

## 1. Genel Bakış

Bu proje, **gerçek zamanlı sözdizimi vurgulama** özelliğine sahip bir GUI uygulaması geliştirmeyi amaçlamaktadır. Python ve Tkinter kullanılarak **modüler bir yazılım mimarisi** uygulanmıştır. Bu dokümanda tasarım kararları ve kullanılan yöntemler detaylandırılmıştır.

---

## 2. Dil ve Dilbilgisi

- Basit bir programlama dili oluşturulmuştur.
  - Dil aşağıdaki yapı taşlarını destekler:
    - **Aritmetik ifadeler** (örneğin:  $a + b$ )
    - **Değişken atamaları** (örneğin:  $x = 5$ )
  - **Bağlamdan bağımsız dilbilgisi (CFG)** kullanılarak sözdizimsel analiz temel alınmıştır.
- 

## 3. Sözcüksel Analiz

- **Yöntem:** Tablo tabanlı durum diyagramı
  - **Gerekçe:** Regex temelli token eşleştirme; hızlı, esnek ve Python `re` modülüyle kolay uygulanabilir.
  - **Uygulama:**
    - `LexicalAnalyzer` sınıfı; tanımlı düzenli ifadeleri kullanarak girdi metni token'lara ayırır.
    - Hatalı karakterler tespit edilip özel hatalarla bildirilir.
- 

## 4. Sözdizimsel Analiz

- **Yöntem:** Bottom-up (aşağıdan yukarıya) Shift-Reduce ayrıştırma
  - **Gerekçe:** Shift-reduce yöntemi, yığın tabanlı yapısıyla basit diller için etkilidir ve gerektiğinde genişletilebilir.
  - **Uygulama:**
    - `SyntaxAnalyzer` sınıfı token akışını alır ve dilbilgisel kurallara göre üretim yapar.
    - Yanlış yapılar tespit edildiğinde uygun hata mesajları ile redüksiyon başarısız olur.
- 

## 5. Vurgulama

- **Yöntem:** Tkinter `Text` widget'ında tag tabanlı renklendirme
  - **Gerekçe:** Tkinter'in tag sistemi, gerçek zamanlı metin vurgulama için uygundur.
  - **Uygulama:**
    - `Highlighter` sınıfı, token'ları analiz edip her birini belirli tag'lerle eşleştirir.
    - Her tuş vuruşunda token listesi güncellenir ve ilgili bölümler yeniden vurgulanır.
- 

## 6. Grafik Arayüz (GUI)

- **Yöntem:** Tkinter üzerinden oluşturulmuş kullanıcı arayüzü
  - **Gerekçe:** Tkinter, Python ile birlikte gelen, basit ama etkili bir GUI aracıdır.
  - **Uygulama:**
    - `SyntaxHighlighterGUI` sınıfı, kullanıcıdan girdi alır, analizleri çağırır ve hataları gösterir.
- 

## 7. Modülerlik

Proje aşağıdaki modüllere ayrılmıştır:

| Modül Adı                   | Görev Açıklaması             |
|-----------------------------|------------------------------|
| <code>lexer.py</code>       | Sözcüksel analiz işlemleri   |
| <code>parser.py</code>      | Sözdizimsel analiz ve üretim |
| <code>highlighter.py</code> | Vurgulama mantığı            |
| <code>gui.py</code>         | Arayüz yönetimi              |
| <code>main.py</code>        | Uygulama başlatıcı dosyası   |

Bu yapı, **bakımı kolaylaştırır**, **test yazmayı hızlandırır** ve gelecekteki geliştirmelere zemin hazırlar.

---

## 8. Testler

- **Yöntem:** Python `unittest` kütüphanesi
  - **Kapsam:**
    - Her modül için birim test senaryoları oluşturulmuştur.
    - `lexer`, `parser` ve `highlighter` bileşenlerinin doğru çalıştığı test edilmiştir.
- 

## 9. Sonuç

Tasarım, **basitlik** ve **esneklik** arasında başarılı bir denge kurar.

Yapı, ileride şu özelliklerle geliştirilebilir:

- Daha karmaşık dilbilgisi kuralları
- Yeni token türleri
- Farklı sözdizimi vurgulama temaları