BURSA TEKNİK ÜNİVERSİTESİ

# Bilgisayar Mühendisliği Bölümü

## Programlama Dilleri Dersi Proje Final Raporu

## C Syntax Highlighter

### Öğrenci Bilgileri

Ad Soyad: Ahmet Çimen

Numara: 23360859073

## İçindekiler

1. Giriş

2. Proje Gereksinimleri

3. Kullanılan Teknolojiler ve Yaklaşımlar

4. Implementasyon Detayları

5. Sonuç ve Değerlendirme

## 1. Giriş

Bu proje, C programlama dili için sözdizimi vurgulama özelliğine sahip bir GUI uygulaması geliştirmeyi amaçlamaktadır. Uygulama, sözdizimi analizi, sözcüksel analiz ve ayrıştırma işlemlerini formal bir gramer yapısına dayalı olarak gerçekleştirmektedir.

## 2. Proje Gereksinimleri

Proje kapsamında aşağıdaki temel gereksinimler belirlenmiştir:

1. Sözdizimi vurgulama özelliği

2. Grafiksel kullanıcı arayüzü (GUI)

3. En az 5 farklı token tipinin vurgulanması

4. Lexical ve Syntax analiz implementasyonu

5. Dokümantasyon ve demo video

## 3. Kullanılan Teknolojiler ve Yaklaşımlar

### Programlama Dili ve Kütüphaneler

Uygulama Dili: Python

GUI Framework: Tkinter

Hedef Dil: C Programming Language

### Sözcüksel Analiz (Lexical Analysis)

Projede "State Diagram & Program Implementation" yaklaşımı tercih edilmiştir:

**Token Tipleri ve Renk Şeması**

|  |  |  |
| --- | --- | --- |
| Token Tipi | Renk | Örnek |
| Anahtar Kelimeler | Mavi | int, void, return |
| Stringler | Yeşil | "Hello World" |
| Yorumlar | Açık Yeşil | // Bu bir yorum |
| Önişlemci Direktifleri | Mor | #include |
| Sayılar | Kırmızı | 42, 3.14 |
| Operatörler | Siyah | +, -, \*, / |
| Parametreler | Turuncu | void foo(int param) |

### Ayrıştırıcı (Parser)

Top-Down Parsing yaklaşımı kullanılmıştır:

1. Recursive Descent Parser

- Preorder traversal ile parse tree oluşturma

- Hiyerarşik sözdizimi analizi

- Detaylı hata tespiti ve raporlama

2. Hata Yönetimi

- Sözdizimi hatalarının tespiti

- Error sekmesinde hata konumlarının gösterimi

- Hata mesajları ve konumları

## 4. Implementasyon Detayları

### Proje Yapısı

Proje/

├── gui.py # GUI implementasyonu

├── Lexer.py # Sözcüksel analiz

├── Parser.py # Sözdizimi analizi

└── error.py # Hata yönetimi

### GUI Özellikleri

1. Metin editörü

2. Token listesi görüntüleme

3. Ayrıştırma ağacı görselleştirme

4. Hata listesi ve vurgulama

5. Anlık olarak highlight gösterimi

## 5. Sonuç ve Değerlendirme

Proje, belirlenen hedeflere büyük ölçüde ulaşmıştır:

**1. Başarılar**

- 8 farklı token tipinin başarılı vurgulanması

- Kullanıcı dostu GUI implementasyonu

- Detaylı hata tespiti ve raporlama

- Gerçek zamanlı analiz performansı

**2. Geliştirme Alanları**

- Daha kapsamlı C dili desteği