



طراحی کامپایلرها

نیم سال اول ۹۶-۹۷

مدرس: دکتر قاسم ثانی

دستیاران آموزشی: مریم غلامعلی تبار - رومینا جعفریان

پروژه پایانی

پروژه عملی درس شامل پیاده سازی یک کامپایلر برای زبان مینی جاوا (زیرمجموعه ای از زبان جاوا) است. این زبان شی گرا است ولی در این پروژه، پشتیبانی قسمت بسیار محدودی از ویژگی های شی گرایی در نظر گرفته شده است. بنابراین گرامر زبان مینی جاوا با ساده سازی بسیار، مبنای کار پروژه قرار گرفته که توضیحات کامل آن در ادامه آمده است.

مشخصات کامپایلر

- کامپایلر تک گذره است و از ۵ جزء تشکیل شده است:

بخش	اجزاء	نمره	توضیح
۱	تحلیل گر لغوی (اسکنر)	۰,۵	
۲	تحلیل گر نحوی (پارسر)	۰,۷۵	به روش LL(۱) - در صورت استفاده از روش های دیگر، هیچ نمره ای به پروژه تعلق نمی گیرد.
	خطا پرداز		به روش panic mode
۳	تحلیل گر معنایی	۰,۷۵	
	مولد کد میانی	۲	در قالب کدهای ۳ آدرس (در غیر این صورت نمره ای نخواهد داشت).

- اخذ نمره ی هر بخش منوط به پیاده سازی بخش های قبل از آن است.
- استفاده از کدهای موجود در مرجع درس یا سایر کتب کامپایلر، در صورت تسلط بر آن کد و اعلام مأخذ اشکالی ندارد.
- استفاده از برنامه های مولد اسکنر، پارسر یا تولید کد میانی و همچنین کدها و برنامه های موجود در سایت ها و کدهای سایر گروه ها (در همین نیم سال یا در سال های گذشته) اکیدا ممنوع است و جریمه ای به دنبال خواهد داشت که ممکن است سبب مردودی در درس شود. (تفاوتی بین گروه دهنده و گروه گیرنده کد نیست).
- برای تولید جدول پارس می توانید از برنامه ی JFLAP که در کوئرا بارگذاری شده استفاده کنید. (توجه کنید که تنها برای تولید جدول پارس می توانید از این برنامه استفاده کنید، نه پارسر).
- گرامر داده شده بایستی با تغییراتی از قبیل حذف چپ گردی صریح، فاکتورگیری از چپ و غیره به فرم قابل استفاده در روش پارس LL(۱) تبدیل شود.

- ورودی کامپایلر، یک فایل متنی حاوی برنامه‌ای است که کامپایلر شما باید آن را ترجمه کند.
- خروجی کامپایلر، یک فایل متنی حاوی کد میانی تولیدشده است.
- کامپایلر شما نیاز نیست از فراخوانی‌های بازگشتی پشتیبانی کند.
- در نسخه‌ی ساده‌شده‌ی مینی‌جاوا، فرض کنید که هیچ اشاره‌ی رو به جلویی رخ نخواهد داد؛ چرا که در این صورت کامپایلر تک‌گذره نخواهد شد.

گرامر MiniJava

توجه کنید که پایانه‌ها پررنگ‌تر از غیرپایانه‌ها نمایش داده شده‌اند.

