

# YAZILIM LABORATUVARI 2

## 1. PROJE

Barış KAKILLI

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[baris\\_kakilli38@hotmail.com](mailto:baris_kakilli38@hotmail.com)

### Özet

Bu rapor Yazılım Laboratuvarı - 1 dersi 1. proje için açıklamaya yöneliktir.

Raporda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Rapor sonunda projemi hazırlarken kullandığım kaynaklar bulunmaktadır.

### 1. Proje Tanımı

Projenin amacı Java dili kullanılarak iki veya daha fazla metnin uygun bir şekilde birleştirilmesini sağlamak. Geliştirilen algoritmanın performansı önemlidir bu sebeple çalışması süresini de hesaplıyoruz.

Yaptığımız birleştirme işlemini kullanıcı istediği takdirde bir veritabanına kayıt olarak ekleyebilecek. Proje için istenen veritabanı MongoDB olarak belirlenmiştir. Hem birleştirme hem de veritabanı operasyonları için Java dili kullanılmıştır.

### 1.1 Projenin Özellikleri

Projede web kısmı için java spring boot kullanılmıştır.

Başlangıç sitesinde kullanıcının yapabileceği 4 işlem vardır.

- 1- Giriş kutusu ekleme
- 2- Giriş kutusu silme
- 3- Girilen metinleri birleştirme işlemi için gönderme
- 4- Veritabanında bulunan tüm verileri strings.json isimli dosyaya kaydetme.

### StringConcat

Add	Remove	Send	Export
•	<input type="text"/>		
•	<input type="text"/>		
•	<input type="text"/>		

Kullanıcının 'Send' işlemini seçmesinin ardından ikinci bir sayfaya yönlendiriliyor ve bu sayfada birleştirme işlemine ait sonuçlar yer alıyor.

## Words Sent

- merdivenden çıktı ve ali eve geldi
- ali eve geldi kapıyı açtı
- kapıyı açtı ve eve girdi

## Result

Concatenated words: merdivenden çıktı ve ali eve geldi kapıyı açtı ve eve girdi

## Time Took in ms

Time took: 0.0797

[save to MongoDB](#)

## 2. Sayfa örnek görüntüsü

İkinci sayfa için bir adet buton bulunuyor bu da veriyi MongoDBye kaydetme butonu.

```
{
  "result": "merdivenden çıktı
ve ali eve geldi kapıyı açtı
ve eve girdi",
  "duration": 0.9511,
  "strings": [
    "ali eve geldi kapıyı
açtı",
    "kapıyı açtı ve eve girdi",
    "merdivenden çıktı ve ali
eve geldi"
  ],
  "_id": {"_id": "641cc4beeab91d616945a3e8"},
  "_class": "com.yazlab.stringconcat.Strings"
}
```

Yukarıda örnek bir kayıt gözükmemtedir.

Projenin veritabanından strings.json dosyasına çevirme işlemi için koleksiyondaki tüm verileri alıp javanın dosyalama işlemleri ile kaydediliyor ve dosya sonrasında kapatılıyor.

## 2 - Birleştirme İşleminin Yapımı

Projenin en önemli noktası birleştirme işlemi ve bu işlem için 3 adet fonksiyon kullanıyorum

A - sortStringsAndConcat:

bu fonksiyon ilk olarak gelen `List<String>` içerisindeki tüm stringleri tek bir kelimedenden oluşup oluşmadığını kontrol ediyor eğer hepsi tek bir kelimedenden oluşuyorsa 2. fonksiyon olan `ConcatForWords`'ü çağırıyorum.

B - `ConcatForWords`:

Bu fonksiyonun amacı birleştirme işlemini kelime bazlı yapmasıdır çalışma şekli ise aşağıdaki gibidir.

Kelime kontrolünü ve cümle kontrolünü ayırmamdaki sebep proje için verilen örneklerde bu ikisinin farklı bir şekilde işlendiğini görmemdir kelime bazlı birleştirme işlemini verilen "abcdefgh" ve "abcefg" kelimelerinde test etmiş olup aynı sonucu verdiğini teyit ettim. Bu verilen örneğe ve bizden istenen istere göre de uygun programı aşağıda açıkladığı şekilde gerçekleştirdim.

