**پروژه ۴ کامپایلر**

هدف از پروژه‌های کامپایلر، پیاده‌سازی اجزاء مختلف یک کامپایلر واقعی برای نسخة محدود شده‌ای از زبان جاوا (که به آن زبان **میوا** -مخفف مینیمال جاوا- می‌گوییم) می‌باشد. سینتکس زبان میوا کاملا مشابه با زبان جاوا می‌باشد که توضیح آن به دقت در صورت سوال پروژه قبلی آورده شده است.

**پروژه چهارم** در مورد پیاده سازی قسمت بررسی اسکوپ از فاز سمنتیک برای زبان میوا می‌باشد. البته فرض بر آنست که شما روال ساخت درخت پارس را در پروژه سوم با موفقیت پیاده‌سازی نموده اید. همانطور که در کلاس تشریح شد، برای پیاده سازی بررسی اسکوپ نیاز به پیمایش درخت پارس و همینطور پیاده سازی جدول سمبل دارید. وظیفه جدول سمبل نگهداری متغیرهای تعریف شده در برنامه به همراه نوع آنها می‌باشد. یعنی در هر مرحله از پیمایش گره‌های درخت پارس، استفاده از متغیرها را با استفاده از ۵ تابع جدول سمبل (که آنرا در کلاس توضیح داده‌ایم) بررسی می‌کنید و در صورت وجود خطا آنرا اعلام می‌نمائید. در صورت تعریف متغیر نیز بعد از بررسی صحت آن، جدول سمبل را آپدیت می‌کنید. جدول سمبل و توابع آن پیاده‌سازی شده‌اند، شما باید صرفا از آنها استفاده کنید تا روال بررسی اسکوپ را در هر گره از درخت پارس انجام دهید. در صورتیکه در هر مرحله خطائی از نظر اسکوپ وجود داشته باشد آنرا اعلام نمایید. برنامه شما باید کل خطاهای اسکوپ را شناسائی نماید،‌ یعنی بعد از مواجهه با یک خطا، روال بررسی اسکوپ باید ادامه پیدا کند. فرض بر آنست که برنامه ورودی فاقد خطاهای لغوی و پارسی است، بنابراین صرفا خطاهای سمنتیک (از نظر اسکوپ) را بررسی کنید.

فرض بر آنست که گرامر زیر در میوا قابل قبول نیست (یعنی تعریف متغیر در داخل دو آکلاد):

{

int a = 0;

}

ولی کد زیر در میوا قابل قبول است (یعنی تعریف متغیر در بدنه if):

if (true) {

int a = 0;

}

کلاس اصلی پروژه چهارم کلاسهای نوشته شده در فایل SymbolTable.java و مخصوصا کلاس SymbolTable است که البته نیازی به تغییر دادن آن ندارید. برای کار با آن، صرفا یک نمونه از این کلاس ایجاد نموده و توابع آنرا برحسب روالی که در کلاس تشریح شده (در هنگام پیمایش درخت پارس) فراخوانی نمایید.