

Java Gerek Zamanlı Sz dizimi Vurgulayıcı (Syntax Highlighter)

Bu proje, Java dilinde yazılmış bir **gerek zamanlı sz dizimi vurgulayıcı (syntax highlighter)** uygulamasıdır. Proje, kullanıcı tarafından yazılan kodu analiz ederek hem **lexical (szlk)** hem de **syntax (sz dizimi)** analizi yapar ve kullanıcıya yazdığı kodun anlık durumuna gre renkli geri bildirim sunar.

Projenin Amacı

- Gerek zamanlı szlk ve sz dizimi analizi yapmak
- Basitleştirilmiş bir C benzeri dilin gramerine gre kodu doėrulamak
- 5 farklı token trne gre vurgulama yapmak
- Syntax hatalarını kullanıcıya bildirmek

Kullanılan Dil ve Gramer

Proje, basitleştirilmiş bir **C benzeri dil** kullanır ve aőaėıdaki yapıları destekler:

- **Anahtar kelimeler (Keywords):** int, if, else, while
- **Operatrler:** +, -, *, /, =, ==, !=, <, >, <=, >=
- **Semboller:** (,), {, }, ;, ,
- **Tanımlayıcılar (Identifiers)** x, i, ...
- **Sayılar**

Lexical Analiz (Lexer)

Lexer sınıfı, kullanıcıdan alınan kodu stringini baőtan sona tarar ve her bir token iin bir Token nesnesi oluőturur. Token'lar őu trlere ayrılır:

- KEYWORD – Anahtar kelimeler (int, if, vb.)
- IDENTIFIER – Deėiőken isimleri
- NUMBER – Sayılar (pozitif tam sayılar)
- OPERATOR – Matematiksel ve karőılaőtırma operatrleri
- SYMBOL – Noktalı virgl, parantez, ssl parantez vb.

Sz dizimi Analizi (Parser)

Parser, **recursive-descent** yöntemiyle Lexer'dan alınan token listesini analiz eder.

Beklenen yapılar şunlardır:

- Bildirimler: `int x = 5;`
- Koşullu ifadeler: `if (x > 3) { ... }`
- Döngüler: `while (x < 10) { ... }`

Hatalı veya eksik sözdizimi durumunda kullanıcıya "Syntax error at: ...", doğru ise "No syntax errors." mesajı gösterilir.

Parser Fonksiyonlarının Çalışma Mantığı

◆ **matchType(String expectedType)**

- aldığı String'i o anki token tipiyle karşılaştırarak istenen token tipiyle eşleşip eşleşmediğini döndürür.

◆ **matchValue(String expectedValue)**

- aldığı String'i o anki token değeri(value) ile karşılaştırarak istenen değerle eşleşip eşleşmediğini döndürür.

◆ **parseFunc()**

- Kod stringinin bütünüyle incelendiği fonksiyondur.
- İlk iş keyword (int) ve identifier kontrolleriyle koda fonksiyon tanımıyla başlanmasını zorunlu kılar.
- `parseArgList()` ile opsiyonel argüman/argümanlar tanımını ayırıştırır.
- Asıl fonksiyon bloğuna inildiğinde `parseCompStmt()` çağrılır.

◆ **parseArgList()**

- Argüman listesi işler.
- Her keyword ile karşılaştığında tokenı geçmeden `parseArg()` çağrılır.
- devamında ',' ile karşılaşırsa birden çok argüman vardır. Hepsi tükenene kadar `parseArg()` ile tek tek ayırıştırır.

◆ **parseArg()**

- `matchType()` ile arka arkaya keyword ve identifier kontrolü yapar.

◆ **parseCompStmt()**

- Bir `{ ... }` bloğunu işler.

- İçeriği `parseStmtList()` fonksiyonu ile ayrıştırır.

◆ `parseStmtList()`

- Blok içindeki birden fazla ifadeyi sırayla işler.
- Her ifade için `parseStmt()` çağrılır.

◆ `parseStmt()`

- Genel ifadeleri tanır:
 - `if, while` → kontrol yapıları
 - `{` → blok ifadesi
 - `identifier` → atama ifadesi

◆ `parseExpr()`

- Aritmetik ve mantıksal ifadeleri işler.
- Operatör önceliğine göre non-terminal elementleri ifade eden alt fonksiyonlar çağrılır:
 - `parseTerm()`
 - `parseFactor()` → `IDENTIFIER / NUMBER` tokenlarıyla ilgilenir. Parsingin tamamlandığı noktadır.

