

به نام خدا

محمد رضا حسن زاده

۹۷۳۱۷۰۳

تمارین برنامه نویسی پیشرفته

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؟

چون اندیس‌ها خارج از بازه‌ی بین صفر تا سائیزشان منهای یک نیستند. در نتیجه به ارور `IndexOutOfBoundsException` بر می‌خوریم.

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

اگر از یک مجموعه‌ی ۲۰ تایی عنصر دهم را حذف کنیم اندیس آخرین عنصر برابر با ۱۸ می‌شود. امکان اضافه کردن در بین اشیای دیگر هم وجود دارد می‌توان با استفاده از متد `add` و با وارد کردن دو پارامتر می‌توان بین آنها نیز یک شی‌ای را وارد کرد. در این حالت اندیس آخرین شی برابر با ۲۰ می‌شود.

استفاده از `while` و `index`: چگونه با استفاده از `while` می‌توان یک مجموعه را پیمایش کرد؟

می‌توان به وسیله‌ی قطعه کد زیر این کار را انجام داد.

```
int temp = 0;
while(temp < collection.size())
{
    //body of the loop
    temp++;
}
```

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

علت این است که به هنگام استفاده از این کلاس باید در بعد از Iterator از <> استفاده کنیم و تایپ متغیری را که می‌خواهیم iterate کنیم را مشخص کنیم. خود کامپایلر با استفاده از نوع ورودی‌ای که در داخل <> قرار می‌دهیم خودش تابع مناسب را فراخوانی می‌کند و با آن‌ها استفاده می‌کند.

اشکال زدایی

۱- برای متغیر tracks باید یک کانستراکتور تعریف شود

۲- در داخل حلقه‌ی for قسمت شرطش باید اصلاح شود چراکه متغیر i نمیتواند تا خود سایز کالکشن برود. همیشه بیشترین اندیس، یکی از تعداد سایز کالکشن‌ها کمتر است.

۳- در اینجا پیمایش از اول به آخر ایراد دارد چراکه اگر که متد remove اجرا شود اندیس‌ها همگی یکی کم می‌شود و امکان دارد این بین‌یه سری مشکلات در داده‌ها پیش بیاید. برای رفع این مشکل از آخری به اول پیمایش می‌کنیم که این مشکل پیش نیاید.

```
import java.util.ArrayList;

public class MusicOrganizer {
    private ArrayList<String> tracks = new ArrayList<String>();

    public void removeTrack(String nameLike){
        for(int i = tracks.size() - 1; i >= 0; i--){
            if(tracks.get(i).contains(nameLike)){
                tracks.remove(i);
            }
        }
    }
}
```

و همچنین مجاز هستیم که از فراخوانی زنجیره‌ای استفاده کنیم.

پاسخ دهید

۱- در اجرا این دوتا کد با یکدیگر تفاوتی ندارند اما در قطعه کد اول آبجکتی که ساخته شده داخل یک متغیر ذخیره شده است و می‌توان از آن در ادامه‌ی کد استفاده کنیم اما در قطعه کد دوم به این گونه نیست و نمی‌توان از آن در ادامه‌ی کد استفاده کرد.

۲-

- ۱. Teachers
- ۲. Students
- ۳. Courses
- ۴. InformationGathering

۳-

- ۱. Hashset: این کلاس Set را پیاده سازی کرده است به وسیله‌ی یک جدول hash آن پیاده سازی شده است. این کالکشن به مانند یک مجموعه در ریاضیات عمل می‌کند.
- ۲. ArrayList: این کلاس در جاوا به صورت داینامیک می‌تواند اطلاعات را ذخیره بکند و همچنین سایز آن ها رشد می‌کند با افزایش عناصر.
- ۳. Hashmap: این کلاس جنریک مانند یک ArrayList می‌باشد اما این گونه که به جای اینکه اندیس ها به صورت خودکار از صفر تا سایز منهای یک بروند، ما برایش اندیس تعریف می‌کنیم که چه چیزی باشد. مثلاً برای دفترچه تلفن می‌توان از این کلاس استفاده کرد، کاربرد این کلاس جایبست که می‌خواهیم برای هر داده‌ای یک اندیس خاصی تعریف کنیم و به وسیله‌ی آن اندیس به داده‌ی داخل آن عنصر دسترسی پیدا کنیم.

۴- Apache POI