**Gerçek Zamanlı Java Sözdizimi Vurgulayıcı ve Grafik Arayüz Uygulaması**

Kullandığımız kod editörleri (IDE), yazılım geliştirme süreçlerinde geliştiricilerin yazdığı kodu anlamasına ve hatalarını hızlı bir şekilde fark etmesine olanak tanır. Bu editörlerin en önemli özelliklerinden biri gerçek zamanlı sözdizimi vurgulamasıdır. Bu projede, java dilini kullanarak, grafik arayüzüne sahip, gramer tabanlı bir sözdizimi vurgulayıcı uygulaması gerçekleştirilmiştir.  
  
Uygulama; sözcük çözümleme (lexical analysis), yapı çözümleme (syntax analysis) ve gerçek zamanlı kullanıcı etkileşimi adımlarını birleştirerek, programlama dillerine özgü gramer yapılarının algılanması ve renklendirilmesini sağlamaktadır.

**Kullanılan Teknolojiler**

* **Programlama Dili:** Java
* **Arayüz:** Swing (javax.swing, javax.swing.text)
* **Regex Motoru:** java.util.regex

**Sözcük Çözümleme Yöntemi**

Bu projede **State Diagram & Program Implementation** yöntemi kullanılmıştır. Token tipleri için belirlenen regex desenleri, Java'da Pattern ve Matcher sınıflarıyla işlenmiş ve her desen doğrudan koda gömülerek programatik olarak eşleştirilmiştir. Her token tipi, ayrı bir desenle tanımlanmış ve LexicalAnalyzer.java sınıfı içinde işlenmiştir. Eşleşme öncelikleri manuel olarak belirlenmiş ve çakışmaları önlemek için yorumlar ve string ifadeler gibi özel durumlara öncelik verilmiştir.

**Yapı Çözümleme Yöntemi**

Bu projede **Top-Down Parsing** yaklaşımı ve özellikle **Recursive Descent Parsing** uygulanmıştır. Her gramer kuralı, SyntaxParser.java içerisinde ayrı bir metotla tanımlanmıştır. Bu sayede hem gramer okunabilirliği sağlanmış hem de kolay hata takibi yapılabilir hale getirilmiştir.

Parser, yazılan ifadeleri kontrol ederken tanımsız ya da eksik yapıları tespit ederek kullanıcıya hata mesajı göstermektedir. Bu hatalar GUI üzerinden anlık olarak kırmızı ve altı çizili biçimde sunulmaktadır.

**Token Türleri**

1. Anahtar Kelimeler (KEYWORD)
2. Veri Türleri (TYPE)
3. String Literaller (STRING)
4. Yorumlar (COMMENT)
5. Sayılar (NUMBER)
6. Operatörler (OPERATOR)
7. Ayırıcılar (DELIMITER)
8. Tanımlayıcılar (IDENTIFIER)
9. Sabit Değerler (LITERAL)
10. Annotation’lar (ANNOTATION)
11. Preprocessor Direktifleri (PREPROCESSOR)
12. Hatalı Tokenlar (ERROR)
13. Boşluklar (WHITESPACE)
14. Dosya Sonu (EOF)

Her token türü için TokenType.java sınıfında belirlenmiş sabitler ve HighlightManager.java içinde renk eşleştirmeleri tanımlanmıştır.

**Grafik Arayüz ve Özellikler**

Uygulamanın kullanıcı arayüzü Java Swing kullanılarak geliştirilmiştir. MainFrame.java ve EditorPanel.java sınıfları üzerinden oluşturulan pencere, kullanıcı tarafından yazılan kodu anlık olarak analiz ederek StyledDocument ile renklendirme yapmaktadır. Bu renklendirme sayesinde geliştiriciler kodlarını gönül rahatlığıyla yazabilmektedir

Öne çıkan özellikler şunlardır:

* Gerçek zamanlı renklendirme
* Karanlık tema desteği
* Her token türü için ayrı renk şeması
* Hatalı token'lar için kırmızı altı çizili gösterim
* Durum bilgisi gösteren bilgi çubuğu

Bu yapı sayesinde kullanıcı dostu bir editör deneyimi sağlanması amaçlanmıştır. Yazılan her karakterin ardından analiz işlemi tetiklenmektedir.

**Karşılaşılan Zorluklar**

Gerçek zamanlı işleme sırasında karşılaştığım zorluklar aşağıdaki gibidir:

* **Performans Sorunları:** Program çalıştırıldıktan sonra sürekli kullanıcı arayüzünde donmalar ve bilgisayara yoğun bir yüklenme söz konusu vardı. Çözüm olarak zaman gecikmeli işleme (rate limiting) uygulandı.
* **Regex Çakışmaları:** Bir token birden fazla desene uyuyordu. Öncelik sırası belirlenerek çözüldü.
* **GUI Donmaları:** Arayüz iş parçacığında işlem yapılması nedeniyle sistem yanıt vermiyordu. SwingUtilities.invokeLater() ile çözüldü.
* **Alt Satıra Geçince Kayma:** Kod yazılırken her alt satıra geçildiğinde renklendirmede kaymalar meydana geldi. Yazılan her bir kelimenin örnek olarak 3 harfi beyaz 2 harfi mavi şeklindeydi. Regex kodlarındaki düzenlemeler sayesinde sorun çözüldü.

**Sonuç ve Yayınlar**

Proje, kullanıcı dostu ve teknik açıdan sağlam bir uygulamadır. Harici kütüphane kullanılmadan gerçekleştirilen bu uygulama, aynı zamanda performans ve okunabilirlik açısından mümkün olduğunca optimize edilmeye çalışılmıştır.

Uygulamanın kaynak kodlarına ve tanıtım videosuna aşağıdaki bağlantılardan ulaşabilirsiniz:

* **GitHub Kaynak Kodu:**  
  <https://github.com/yagmurabdil/Real-Time-Syntax-Highlighter>
* **Demo Videosu:**  
  <https://www.youtube.com/watch?v=Kd8eK0wmBqo>