Örnek Kod (Desteklenen Sözdizimi)

```
int main(){  
    int x = 5;  
    if (x > 0) {  
        x = x - 1;  
    } else {  
        x = 0;  
    }  
}
```

Vurgulama Şeması

Her token tipi için farklı renk atanmıştır:

Token Tr	Renk (RGB)
KEYWORD	Mavi (64, 64, 231)
IDENTIFIER	Yeřil (0, 128, 0)
NUMBER	Pembe (228, 71, 228)
OPERATOR	Kırmızı (200, 0, 0)
SYMBOL	Turuncu (254, 133, 76)

GUI

Highlighter sınıfı, bu projenin grafik kullanıcı arayzn (GUI) saėlayan bileřenidir. Java Swing kullanılarak geliştirilen bu sınıf, kullanıcıdan alınan kod stringini gerek zamanlı olarak analiz eder, sözdizimini denetler ve token trlerine göre renklendirerek grsel geri bildirim sunar.

Temel Özellikler

- **Gerek zamanlı kod analiz ve vurgulama** (Lexer ve Parser kullanarak)
- **Renkli sözdizimi gsterimi**
- **Hata mesajı durumu gstergesi (status bar)**
- **Debounce mekanizması ile gereksiz analizlerin önlenmesi**

Nasıl alışır?

1. Arayz Oluřturma

- JFrame içinde bir JTextPane (kod yazma alanı) ve bir JLabel (hata durumu için) oluşturulur.
- Kullanıcı yazdıka olay dinleyicileri (DocumentListener) deėiřiklikleri takip eder.

2. Debounce Mekanizması

- Her yazma olayından sonra 100ms beklenir (debounceTimer) ve applyHighlighting() fonksiyonu alıştırılır.
- Bu bekleme, ok sık analiz alıştırılmasını önleyerek program yükn azaltır.

3. Tokenize ve Vurgulama

- Lexer sınıfıyla oluşturulan her Token, tipine göre aşağıdaki renklere boyanır:
 - KEYWORD → Mavi (#4040E7)
 - IDENTIFIER → Yeşil (#008000)
 - NUMBER → Mor (#E447E4)
 - OPERATOR → Kırmızı (#C80000)
 - SYMBOL → Turuncu (#FE854C)

4. Sözdizimi Analizi

- Ekrandaki anlık söz dizisi parse edilir.
- Eğer bir hata varsa, `parser.getLastError()` ile alınarak alt kısımdaki status bar'da gösterilir.
- Hata yoksa "No syntax errors." mesajı gösterilir.

Teknik Detaylar

- **StyledDocument:** JTextPane üzerinde renklendirme için kullanılır.
- **SimpleAttributeSet:** Her token'a renk atamak için kullanılır.
- **Lexer:** Kullanıcının yazdığı metni dil kurallarına uygun şekilde Token nesnelerine kategorize ederek sırayla tek bir tokens listesinde saklar.
- **Parser:** Token dizisinin sözdizimsel olarak doğru olup olmadığını kontrol eder.

Bilinen Kısıtlama: Satır Sonları ve Vurgulama Hatası

- Highlighter sınıfı, satır atlama yapıldığı, yani `\n\r` türevi girdilerle karşılaştığında, renklendirmede birer karakter kayma yaşar. Yapılan testlerle Lexer ve Parser sınıflarının gereğince tokenizasyon ve ayrıştırma yapsa da Highlighter'ın renklendirmede indeks kayması yaşadığı tespit edilmiştir.
- Söz konusu renklendirme hatası tek satırlık, sadece boşluk içeren kodlarda yaşanmaz.
-

Yanlış Renklendirmeye Sebep Olan örnek Kod String'i

```
int main(){  
  
    int x = 5;  
  
    if (x > 0) {  
  
        x = x - 1;
```

```
} else {  
    x = 0;  
}  
}
```

Doğru Renklendirilen Örnek Kod String'i

```
int main(){int x = 5;if (x > 0) {x = x - 1;} else {x = 0;}}
```