

سوالات در طول متن

سوال اول: اگر این اتفاق بیفتد متود `IndexOutOfBoundsException` میدهد.

سوال دوم: شماره آخرین شی 18 همیشه در واقع همه اشیا تو اون لیست شیفیت میخورن. بله این امکان وجود داره و اون شی میره ته لیست و همیشه آخرین شی که شماره اش همیشه 19.

سوال 3: چجوری با `while` پیمایش کنیم لیست رو)

با متود `size` تعداد اعضای اون لیست رو پیدا میکنیم و به این شکل:

```
while(index < list.size){  
    Do something;  
    index++  
}
```

سوال 4: چرا `iterator` جنریک است؟

چون که ما از قبل نمیدونیم اون لیستی که میخوایم روش پیمایش کنیم اشیا توش از چه کلاس اند واسه همین باید `generic` باشه تا ما خودمون دستی نوع اعضا رو تعیین کنیم.

سوال 5: اشکال زدایی کد

یک اینکه

دوم اینکه متود `contains` به این صورته که شما یک ابجکت بهش میدی و اون لیست اگر اون ابجکت رو داشت `true` برمیگردونه اگر نه `false` پس ازش نمیشه به این شکل استفاده کرد و برای انجام این کار باید از این دستور استفاده کرد و یکی اینکه وقتی که میخوایم روی لیست پیمایش کنیم و یک چیزی رو `remove` کنیم نباید از `for` استفاده کنیم بلکه از `iterator` باید استفاده کنیم. و اینکه باید تو کانستراکتور `arraylist` رو `new` کنیم.

شکل درست کد:

```
package Music;  
  
import java.util.ArrayList;  
import java.util.Iterator  
  
public class MusicOrganizer {  
  
    private ArrayList<String> tracks;  
  
    public MusicOrganizer
```

```

    {
        tracks = new ArrayList<String>();
    }

    public void removeTrack(String nameLike) {
        Iterator<String> it = tracks.iterator();
        while (it.hasNext()) {
            String t = it.next();
            if (t.equals(nameLike))
                tracks.remove(it);
        }
    }
}

```

پاسخ دهید:

1. تفاوت کد اول و دوم این است که در کد اول ابجکت های کلاس **Student s** سورس دارن ولی در کد دوم **anonymous object** داریم. **anonymous** یعنی بدون اسم و در واقع هیچ اسم و رفرنسی نداره.

2. کلاس درس که یک لیست داره از اشیا دانشجو ها و اسم استاد درس کلاس دانشجو که شامل اسم و فامیل و شماره دانشجوییش میشه و نمراتش در درس های مختلف 3. سه نمونه از کلاس های جاوا برای دسته بندی اشیا:

این کلاس اشیا رو با دو ایتم کلید و مقدار ذخیره میکنه که کاربردش برای وقتی که ما: **Hash map** میخوایم با استفاده از اون کلید به اون مقدار دسترسی داشته باشیم مثل دفترچه تلفن

که مثل مجموعه هاست و عضو تکراری نمیگیره و عضو نال هم میتونه داشته باشه: **Hash set** و ترتیب اشیا حفظ نمیشه و برای عملیات سرچ خیلی مناسبه

که اندازه اش ثابت نیست و برای زمان هایی مناسبه که تعداد دقیق اشیا رو نمیدونیم و: **Linked list** توش میشه اعضای تکراری داشت و ترتیب هم توش حفظ میشه

#### 4.APACHE POI