



پروژه درس اصول طراحی کامپایلر (فاز سه)

دکتر امین طوسی

دستیاران آموزشی: نجمه حبیبی، فائزه شایسته

مقدمه

در فاز قبل پروژه به تکمیل جدول علایم و ذخیره موارد مورد نیاز به صورت سازماندهی شده پرداختیم. در فاز سوم پروژه می خواهیم با استفاده از جدول علایم به بررسی خطاهای موجود در برنامه بپردازیم. خطاهای لغوی و نحوی در فاز قبل پروژه بررسی شده اند بنابراین ما در این فاز خطاهای معنایی را بررسی خواهیم کرد.

بررسی خطا

فرمت گزارش خطا:

خطاهای موجود در برنامه را بر اساس فرمت زیر گزارش دهید:

line شماره خط ارور و column پوزیشن آن را در یک خط نشان میدهد.

- تعریف دوباره یک کلاس:

```
Error101 : in line [line:column], class [name] has been defined already
```

- تعریف دوباره متد در یک کلاس:

```
Error102 : in line [line:column] , method [name] has been defined already
```

- تعریف دوباره یک متغیر محلی در یک حوزه:

```
Error103 : in line [line:column], var [name] has been defined already
```

- تعریف دوباره یک خصیه در یک کلاس

```
Error104 : in line [line:column], field [name] has been defined already
```

نکته: دو نوع متفاوت میتوانند هم نام باشند به عنوان مثال اگر یک فیلد و متد هم اسم باشند مشکلی نیست.

نکته: در صورت تعریف دوباره یک کلاس، متد، فیلد یا لوکال اسم آن را عوض میکنیم و به سیمبل تبیل اضافه میکنیم و اسم آن را به این صورت ذخیره میکنیم: `name_line_column`

بعنوان مثال اگر متغیر `d` دوباره تعریف شود آن را به صورت `d_34_48` ذخیره می نماییم

نکته: هر کدام از موارد ذکر شده اگر دوبار تعریف شوند مورد دوم مطرح نیست و فرض میکنیم اصلا وجود ندارد و تنها از مورد اول استفاده می شود.

به عنوان مثال اگر یک کلاس دوبار تعریف شده باشد تنها میتوان از کلاس اول استفاده کرد.

- استفاده از یک کلاس تعریف نشده در یک حوزه :

```
Error105 : in line [line:column], cannot find class [className]
```

- استفاده از یک متغیر یا خصیه تعریف نشده :

```
Error106 : in line [line:column], Can not find Variable [name]
```

- عدم تطابق نوع بازگشتی متد با نوع بازگشتی تعریف شده توسط متد:

```
Error210 : in line [line:column], Return type of this method must be [MethodReturn type]
```

- تعداد و نوع پارامترهای متد در هنگام فراخوانی با تعداد پارامترهای رسمی در هنگام تعریف برابر نباشد:

```
Error220: in line [line:column], Mismatch arguments.
```

- در دستورات انتساب نوع عملوند چپ و راست یکی نباشد:

```
Error 230 : in line [line:column], Incompatible types : [LeftType] can not be converted to [RightType]
```

- وجود دور در ارث بری :

```
Error410 : Invalid inheritance [classname1] -> [classname2] -> [classname3] ...
```

نکات تکمیلی:

- می توانید پروژه را در گروه های حداکثر دو نفری انجام دهید.
- پروژه را حتما با فرمت stuName1-stuNum1-stuName2-stuNum2-phase1.zip اپلود کنید.
- توجه شود در صورت اثبات تقلب در پروژه، نمره طرفین 100- ثبت خواهد شد

موارد تحویلی:

- کد های مربوط به پروژه

موفق باشید