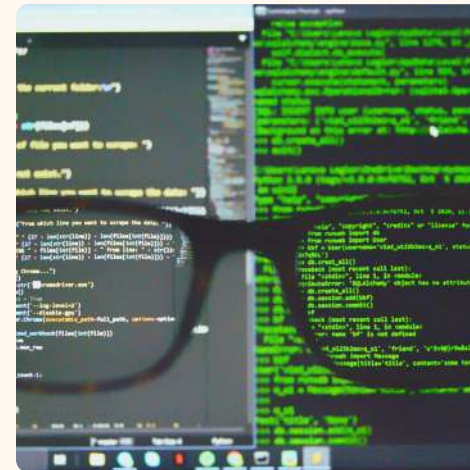
Decisão do tema

Antes de começar a desenvolver a linguagem já tinha decidido que queria fazer algo relacionado a esporte, mas até ter certeza do futebol pensei em várias opções, como basquete e vôlei



Sintaxe

Queria desenvolver algo que fosse fácil de ser utilizado, por isso misturei a sintaxe de python com alguns conceitos de julia e criei algo simples e funcional



Multiuso

A linguagem tem como objetivo principal ser utilizada para simular partidas de futebol. Entretanto, por conta de sua estrutura e sintaxe pode ser utilizada para diversas coisas, como uma linguagem tradicional

Exemplos

```
DECLARE TEAM1_SCORE AS INTEGER
DECLARE TEAM2_SCORE AS INTEGER
DECLARE POSSESSION AS INTEGER
DECLARE GAME_TIME AS INTEGER

FUNCTION increment_score(score AS INTEGER)
    score = score + 1
    RETURN score
ENDFUNCTION

TEAM1_SCORE = 0
TEAM2_SCORE = 0
POSSESSION = 1
GAME_TIME = 0
```

```
WHILE GAME_TIME < 90 DO
    GAME_TIME = GAME_TIME + 1

    IF POSSESSION == 1 THEN
        IF TEAM1_SCORE < 5 THEN
            TEAM1_SCORE = increment_score(TEAM1_SCORE)
            PRINT "Team 1 scores!"
        ENDIF
        POSSESSION = 2
    ELSE
        IF TEAM2_SCORE < 4 THEN
            TEAM2_SCORE = increment_score(TEAM2_SCORE)
            PRINT "Team 2 scores!"
        ENDIF
        POSSESSION = 1
    ENDIF
ENDWHILE
```