

Bursa Teknik Üniversitesi  
BLM0238 Programlama Dilleri

**Projeden Sorumlu Öğrencinin:**

Adı:Yusuf

Soyadı:Çil

Numarası:22360859012

1. Projeye Genel Bakış (Introduction)
2. Projede Kullanılan Yöntemler
3. Kod Uygulamaları (Implementation)
4. Sonuç
5. Proje Kodundaki Eksikler ve Eklenmesi Planlanan Geliştirmeler
6. Kaynakça

## 1. Projeye Genel Bakış (Introduction)

Bu projenin temel amacı gerçek zamanlı olacak şekilde uygulamanın paneline yazılan program parçacıklarını tokenlerine ayırmak ve tokenlerine ayırdığı kod parçacıklarını gerçek zamanlı olacak şekilde analiz etmek ve panel üzerinde renklendirmektir. Kod analizi düzgün yapılmadığında program tasarlayan ve kod yazan bir kişi ufak dikkat hatalarından dolayı dilin syntax kurallarına uymakta zorlanabilmekte ve hata yapabilmektedir. Bu projede hedeflenen şey, kişinin daha geniş kapsamlı IDE'lerde olduğu gibi yazılan satırlarda renklendirme yapılarak kod yazan kişinin hata yapma sayısını azaltmak ve yazılan kodun dilin genel yazım kurallarına uygun olup olmadığını inceleyerek kişiye göstermektir.

## 2. Projede Kullanılan Yöntemler

Projeyi gerçekleştirebilmek için başvurulmuş başlıca yöntemler lexical analysis ve syntax analysis yöntemlerine kullanılmıştır. Eğer bu yöntemlere daha detaylıca yaklaşacak olursak lexical analysis için kullanılan başlıca yöntem state diagram & program implementation olmuştur. Bu yöntemle ilgili olarak kod içerisinde token türleri oluşturulmuş ve token türleri listeler içerisinde sabit olacak şekilde tanımlanmıştır. Tokenlerin sınıflandırılması işlemi aynı sınıf içerisinde yer alan detectTokenType() metodu ile yapılmaktadır.

Syntax analysis içinse Top-Down Parsing yöntemi kullanılmıştır. Bu yöntem, parse sınıfı içerisinde parsing işlevi gören metotların recursive şekilde birbirlerini çağırması yoluyla sağlanmıştır. Her metot aynı zamanda bir CFG kuralının gerçekleştirilmesini sağlamaktadır.

### 3. Kod Uygulamaları

Projenin kodlaması yapılırken kullanılan programlama dili Java'dır. Kodları analiz etmek için ölçüt alınan dil ise C programlama dilinde yer alan temel ve daha komplike aritmetik işlemler, karşılaştırma ve mantıksal ifadeler (if conditions), genel fonksiyon çağrılar ve bazı özel fonksiyonlar (printf() ve scanf() fonksiyonları gibi), atama ifadeleri, değişken tanımlama ifadeleri, o programlama dilinde anahtar kelime görevi gören bazı rezerve kelimeler ve son olarak while döngüsünü içeren bir C programlama dili alt kümesidir. Bu özellikler sayesinde temel aritmetik işlemleri gerçekleştiren, mantıksal işlemler yapabilen, o esnada program içerisinde bulunduğu varsayılan başka alt programları (fonksiyonları) çağırabilen ve analiz edebilen, dönüş tipine bağlı olarak yazılan alt programı sonlandırabilen kod parçacıkları analiz edilebilmektedir.

Proje Java yazılım dili ile gerçekleştirildiği için Java'nın güçlü yönlerinden olan sınıf yapısı proje kodunda kullanılmıştır. Projede birisi main metodunun bulunduğu Test sınıfı olmak üzere toplamda 5 tane sınıf yapısı kullanılmıştır. Bu sınıflar Token sınıfı, SimpleHighlighterGUI sınıfı, SimpleTokenizer sınıfı, Parse işlemlerini gerçekleştirildiği Parse sınıfı ve Test sınıfıdır.

Bu sınıflardan sırasıyla bahsedecek olursak Test sınıfı main metodunun yer aldığı ve programın başladığı sınıf olarak tanımlanabilir.

SimpleTokenizer sınıfı, tokenlerin tiplerine göre set halinde listeler içerisinde tutulduğu ve program başladıktan sonra tokenlerin türlerinin analiz edilme işlemlerinin gerçekleştiği sınıftır.

Parse sınıfı C programlama dilinin alt kümesinin içerdiği CFG (Context Free Grammer) kurallarına göre parsing işlemlerinin yapıldığı sınıftır.

SimpleHighlighterGUI sınıfı, program çalıştırıldığında ekrana açılan panelin ve panelin özelliklerinin yer aldığı sınıftır.

Token sınıfı, diğer sınıflarda kullanılan tokenler ve özelliklerini kullanabilmek için oluşturulmuş bir sınıftır.

## 4. Sonuç

Kod çalıştırıldığında ekrana bir panel açılmakta. Bu panel üzerinde gözüken imleç yardımıyla yazmak istediğiniz kodu ekrana yazabiliyorsunuz. Alt satıra geçmek için klavyenizde bulunan enter tuşuna basmanız gerekiyor. Boşluk karakteri girene kadar alt satıra geçme işlemi devam ediyor. Boşluk karakteri girildiğinde ise panele yapılan kod girişi durduruluyor. Tüm bu işlemler gerçekleştirilirken yazdığınız kod üzerinde gerçekleştirilen tokenlerine ayırma ve parsing işlemlerini ekrandan takip edebiliyorsunuz. Parsing işleminin doğruluğu ya da yanlışlığı ekrana açılan panelin alt kısmında doğruluk durumuna göre yeşil ya da kırmızı renkler ile kullanıcıya aktarılıyor. Tokenlere ayırma işleminin sonucunu ise programı açmak için kullandığınız IDE uygulamasının konsolundan takip edebiliyorsunuz.

## 5. Projedeki Eksikler ve Geliştirmeler

Bu proje, ne kadar C programlama dilinin temel özelliklerinin alt kümesini ve içeriğini kapsasa da analiz etmek için yazılan kodların karmaşıklığı arttıkça ve yazılan kodun konu olarak içeriği genişledikçe (pointer, struct, diziler gibi yapılar) analiz yapma ve tokenlere ayırma işlemi başarısız bir hal almaktadır. Buna ek olarak yazılan kodlarda konu olarak farklılık olmasa bile bazı noktalarda işlemleri analiz etmede programın yetersiz kalabildiği hissedilmektedir.

## 6. Kaynakça

Intro\_to\_Java\_Programming(10th\_Edition) online pdf

[www.openAl.com](http://www.openAl.com)

Eclipse IDE

Visual Studio Code IDE-Copilot

