



JAVA CHALLENGE

1- SAP Commerce (Hybris) nedir? Hangi amaçlarla kullanılır? Kullandığı teknolojiler nelerdir? Kısaca açıklayınız.

SAP Commerce e-ticaret platformları için geliştirilen bir yazılımdır. B2B, B2C gibi e-ticaret çözümleri sunar. Kullanıcı deneyimini kişiselleştirme, ürün verisi yönetimi gibi özellikler sunarak bir e-ticaret deneyimi sağlar. Kullandığı teknolojiler Java, Spring Framework, Restfull Api gibi teknolojiler kullanır.

2- Birbirinden bağımsız iki platformun birbiriyle haberleşmesi nasıl sağlanabilir? Örneğin, X platformu Java ile yazılmış olsun, Y platformu C# ile. Bu iki platformun birbiriyle iletişim halinde request-response ilişkisi kurması gerekiyor. Bu yapıyı nasıl sağlarız? Bu iletişim sırasında güvenlik nasıl sağlanır?

Farklı platformların birbirleriyle haberleşmesi için API tabanlı iletişim kullanılır. Her iki platformda API tanımlanır ve bu API'ler birbirlerine HTTP üzerinden veri gönderip alırlar. Bu yapıyla birbirlerinden bağımsız olmalarını ve farklı teknolojilerin birbirleriyle iletişim kurmalarını sağlar.

3- SOLR Nedir? Kullanım alanlarını araştırınız. Kurumsal bir projede kullanılabilecek iki farklı kullanım alanı örneği veriniz.

SOLR, açık kaynaklı bir arama platformudur. Veritabanları ve büyük veri sistemlerinde hızlı arama yapmak amacıyla kullanılır. Web Arama Motorlarında verileri hızlı bir şekilde aramak ve filtrelemek için ve E-Ticaret Sitelerinde ürünlerin hızlı ve etkili bir şekilde aranabilmesi için kullanılabilir.

4- Aşağıdaki algoritma için uygun çözümü üretin.

- Java'da 100 adet random sayıya sahip bir liste oluşturun.
- Daha sonra bu listenin bir kopyasını oluşturun.
- 0 ile 100 arasında rastgele bir sayı üretin.
- Kopya listedeki bu random sayının olduğu indisteski değeri silin.
- Şimdi elinizde iki adet liste var ve kopya listede orjinal listeye göre bir eleman eksik.
- Hangi elemanın eksik olduğunu bulan bir metot oluşturun.

```

public static List<Integer> randomList(int size) { 1 usage

    Random random = new Random();

    List<Integer> list = new ArrayList<>();

    for (int i = 0; i < size; i++) {
        list.add(random.nextInt( bound: 101));
    }
    return list;
}

```

Burada List tipinden bir ArrayList oluşturup 0 ile 100 arasındaki sayılardan bir random liste oluşturdum.

```

public static int findMissingNumber(List<Integer> originalList, List<Integer> copyList) { 1 usage

    Map<Integer, Integer> numberMap = new HashMap<>();

    for (int number : originalList) {
        numberMap.put(number, numberMap.getOrDefault(number, defaultValue: 0) + 1 );
    }

    for (int number : copyList) {
        if (numberMap.containsKey(number)) {
            numberMap.put(number, numberMap.get(number) - 1);
        }
    }

    for (Map.Entry<Integer, Integer> entry : numberMap.entrySet()) {
        if (entry.getValue() > 0) {
            return entry.getKey();
        }
    }

    return -1;
}

```

Burada Map tipinden bir HashMap oluşturdum. HashMap oluşturmamdaki amacım HashMap bir hash table kullanır bu nedenler verilere erişim , arama gibi işlemlerde $O(1)$ time complexity'e sahip olmasıdır. Metotta originalList'in içerisindeki her sayıyı listenin içerisinde kaç defa olduklarına göre izleyebilmek için HashMap'e ekledim bu sayılar key olarak depolandı ve her sayının kaç defa olduğunu görmek için value'lar kullandım. sonrada copyList'imizle kıyasladık. Eğer originalMap'te varsa bu ögenin valuesini 1 azaltıyoruz. Bu kontrol ile copyList'imiz ile originalList'imizdeki öğelerin eşleştiğini görüyoruz. Son olarak originalList'imizde 0 dan büyük bir öge arıyoruz.

```
public static void main(String[] args) {  
  
    List<Integer> originalList = randomList(size: 100);  
  
    List<Integer> copyList = new ArrayList<>(originalList);  
  
    System.out.println("Original List: " + originalList);  
  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.println("Please choose a number from the list :");  
  
    int chooseNumber = scanner.nextInt();  
  
    System.out.println("Choose Number : " + chooseNumber);  
  
    copyList.remove(Integer.valueOf(chooseNumber));  
  
    int missNumber = findMissingNumber(originalList, copyList);  
  
    System.out.println("Missing Number: " + missNumber);  
}
```

NOT: Lütfen internette gördüğünüz veya yapay zekadan oluşturduğunuz metotları kullanmayınız.

En hızlı çözümü üretememiş olsanız bile kendi oluşturduğunuz eşsiz ve mantıklı bir algoritma kullanmayı tercih ediniz.

