

❖ برای دانلود گرامر زبان جاوا در انتلر از این [پیوست](#) استفاده کنید.

Listener Question 1

در زبان جاوا دارای سطوح دسترسی متفاوتی از جمله `public, private, protected` هستند. با استفاده از زبان پایتون و ابزار انتلر برنامه ای بنویسید که با گرفتن آدرس یک فایل جاوا، برای هر یک از کلاس های داخل این برنامه، با توجه به سطوح دسترسی متفاوت موجود، لیست توابع با آن سطح دسترسی را نمایش دهد. (خروجی را برای یک ورودی نمایش دهید).

Listener Question 2

برای بررسی و بهبود کارایی کدها نیاز است تا پیچیدگی زمانی اجرای توابع را بدست آوریم. با استفاده از زبان پایتون و ابزار انتلر برنامه ای بنویسید که با گرفتن آدرس یک برنامه جاوا برای هر یک از توابع موجود در این برنامه، حداکثر تعداد حلقه های تو در تو (`for, while`) در این برنامه را نمایش دهد.

- به عنوان مثال در برنامه نوشته شده در شکل ۱، تنها یک حلقه `for` داریم که داخل آن حلقه دیگری قرار ندارد. پس حداکثر عمق حلقه های تو در تو برای تابع `add` برابر ۱ است.

```
1 public class Calculator{
2     public int add(int[] arr){
3         int sum=0;
4         for(int i=0;i<arr.length;++i){
5             if(i<5){
6                 sum+=arr[i];
7             }
8         }
9         return sum;
10    }
11 }
```

شکل 1: نمونه کد جاوا

Listener Question 3

با استفاده از زبان پایتون و ابزار انتلر برنامه ای بنویسید که ورودی آن آدرس يك فایل C++ و خروجی آن نام کلاس های موجود در فایل ورودی و تعداد آن ها باشد.

- ❖ برای دانلود گرامر زبان جاوا در انتلر از این [پیوست](#) استفاده کنید.
- ❖ برای دانلود گرامر زبان سی پلاس پلاس در انتلر از این [پیوست](#) استفاده کنید.
- ❖ سوالات 1 تا 3 برای گرامر زبان جاوا و سوال 4 برای گرامر زبان سی پلاس پلاس طرح شده است.

Visitor Question 1

با استفاده از زبان پایتون و ابزار انتلر برنامه ای بنویسید که برای فیلدهای خصوصی یک کلاس مانند مثال زیر، یک متد getter و setter با نام های <field_name> و <field_name> set به کلاس اضافه کند. این دو متد از نوع public هستند. همچنین، اگر فیلد خصوصی با کلیدواژه final تعریف شده بود (متغیر از نوع constant بود)، فقط متد getter اضافه شود.

مثال:

Before:

```
public class Sample {  
    private int field;  
    private final int constantField;  
}
```

After:

```
public class Sample {  
    private int field;  
    private final int constantField;  
  
    public setField(int value)  
    {  
        field = value;  
    }  
    public int getField()  
    {  
        return field;  
    }  
  
    public int getConstantField()  
    {  
        return constantField;  
    }  
}
```