

Gerçek Zamanlı Sözdizimi Renklendirici: Lexer ve Parser Tabanlı Bir Swing Uygulaması

Giriş

Bu proje, Java programlama dili için gerçek zamanlı bir sözdizimi renklendirici (syntax highlighter) geliştirmeyi hedeflemektedir. Proje kapsamında, bir kod editörü içerisine yazılan Java kodları eş zamanlı olarak analiz edilerek sözdizimi öğeleri renklendirilmiş, hatalı ifadeler tespit edilip kullanıcıya gösterilmiştir. Bu işlem hem leksik (lexical) hem de sözdizimsel (syntactic) analiz adımlarını içerir ve tamamen kullanıcı arayüzü üzerinde gerçekleşir.

Amaç ve Hedefler

Projenin temel amacı:

- En az beş farklı token türünü tanıyan,
- Gerçek zamanlı çalışan,
- Kütüphane kullanmadan yazılmış bir renklendirici oluşturmak.

Buna ek olarak, proje sırasında kullanılan yöntemlerin belgelenmesi, uygulamanın herkese açık şekilde sunulması ve bir demo videosunun yayımlanması da hedeflenmiştir.

Kullanılan Yöntemler

Leksik Analiz (Lexer)

Proje içerisinde `Lexer.java` sınıfı, karakter girdilerini `Token` nesnelere dönüştürmekle sorumludur. Token tipleri şunlardır:

- KEYWORD (örneğin class, if, int)
- IDENTIFIER (değişken/metot isimleri)
- NUMBER (sayısal ifadeler)
- STRING (tırnak içinde metinler)
- SYMBOL (parantezler, noktalı virgül, vs.)

Lexer, manuel olarak yazılmış bir durum makinesi (state machine) ile çalışmaktadır.

Sözdizim Analizi (Parser)

Parser.java sınıfı, Token listesini alır ve Recursive Descent yöntemi ile parse eder. Top-Down yaklaşımı kullanılmıştır. Basitleştirilmiş bir Java grammar'ı tanımlanarak şunlar desteklenmiştir:

- Değişken tanımlamaları
- Metot tanımı ve çağırısı
- if-else blokları
- System.out.println ifadeleri

Her hata, ilgili satırda alt kısımda kullanıcıya bildirilmektedir.

Gerçek Zamanlı Renklendirme

Her kullanıcı girdisi sonrası StyledDocument üzerinden içerik tekrar analiz edilir. Token tiplerine göre renkler atanır:

| Token Türü | Renk |
|------------|---------|
| Keyword | Mavi |
| Identifier | Siyah |
| Number | Turuncu |
| String | Yeşil |
| Operator | Kırmızı |

Grafik Arayüz (GUI)

Swing tabanlı kullanıcı arayüzü aşağıdaki bileşenlerden oluşur:

- JTextPane: Kod yazma alanı
- StyledDocument: Renklendirme
- JLabel: Hata mesajı gösterimi

Her metin değişikliğinde hem lexer hem parser yeniden çalıştırılarak içerik değerlendirilir.

Örnek Kullanım

```
Class Main{  
    Public static void main(){  
        int a=4;  
        if(a<7)  
            System.out.println("a 7'den küçüktür");  
        else  
            System.out.println("a 7'den büyüktür");  
    }  
}
```

Proje Yapısı

```
javaSyntaxHighlighter/  
|  
├── Panel.java    # GUI ve renk mantığı  
├── Lexer.java    # Lexical Analyzer  
├── Parser.java    # Top-Down Parser  
├── Token.java    # Token sınıfı ve türleri  
└── Test.java     # Ana uygulama başlatıcısı
```

Demo Videosu

Projenin çalışmasını görmek için tanıtım videosunu izleyebilirsiniz:

 https://www.youtube.com/watch?v=jXF_bsAEGjY

Katkı ve İletişim

Bu proje bireysel olarak geliştirilmiştir. Kodlara GitHub üzerinden erişebilir ve katkıda bulunabilirsiniz.

Görüş, öneri ya da katkı talepleriniz için iletişime geçebilirsiniz.



E-posta: kgnmr16@gmail.com



GitHub: <https://github.com/KaganEM16/Java-Syntax-Highlighter>