# Test Cases

Aşağıdaki sayfalarda bulunan iki adet kodlamanın yapılıp

1. Soru için sadece metodun gönderilmesi

2. Soru için Maven veya Gradle projesi yaratılıp source kodlarla beraber bir zip dosyası olarak gönderilmesi beklenmektedir.

# 1. Java 8+ Knowledge Test

Aşağıdaki Özellikleri sağlayacak bir metot yazılmasını istiyoruz;

## 1.1. Parametreler

Parametre olarak stream alacak bu streamin elemanları da stream olacak.

İç streamin elemanları ise String objesi olacak.

Ancak yayımdaki sıkıntılar nedeniyle bu stringlerin içinde null olanlar ve nümerik olmayanlar var.

## 1.2. Metodun amacı

Yayımdaki sıkıntılar nedeniyle hatalı gelmiş olan verileri temizleyip aşağıdaki algoritma kullanılarak doğru şekilde filtrelenip bir List<Integer> elde etmek

## 1.3. Algoritma

iç streamlerin hepsi sıralı olarak ele alınacak yani ilk stream bittikten sonra ikinci streamin ilk elemanı ile devam edilecek.

Bu string objeleri öncelikle integer a çevrilecek ve bu integerlar 3 lü gruplara ayrılacak.

Bu üçlü gruplar içinde toplamları 90’dan küçük olanlar elenecek.

Büyük olanlar geldikleri sıra ile bir listenin içine konulup metottan return edilecektir.

Sonda üç adetten az Integer kaldıysa yok sayılacaktır ve dönüş listesine eklenmeyecektir

## 1.4. Metod

public static List<Integer> fixStream(Stream<Stream<String>> str) {

// write your code here

}

1.5. Örnek Yayım

1.5.1. Örnek Parametre

ilk Stream: "0", "s1", null, "35", "90", "60"

ikinci Stream: "ttt", null, null , "15",

üçüncü Stream: "75", "95", "0", "0", null, "ssss", "0", "-15"

Dördüncü Stream: "25", "fgdfg", "", "105", "dsfdsf", "-5"

### 1.5.2. Örnek Sonuç

0, 35, 90, 60, 15, 75, 95, 0, 0

# 2. Spring Boot Knowledge Test



Yukarıdaki mobil sayfanın tasarımını oluşturmayı planlıyoruz. Bu sayfa, filomuzdaki araçlar için küçük bir ekspertiz yapmayı amaçlamaktadır. Araç bilgileri başka bir mikro serviste tutulmakta ve mobil uygulama tarafından iletilen tek bilgi "carId" alanıdır. Bu sayfada araçla ilgili “carId” dışında herhangi bir bilgi saklama gerekliliği bulunmamaktadır.

Sorular, genel olarak başka bir ekrandan oluşturulacak gibi düşünülmelidir (Bu ekran ve gerekli servisler zaten mevcuttur). Her soru altında "Evet, var" veya "Hayır, yok" seçenekleri bulunacaktır. Eğer cevap "Evet, var" olarak işaretlenirse, zorunlu olarak en az 1 ve en fazla 3 adet fotoğraf çekilmeli ve bir açıklama eklenmelidir.

Not: Soruların veritabanına eklendiği varsayılabilir, bu da şu anlama gelir ki, veritabanı yapısını oluşturduktan sonra yeni sorular eklemek için ayrı bir servis yazma ihtiyacı bulunmamaktadır. Eklemeyi kolaylaştırmak için projeye ekleme scriptleri eklemeniz beklenmektedir.

Örnek görselde, sağ üst köşesinde kırmızı bir çarpı işareti olan fotoğraflar, yeni çekilenlerdir. Diğer fotoğraflar ise önceki ekspertizden alınmıştır.

**Burada sizden talebimiz, bu sistemin Spring Boot ve JPA kullanılarak oluşturulması ve Restful backend servislerinin tasarlanmasıdır. Aşağıda her bir metod için rehber bilgiler sunulmuştur:**

**Read Metodu:** Bu metod, "carId" bilgisini parametre olarak alır ve bir dizi nesne içeren bir yanıt döndürür. Bu nesneler, soruları ve ayrıntıları içerir. Eğer bu "carId" ile önceki bir ekspertiz yapılmışsa ve bu önceki ekspertizde "Evet, var" seçeneği işaretlenmişse, önceki ekspertizde eklenen fotoğrafların URL'leri nesnelerin içinde alt bilgi olarak döner. Eğer aynı araç için daha önce birden fazla ekspertiz yapıldıysa, bu araç için en son yapılan ekspertize bakmak yeterli olacaktır.

**Create Metodu**: Bu metod, Read metodunu kullanarak oluşturulan ekspertiz ekranında sorular yanıtlandıktan sonra "carId" bilgisini de ekleyerek çağrılır. Çekilen fotoğraflar otomatik olarak mobil uygulama tarafından buluta yüklenir ve yazılacak olan servise, kullanıcının seçtiği cevabı ve fotoğrafların URL bilgilerini içeren bir dizi string olarak iletilir.

Notlar:

Servis URL'leri, JSON mimarisi ve veritabanı tasarımının sizin tarafınızdan yapılması beklenmektedir. Yukarıdaki bilgilere dayalı olarak backend servislerinin oluşturulması ve mobil uygulamanın kullanımına sunulması gerekmektedir.

"Evet, var" seçeneğinin fotoğraf menülerini açma, fotoğraf çekme ve silme işlemleri mobil uygulamanın sorumluluğundadır.

Projenin unit testlerinin yazılması, log ve tracing eklemelerinin olması pozitif değerlendirilecektir.