**پروژه۳ 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 سوم** در مورد پیاده سازی کامل پارسر به همراه تولید درخت پارس برای زبان میوا می‌باشد. در پروژه دوم، پیاده‌سازی قسمت اولیه پارسر انجام شد که در آن صرفا درستی و یا عدم درستی برنامه از نظر قواعد گرامری زبان اعلام می‌شد. در پروژه سوم، باید کد پروژه قبلی را به نحوی تغییر دهید که اضافه بر بررسی صحت و سقم برنامه از نظر قواعد گرامری، درخت پارس نیز ساخته شود (البته فقط در صورتیکه برنامه خطایی از نظر گرامر زبان نداشته باشد).

برنامه شما یک فایل را از ورودی دریافت کرده و در صورت درست بودن فایل از نظر تحلیل‌گر لغوی، توکن‌های موجود در فایل را بازمی‌گرداند. سپس پارسر لیست توکن‌ها را دریافت نموده و **الگوریتم نزولی بازگشتی** را بر روی آن اجرا می‌کند. در حین پارس برنامه، بایستی درخت پارس طبق فرمت تعریف شده توسط آقای زارع زاده (کلاس ASTNode که در سامیاد آپلود شده است) ایجاد شده و در نهایت، به عنوان خروجی چاپ شود. در صورتیکه برنامه از نظر تحلیلگر لغوی و یا قواعد گرامری خطایی داشت نیازی به تولید درخت پارس نیست و صرفا اعلام خطا (به همراه خط مربوطه) کفایت می‌کند.

**نکته مهم:** کد خود را فقط و فقط در داخل فایلهای نمونه ارسال شده به گروه (توسط آقای زارع زاده) بنویسید. البته میتوانید بسته به نیاز خود فایلها، کلاسها و یا توابع دیگری نیز به فایلهای نمونه اضافه کنید. کلاس اصلی پروژه سوم کلاس Parser و همینطور کلاس ASTNode می‌باشد. وظیفه اصلی شما تغییر همان توابع نوشته شده در پروژه دوم است به نحویکه به همراه پارس، درخت پارس نیز به تدریج کامل شود. در نهایت، برای پرینت درخت می‌توانید از تابع print در کلاس ASTNode استفاده کنید.