

TBL220 Python Programlama

Final Proje Raporu

Kocaeli Üniversitesi
Teknoloji Fakültesi
Bilişim Sistemleri Mühendisliği
Barış Adıgüzel - 181307059

ÖZET

İçerikten bağımsız gramer (context free gramer) dil tasarımlarında kullanılan bir gramer tipidir. Bizden istenen dil kurallarında ikili sembollerle oluşturulan (örneğin 0 ve 1 karakterleri) kelimelerin palindrom (tersten okunuşu ile aynı) özelliğe sahip olmasıdır. Bu yapıyı biz Python'ın nesneye dayalı programlama özelliklerini kullanarak, yani sınıfları, yapabiliriz.

Sınıf ve kalıtım (Inheritance) yapılarını kullanarak istenilen dil kurallarını gerekli gördüğümüz fonksiyonları oluşturup onların üstünden işlemlerimizi yaparız.

A. Giriş

İçerikten bağımsız gramer, belli bir alfabe üzerinden dil tanımlamıza yardımcı olur. Bizden istenen, ikili sembollerden oluşturulan kelimelerin palindrom özelliğe sahip olması istenmektedir. Bu ikili semboller her hangi iki karakter olabilir.

Biz problemimizde 0 ve 1 karakterlerini kullanarak bir palindrom kelime üretmeye çalıştık. Palindrom bir kelime üreten bir Palindrom isminde sınıf tanımlaması yaptık. Bu sınıftan türetilen her objenin rastgele uzunluğa sahip bir palindrom kelime sahip olması için yapılandırıcı fonksiyonunda kurallarını belirttik.

Bizden ekstra olarak istenen 3 tane fonksiyon vardır. *yazdir* fonksiyonu, *test* fonksiyonu ve *uzunluk* fonksiyonudur.

yazdir fonksiyonu, oluşturulan palindrom kelimeyi ve uzunluğunu gösterir.

test fonksiyonu, kendisine gelen parametreyi palindrom bir kelime olup olmadığına bakar. Palindrom bir kelime gelmiş ise bir değişken kullanarak gelen kelimenin açılımı yapılır. Örneğin $w=1001$ için 1A1, 10A01, 1001 şeklinde bir açılım serisini gösteririz.

Son olarak *uzunluk* fonksiyonu, gelen iki palindrom kelimenin (değiller ise bir bilgi mesajı verilir) uzunluklarını kıyaslar.

Ekstra olarak Palindrom sınıfından bir kalıtım yapılarak ÜnlemlüPalindrom sınıfı yapmamız istenmektedir. Tek değişken durum oluşan kelimenin başı ve sonuna ünlem işareti karakteri gelmektedir.

B. Yöntem

Her bir algoritmik problem için birden fazla yöntem mevcuttur. Benim oluşturduğum gramer kurallarında kullandığım yöntemi kısaca özetlemem gerekirse:

- Palindrom sınıfından her nesne üretildiğinde 0-50 sayıları aralığında oluşturulacak kelimenin uzunluğunu belirle.
- Uzunluğun yarısını alarak 0 ve 1 karakterlerinden oluşan bir liste oluştur.
- Oluşan listeyi ters çevirip başka bir liste oluştur.
- Eğer oluşacak kelimenin uzunluğu çift ise, ilk liste ile ters çevrilen listeyi birleştirip kelimeyi oluştur.
- Eğer oluşacak kelimenin uzunluğu tek ise, ilk liste ile ters çevrilen listenin arasına rastgele 0 veya 1 karakteri koyarak listeleri birleştir ve kelimeyi oluştur.