İlk olarak gelen listedeki kelimelerin hepsi ortak harf kontrolüne tabi tutuluyor ve ortak olan tüm kelimeler kaydediliyor ardından gelen listedeki son kelimenin geri kalan kısmı bu ortak kelime kümesine dahil ediliyor ve sonuç olarak geri döndürülüyor.

Programımız

`List<String>`

içerisindeki tüm stringlerin tek bir kelimedenden oluşmadığı durumda bu fonksiyona girmeden 1. fonksiyonda devam ediyor.

C - sortStringsAndConcat devamı:

Kelime bazlı olmadığı için programımıza buradan devam ediyoruz ilk yaptığımız işlem kısa bir cümle eğer uzun bir cümle içerisinde tamamı ile geçiyorsa o kısa cümleyi silmek bunu yapma sebebimiz hem programdaki beklenmedik sonuçları kaldırmak hem de programı hızlandırmak. Bu şekilde yaptığımızda hem aynı olanları kaldırıyoruz hem de kısa olanları o yüzden bu kısım güzel bir ilk kontrol sağlıyor.

Bu adımdan sonra yaptığımız işlem sıralama birleştirme işleminde bir önceki string ile bir sonraki stringi işleme tabi tuttuğum için bu stringlerin öncelikle düzgün bir sırada olması lazım.

Bize verilen örneklerde bu sıra her zaman doğru bir şekilde verilmiş olsa da programın doğru sırada verilmeyen stringler için de çalışmasını sağlamak için bu sıralama işlemi gerekli bir işlem.

public interface Comparator<T>

```
unsorted.sort((s1, s2) -> {  
    . . .  
})
```

sıralama işlemini java da mevcut olan comparator interface ini programıma uyarlıyorum. Bu interface eğer 0 dönerse karşılaştırılan stringlerin yerini değiştirmiyor negatif ve pozitif sayı dönerse stringleri ona göre değiştiriyor.

Sıralama için 2 for içinde tüm stringleri dolaşıyorum eğer aynı olan bir kelime denk gelirse başka bir döngüye giriyor ve burada s1 stringi ile s2 stringi aynı şekilde ilerliyor mu kontrol ediliyor.

eğer aynı şekilde ilerliyorlarsa bu benzerlik var demektir ve bu durumda birisi öne birisi arkaya geçmek zorunda orada da kelimenin uzunlukları devreye giriyor.

```
"merdivenden çıktı ve ali eve geldi",
```

```
"ali eve geldi kapıyı açtı"
```

Yukarıda gördüğümüz iki kelimenin "ali eve geldi" kısmı aynı. birisi kelimenin en sonunda diğeri ise başında kontrol yaptığımızda "ali eve geldi" kısmı sonda olduğu için bu cümle daha önce gelmesi gerekiyor demektir o yüzden bu cümleyi öne alıyoruz.

Bu sıralama işlemi tamamlandıktan sonra bu fonksiyonun en alt kısmına gelmiş oluyoruz burada ilk kısımda kelimeler için çağırdığımız fonksiyon olan ConcatForWords yerine cümleler için olan ConcatForSentences'i çağırıyoruz ve fonksiyon burada son buluyor.

D - ConcatForSentences:

Bu fonksiyonun amacı birleştirme işlemini cümle bazlı yapmasıdır.

Aşağıda konu ile ilgili bir örnek verilmiştir.

Text 1: Ali eve gel

Text 2: eve gel sonra

Text 3: eve gel sonra çarşı

Text 4: çarşıya git

Çıktı: Ali eve gel sonra çarşıya git

Birleştirme işlemi forumda verilen örnek ve bir arkadaşımızın forumda sorduğu soru baz alınarak yapılmıştır. 2 Örnek için de başarılı sonuç veren işlem sonradan kendi test ettiğim bir kaç örnek için de başarılı sonuçlar vermiştir programın çalışma şekli bizler tarafından belirlendiği için yaptığım program tam olarak istenen bir şekilde çalışıyor mu emin olmamakla birlikte test ettiğim 5 durumu da başarılı bir şekilde gerçekleştirmiştir. Şimdi işlemin nasıl olduğunu bakalım.

