



BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

BLM0238 PROGRAMLAMA DİLLERİ

Dönem Projesi – Canlı Sözdizimi Vurgulayıcı

RUMEYSA ERSOY

22360859070

2024-2025

1)Proje Amacı

Bu proje kapsamında, Python benzeri girinti tabanlı bir programlama dili için gerçek zamanlı sözdizimi vurgulayıcı geliştirilmiştir. Proje, kaynak kodun lexing ve parsing aşamalarını gerçek zamanlı olarak gerçekleştirerek, kod yazarken anlık sözdizimi hatalarını tespit etmeyi ve kullanıcılara görsel geri bildirim sağlamayı amaçlamaktadır. Bu sayede, kullanıcılar kodlarını daha hızlı ve hatasız yazabilmektedir.

2) Literatür Taraması

Sözdizimi vurgulama, birçok kod editöründe bulunan yaygın bir özelliktir ve genellikle düzenli ifadelerle yapılır. Ancak girinti tabanlı dillerde, özellikle Python’da, kod bloklarının doğru anlaşılması için girinti seviyelerinin izlenmesi gerekir. Context-free grammar (CFG) tabanlı parserlar, bu yapıyı doğru şekilde analiz etmek için kullanılır. Bu proje, gerçek zamanlı lexing ve parsing yöntemlerini birleştirerek bu ihtiyaca cevap vermektedir.

Sözdizimi vurgulayıcıların çoğu yüzeysel renklendirme yaparken, bu projedeki yapı context-free grammar’a dayalı gerçek zamanlı bir sözdizimsel kontrol içerdiği için geleneksel yapıların ötesindedir. Bu tür sistemler, dil öğreniminde ve hata ayıklamada güçlü faydalar sağlar.

3. Yöntem ve Uygulama

Bu projede, gerçek zamanlı sözdizimi vurgulayıcı geliştirmek için üç ana bileşen kullanılmıştır: lexer (tokenizer), parser ve grafiksel kullanıcı arayüzü (GUI).

- **Lexer:** Kaynak kodu, anlamlı tokenlara ayırmak için Python'un `re` modülü ile yazılmış düzenli ifadeler kullanılmıştır. Girinti tabanlı yapıyı desteklemek için satır başındaki boşluklar analiz edilerek `INDENT` ve `DEDENT` tokenları oluşturulmuştur.
- **Parser:** Recursive descent yöntemiyle geliştirilmiş parser, lexer'dan gelen tokenları kullanarak kodun sözdizimini kontrol eder. Atama, `if-else` ve `print` gibi temel ifadeleri analiz eder ve hatalı sözdizimi durumunda anlamlı hata mesajları verir.
- **GUI:** Tkinter kütüphanesi ile oluşturulan arayüz, kullanıcı yazarken gerçek zamanlı olarak kodu lex edip parse eder. Farklı token türleri renklendirilir ve sözdizimi hataları anlık olarak kullanıcıya gösterilir.

Projede, bu bileşenler entegre edilerek hızlı ve kullanıcı dostu bir sözdizimi vurgulayıcı elde edilmiştir.

3.1 Örnek CFG Tanımı:

Bu projede kullanılan parser, aşağıdaki gibi tanımlanmış context-free grammar kurallarına dayanır:

Program \rightarrow StatementList

StatementList \rightarrow Statement StatementList $| \epsilon$

Statement → Assignment | IfStatement | PrintStatement

Assignment → IDENTIFIER "=" Expression

IfStatement → "if" Expression ":" NEWLINE INDENT StatementList DEDENT

PrintStatement → "print" "(" Expression ")"

Expression → IDENTIFIER | NUMBER | STRING | Expression Operator

Expression

Operator → "+" | "-" | "*" | "/"

3.2 GUI Yapısı:

Tkinter ile oluşturulan arayüzde Text bileşeni kullanılarak kullanıcı girişleri alınmakta, tag_config fonksiyonları ile belirli token türlerine renk atanmaktadır. Örneğin:

```
text_widget.tag_config("KEYWORD", foreground="blue")
```

```
text_widget.tag_add("KEYWORD", "1.0", "1.2")
```

4) Sonuçlar ve Değerlendirme

Geliştirilen gerçek zamanlı sözdizimi vurgulayıcı, Python benzeri girinti tabanlı bir dilde başarılı şekilde çalışmıştır. Lexing ve parsing aşamaları gerçek zamanlı olarak kod yazımında anında geri bildirim sağlamış,

sözdizimi hataları doğru ve anlaşılır şekilde tespit edilmiştir. Kullanıcı arayüzü sayesinde farklı token türleri kolayca ayırt edilerek kod okunabilirliği artırılmıştır.