C. Pseudo Kod

Sınıfın yapılandırıcı fonksiyonu

- Palindrom sınıfından bir nesne türet
- 0-50 aralığında bir uzunluk belirle
- Uzunluk sıfır ise kelime olarak epsilon karakterini ata
- Uzunluğun yarısını al ve 0, 1 karakterlerinden oluşan bir liste oluştur
- Oluşan listeyi ters çevirip başka bir liste oluştur.
- Eğer oluşacak kelimenin uzunluğu çift ise, ilk liste ile ters çevrilen listeyi birleştirip kelimeyi oluştur.
- Eğer oluşacak kelimenin uzunluğu tek ise, ilk liste ile ters çevrilen listenin arasına rastgele 0 veya 1 karakteri koyarak listeleri birleştir ve kelimeyi oluştur.

yazdir() metodu

- Sınıftan oluşturulan nesnenin, kelimesi ve uzunluğunu ekranda göster

test() metodu

- Gelen argümanın string olup olmadığını kontrol et, string değil ise "string bekleniyordu" mesajı ver
- Gelen argümanın palindrom olup olmadığını kontrol et, palindrom kelime değil ise "palindrom bekleniyordu" mesajı ver
- Palindrom ise, kelimenin uzunluğunu kontrol et

4. Uzunluk çift ise, kelimenin yarısını al ve yeni oluşturulacak kelime argüman olarak gelen kelimenin uzunluğuna eşit olana kadar ekranda bastır
5. Uzunluk tek ise, kelimenin yarısını al ve yeni oluşturulacak kelime argüman olarak gelen kelimenin uzunluğundan büyük olana kadar ekranda bastır

uzunluk() metodu

1. Gelen argümanların string olup olmadığını kontrol et, birisi bile string değil ise “string bekleniyordu” mesajı ver
2. Gelen argümanların palindrom olup olmadığını kontrol et, birisi bile palindrom kelime değil ise “palindrom bekleniyordu” mesajı ver
3. Gelen ilk argüman, ikinci argümanın uzunluğundan büyük ise “ilk kelime büyüktür” yazdır
4. Gelen ikinci argüman, ilk argümanın uzunluğundan büyük ise “ikinci kelime büyüktür” yazdır
5. 3. veya 4. adımdaki koşullar oluşmuyorsa “iki kelimenin uzunluğu aynıdır” yazdır

D. Örnek Sonuçlar

```
p_word1.yazdir()
-----
Kelime : 110011110011
Uzunluk: 12

p_word2.yazdir()
-----
Kelime : 001010100101001010100
Uzunluk: 21

p_word3.yazdir()
-----
Kelime : !100101101010110100!
Uzunluk: 21
```

Şekil 1 – Sınıftan türetilen objeler ve yazdir() fonksiyonu

p_word1 ve p_word2 objeleri Palindrom sınıfından, p_word3 objesi UnlemlıPalindrom sınıfından türetilmiştir. Üstteki şekilde oluşan rastgele uzunluktaki kelimeleri ve uzunluklarını görüyoruz.

```
Palindrom.test('AHMETEMA')
-----
Girilen kelime, palindrom bir kelime değildir.

Palindrom.test(p_word1.word)
-----
1#1
11#11
110#011
1100#0011
11001#10011
110011#110011
110011110011
```

Şekil 2 – test() fonksiyonun çıktıları

Yukarıdaki şekilde Palindrom sınıfındaki test() fonksiyonuna gelen argümana göre oluşturduğu çıktıları görüyoruz.

```
Palindrom.uzunluk('2312', 'AsA')
-----
Argüman olarak gelen kelimelerin palindrom olması gerekmektedir.

Palindrom.uzunluk(p_word1.word, p_word2.word)
-----
İkinci kelimenin uzunluğu, ilk kelimenin uzunluğundan büyüktür

Palindrom.uzunluk(p_word2.word, p_word2.word)
-----
Girilen iki kelimenin uzunluğu birbirine eşittir
```

Şekil 3 – uzunluk() fonksiyonun çıktıları

Şekil 3'te Palindrom sınıfındaki uzunluk() fonksiyonuna gelen argümanlara göre oluşturduğu çıktıları görüyoruz.

Kaynakça

- [1] <https://www.programiz.com/python-programming/methods/list/reverse>
- [2] https://www.python-course.eu/python3_class_and_instance_attributes.php
- [3] https://www.bogotobogo.com/python/python_differences_between_static_method_and_class_method_instance_method.php
- [4] [Python OOP Tutorials – Working with Classes, Youtube playlist](#)