Cümlelerimizin hepsini sıralamıştık o yüzden ilk gelen cümleyi

```
StringBuilder sb = new  
StringBuilder();  
oluşturduğumuz StringBuilder a ekliyoruz.  
"eve gel sonra çarşı",  
"çarşıya git"
```

Yukarıda verilen stringler için ilki sb ikincisini ise sıradaki gelen cümle olarak varsayalım burada ilk olarak sb cümlesini split(" "); ile kelimelerine ayırıyoruz ve for döngüsünde her kelimeyi bu kelime sıradaki gelen stringin başlangıcında var mı? diye kontrol ediyoruz.

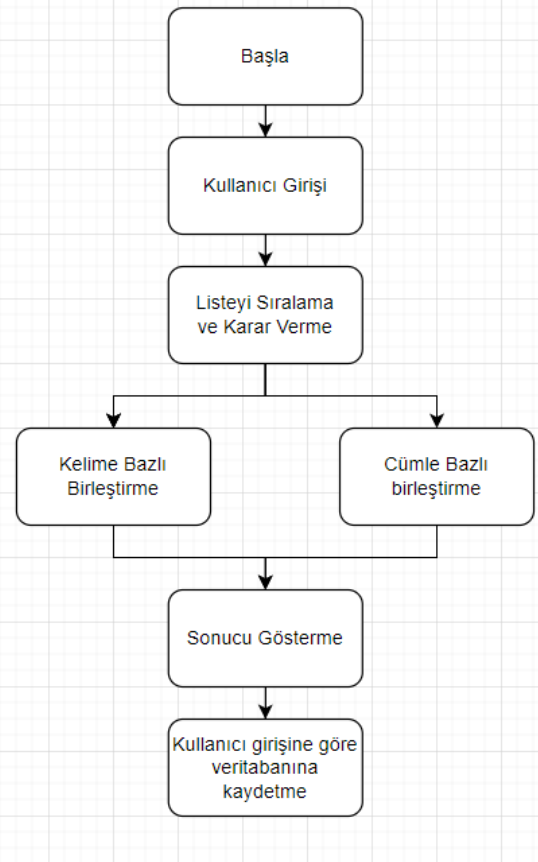
1. döngü: eve - hatalı
2. döngü: gel - hatalı
3. döngü: sonra - hatalı
4. döngü: çarşı - cümlemiz çarşı ile başlıyor.

evet kesmemiz gereken kısmı bulduk bundan sonra cümlelerin devamı da aynı gidiyor mu diye kontrol ediyoruz ama sb içinde kelime kalmadığından bu işlemi bu örnek için atlıyoruz. sırada yaptığımız işlem ise uzunluğu bulmak çünkü stringi birleştirirken "eve gel sonra çarşıçarşıya git" şeklinde bir birleştirme yapmak istemiyoruz sb içindeki kelime uzunluğumuz 5 o zaman 5 ileri atlayarak kelimeyi birleştiriyoruz ve "eve gel sonra çarşıya git" sonucunu elde ediyoruz. başarılı bir şekilde birleştirdik bu elde ettiğimiz birleşmiş stringi artık kullanabiliriz.

### 3- Geliştirme Ortamı

Projeyi IntelliJ IDEA 2022.3 versiyonunda geliştirdim. Projeyi tek başıma gerçekleştirdiğim için versiyon kontrol sistemi kullanmadım. web için java spring boot, gönderilen verileri html'de göstermek için thymeleaf kullandım.

## 4 - Kod Bilgisi



Akış Diyagramı

## Kaynakça

1. MongoDB Kullanımı:

<https://www.mongodb.com/compatibility/spring-boot>

<https://www.baeldung.com/queries-in-spring-data-mongodb>

2. Spring Web işlemleri:

<https://spring.io/guides/gs/rest-service/>

<https://www.jetbrains.com/help/idea/2022.3/spring-support.html>

3. karşılaştığım çeşitli problemler:

<https://stackoverflow.com/>

4. Akış diyagramı oluşturmak için:

<https://app.diagrams.net/>