

«باسمه تعالی»

(پاسخ سوالات دستورکار جلسه سوم)

سوال: در متدهای remove و get با استفاده از index می‌توان یک شی را به دست آورد و یا از مجموعه حذف کرد. اگر در این متدها، عددی منفی یا بیشتر از تعداد عناصر آرایه داده شود، چه اتفاقی می‌افتد؟

با خطای `IndexOutOfBoundsException` مواجه می‌شویم.

برای مثال یک `ArrayList` با 2 عضو تعریف نمودم و خواستم که در خروجی عضو 3 را چاپ کند:



```
1 package com.company;
2
3 import java.util.ArrayList;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         ArrayList<String> myArray = new ArrayList<String>();
9         myArray.add("ali");
10        myArray.add("hosein");
11
12        System.out.println(myArray.get(4));
13    }
14 }
```

Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 4 out of bounds for length 2 <3 internal calls>
at java.base/java.util.Objects.checkIndex(Objects.java:373)
at java.base/java.util.ArrayList.get(ArrayList.java:425)
at com.company.Main.main(Main.java:12)

Process finished with exit code 1

سوال: اگر از مجموعه ۲۰ تایی عنصر دهم را حذف کنید، index آخرین شی برابر با چه عددی خواهد شد؟ آیا امکان اضافه کردن یک شی در بین اشیای دیگر یک `ArrayList` وجود دارد؟ در این حالت index آخرین شی چه تغییری می‌کند؟

اگر یک عضو از مجموعه ی 20 تایی حذف شود هیندکس سایر اعضا نیز آپدیت میشود و بنابراین Index آخرین شیء برابر با 18 میشود.

بله – به کمک متد add و با دادن دو ورودی : یکی Index خانه ی مورد نظر و دیگری آن شیء مدنظر میتوان آن را بین اشیای دیگر در ArrayList اضافه کرد.

اگر به مجموعه ی 20 تایی اولیه یک شیء اضافه شود آنگاه ایندکس آخرین عضو برابر با 20 میشود.

استفاده از while و index: چگونه با استفاده از while می توان یک مجموعه را پیمایش کرد؟

اگر یک collection داشته باشیم به کمک collection.size() می توان سایز یا به عبارتی تعداد المنت های آن collection را پیدا کرد و سپس خواهیم داشت:

```
int index = 0;
while (index < collection.size())
{
    body
    index++;
}
```

Iterator یک کلاس است که امکان حرکت روی عناصر یک collection را فراهم می کند. این کلاس در کتابخانه java.util تعریف شده است و یک کلاس generic است (چرا؟).

زیرا دارای قسمت پارامتر نوع میباشد که درون علامت <> قرار میگیرد و می تواند با آرگومان های ورودی از جنس داده های گوناگون صدا زده شود. کامپایلر براساس نوع آرگومان ارسال شده به تابع generic می تواند فراخوانی تابع مناسب را تشخیص دهد.

اشکال زدایی

قطعه کد زیر سه اشکال دارد. آنها را بیابید و فرم صحیح آن را بنویسید.

```
import java.util.ArrayList;

public class MusicOrganizer {
    private ArrayList<String> tracks;

    public void removeTrack(String nameLike) {
        for (int i = 0; i <= tracks.size(); i++)
            if(tracks.get(i).contains(nameLike))
                tracks.remove(i);
    }
}
```

آیا استفاده از دستوری مشابه `tracks.get(i).contains(nameLike)` مجاز است؟ (راهنمایی: chaining method calls)

مشکل اول: `new` نکردن `ArrayList` در سازنده ی کلاس .

مشکل دوم: مشکل از دست رفتن دیتا در پیمایش را داریم که با پیمایش از آخر یا استفاده از `Iterator` آن را حل کرد.

مشکل سوم: باید علامت مساوی در حلقه `for` حذف شود. چرا که باید تا `size-1` پیش برود.

فرم صحیح کد:

```
1 package com.company;
2 import java.util.ArrayList;
3
4 public class MusicOrganizer {
5     private ArrayList<String> tracks;
6     public MusicOrganizer()
7     {
8         tracks = new ArrayList<>();
9     }
10    public void removeTrack(String nameLike)
11    {
12        for (int i = tracks.size()-1 ; i>=-1; i--)
13            if(tracks.get(i).contains(nameLike))
14                tracks.remove(i);
15    }
16 }
17
```

بله استفاده از آن مجاز میباشد. برای مثال میتوانیم به شکل دیگر این کار را انجام دهیم و ابتدا قسمت اول را در یک متغیر بریزیم و بعد متد `contains()` را روی آن صدا بزنیم.

```
String myTrack = tracks.get(i);  
if( myTrack.contains(nameLike))  
    tracks.remove(i);
```

پاسخ دهید

۱- تفاوت این دو قطعه کد در چیست؟ (راهنمایی: anonymous objects)

قطعه کد اول

```
ArrayList<Student> studentList = new ArrayList<Student>(2);  
Student std1 = new Student("seyed", 9031806);  
studentList.add(std1);  
Student std2 = new Student("saleh", 9131089);  
stringList.add(std2);
```

قطعه کد دوم

```
ArrayList<Student> studentList = new ArrayList<Student>(2);  
studentList.add(new Student("seyed", 9031806));  
stringList.add(new Student("saleh", 9131089));
```

۲- می‌خواهیم سیستمی برای ذخیره و بازیابی اطلاعات دانشجویان و نمرات آن‌ها در درس‌های مختلف طراحی کنیم که اساتید هر درس به این سیستم دسترسی دارند. برای این سیستم چه کلاس‌هایی تعریف می‌کنید؟

۳- سه نمونه از کلاس‌های جاوا برای دسته‌بندی اشیا به همراه کاربرد آن‌ها ذکر کنید.

۴- یک کتابخانه جاوا برای خواندن فایل‌های excel (با فرمت xlsx) پیدا کنید.

جواب سوال اول:

قطعه کد دوم تفاوتی در اجرای برنامه با قطعه کد اول ندارد ولی مانند اولی اسم `std1` و `std2` نداریم و اگر بخواهیم دوباره از آبجکت‌ها استفاده کنیم نمیتوانیم.

جواب سوال دوم:

کلاس هایی نظیر:

Teacher

System

Course

Student

جواب سوال سوم:

AbstractList: AbstractCollection را گسترش داده و بیشترین بخش از اینترفیس List را اجرا میکند.

ArrayList: با گسترش AbstractList یک ردیف پویا را اجرا می کند.

LinkedList: یک لیست لینک شده را با گسترش AbstractSequentialList اجرا می کند.

TreeSet: یک مجموعه ی ذخیره شده در یک درخت را اجرا میکند AbstractSet. را گسترش میدهد.

HashSet: AbstractSet را برای استفاده با یک جدول hash گسترش می دهد.

جواب سوال چهار:

apache.poi