تمرین سوم درس اصول طراحی کامپایلر

استاد: دکتر سعید پارسا

طراح و مصحح: نیما کمبرانی و عماد وثوق

موضوعات تمرین:

- گرامر های LL1
- Reverse در انتلر

❖ **برای دانلود گرامر زبان جاوا در انتلر از این پیوست استفاده کنید.**

سوال ۱:

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، باتوجه به سطوح دسترسی متفاوت موجود، لیست توابع با آن سطح دسترسی را نمایش دهد.(خروجی را برای یک ورودی نمایش دهید.)

سوال ۲:

برای بررسی و بهبود کارایی کد ها نیاز است تا پیچیدگی زمانی اجرای توابع را بدست آوریم. با استفاده از زبان پایتون و ابزار انتلر برنامه‌ای 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 تودرتو برای تابع add برابر ۱ است.**

```java
public class Calculator{
    public int add(int[] arr){
        int sum=0;
        for(int i=0;i<arr.length;++i){
            if(i<5){
                sum+=arr[i];
            }
        }
        return sum;
    }
}
```

شکل 1. نمونه کد جاوا

سوال ۳:

برای هر یک از گرامر های زیر با مشخص کردن از مجموعه های first و follow برای آن، LL(1) بودن یا نبودن گرامر را مشخص کنید. درصورتی که گرامر LL(1) نیست، گرامر LL(1) معادل آن را بدست آوردید. در نهایت جدول تجزیه هر یک از گرامرها را مشخص کنید.

1. **S → AdS | Ba**

   A → aB | Ab

   B → Bd | λ

2.

S → A

A → xAx | C

B → yBy | C

C → zBz | wAw | λ

سوال۴:

برای گرامر زیر مجموعه first , follow را بدست آورده و با توجه به آن جدول تجزیه گرامر را رسم کنید. سپس مراحل تجزیه رشته id * (id + id) را با توجه به جدول تجزیه نشان دهید و در نهایت درخت تجزیه رشته را رسم کنید.

E --> TE'

E' --> +TE' | λ

T --> FT'

T' --> *FT' | λ

F --> id | (E)