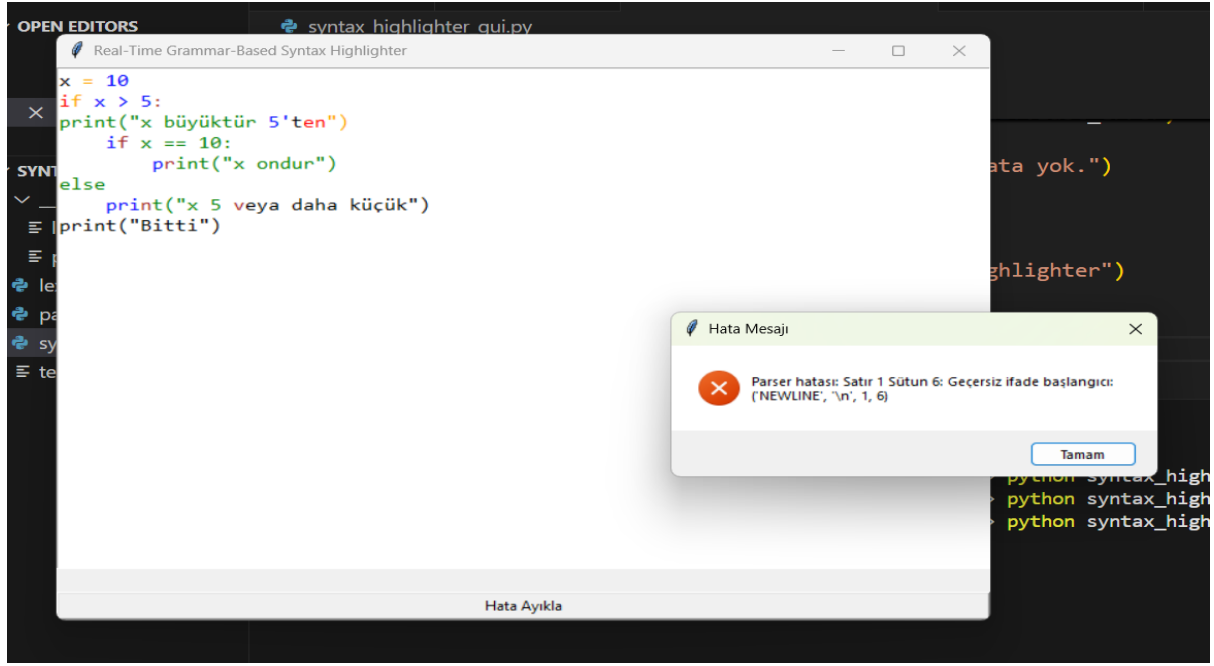
Projede bazı kısıtlamalar da görülmüştür. Örneğin, karmaşık ifadelerin ve ileri seviye dil özelliklerinin tam desteği sağlanamamıştır. Ayrıca, büyük ve uzun kodlarda performans optimizasyonlarına ihtiyaç duyulabilir.

Gelecekte, parser'ın kapsamının genişletilmesi, daha gelişmiş hata raporlama ve otomatik düzeltme özelliklerinin eklenmesi planlanmaktadır. Böylece, projenin hem eğitim amaçlı hem de pratik kullanımlarda daha etkili bir araç olması hedeflenmektedir.

Aşağıda, proje arayüzünün çalışır haldeki örnek ekran görüntüsü yer almaktadır:



```
x = 10
if x > 5:
    print("x büyüktür 5'ten")
    if x == 10:
        print("x ondur")
    else
        print("x 5 veya daha küçük")
print("Bitti")
```



5. Sonuç ve Öneriler

Bu projede, Python benzeri girinti tabanlı bir programlama dili için gerçek zamanlı sözdizimi vurgulayıcı başarıyla geliştirilmiştir. Sistem, lexing ve parsing işlemlerini gerçek zamanlı yaparak kullanıcıya anlık hata bildirimi sağlamış ve kod okunabilirliğini artırmıştır. Proje, özellikle girinti tabanlı dillerde sözdizimi vurgulama ve hata tespiti alanında önemli bir katkı sunmaktadır.

Gelecekte, dil desteğinin genişletilmesi, daha karmaşık dil yapılarının işlenmesi ve performans iyileştirmeleri gibi geliştirmeler yapılabilir. Ayrıca, otomatik hata düzeltme ve öneri sistemleri eklenerek kullanıcı deneyimi daha da zenginleştirilebilir.