1. $\text{Goal} \rightarrow \text{Source } \mathbf{EOF}$
2. $\text{Source} \rightarrow \text{ClassDeclarations } \mathbf{MainClass}$
3. $\text{MainClass} \rightarrow \mathbf{public\ class\ Identifier\ \{ \ public\ static\ void\ main\ ()\ \{ \ VarDeclarations\ Statements\ \}}$
4. $\text{ClassDeclarations} \rightarrow \text{ClassDeclaration } \text{ClassDeclarations} \mid \epsilon$
5. $\text{ClassDeclaration} \rightarrow \mathbf{class\ Identifier\ Extension\ \{ \ FieldDeclarations\ MethodDeclarations\ \}}$
6. $\text{Extension} \rightarrow \mathbf{extends\ Identifier} \mid \epsilon$
7. $\text{FieldDeclarations} \rightarrow \text{FieldDeclaration } \text{FieldDeclarations} \mid \epsilon$
8. $\text{FieldDeclaration} \rightarrow \mathbf{static\ Type\ Identifier\ ;}$
9. $\text{VarDeclarations} \rightarrow \text{VarDeclaration } \text{VarDeclarations} \mid \epsilon$
10. $\text{VarDeclaration} \rightarrow \text{Type } \mathbf{Identifier\ ;}$
11. $\text{MethodDeclarations} \rightarrow \text{MethodDeclaration } \text{MethodDeclarations} \mid \epsilon$
12. $\text{MethodDeclaration} \rightarrow \mathbf{public\ static\ Type\ Identifier\ (\ Parameters \)\ \{ \ VarDeclarations\ Statements\ return\ GenExpression\ ; \}}$
13. $\text{Parameters} \rightarrow \text{Type } \mathbf{Identifier\ Parameter} \mid \epsilon$
14. $\text{Parameter} \rightarrow \mathbf{, \ Type\ Identifier\ Parameter} \mid \epsilon$

15. $\text{Type} \rightarrow \mathbf{boolean} \mid \mathbf{int}$
16. $\text{Statements} \rightarrow \text{Statements Statement} \mid \epsilon$
17. $\text{Statement} \rightarrow \{ \text{Statements} \} \mid \mathbf{if} (\text{GenExpression}) \text{Statement} \mathbf{else} \text{Statement} \mid$
 $\mathbf{while} (\text{GenExpression}) \text{Statement} \mid \mathbf{for} (\text{Identifier} = \text{Integer} ; \text{RelTerm} ; \text{Identifier}$
 $\mathbf{+= Integer}) \text{Statement} \mid \text{Identifier} = \text{GenExpression} ; \mid \mathbf{System.out.println} ($
 $\text{GenExpression}) ;$
18. $\text{GenExpression} \rightarrow \text{Expression} \mid \text{RelExpression}$
19. $\text{Expression} \rightarrow \text{Expression} + \text{Term} \mid \text{Expression} - \text{Term} \mid \text{Term}$
20. $\text{Term} \rightarrow \text{Term} * \text{Factor} \mid \text{Factor}$
21. $\text{Factor} \rightarrow (\text{Expression}) \mid \text{Identifier} \mid \text{Identifier} . \text{Identifier} \mid \text{Identifier} . \text{Identifier} ($
 $\text{Arguments}) \mid \mathbf{true} \mid \mathbf{false} \mid \text{Integer}$
22. $\text{RelExpression} \rightarrow \text{RelExpression} \ \&\& \ \text{RelTerm} \mid \text{RelTerm}$
23. $\text{RelTerm} \rightarrow \text{Expression} == \text{Expression} \mid \text{Expression} < \text{Expression}$
24. $\text{Arguments} \rightarrow \text{GenExpression Argument} \mid \epsilon$
25. $\text{Argument} \rightarrow , \text{GenExpression Argument} \mid \epsilon$
26. $\text{Identifier} \rightarrow \mathbf{identifier}$
27. $\text{Integer} \rightarrow \mathbf{integer}$

فهرست دستورالعمل‌های سه آدرس قابل استفاده برای تولید کد میانی

توضیح	قالب کد سه آدرس
عملوندهای اول و دوم جمع می‌شوند و حاصل در D قرار می‌گیرد.	(ADD, S1, S2, D)
عملوند دوم از عملوند اول کم می‌شود و حاصل در D قرار می‌گیرد.	(SUB, S1, S2, D)
عملوندهای اول و دوم AND می‌شوند و حاصل در D قرار می‌گیرد.	(AND, S1, S2, D)
محتوای S در D قرار می‌گیرد.	(ASSIGN, S, D,)
اگر S1 و S2 مساوی باشند، در D مقدار true و در غیر این صورت، مقدار false ذخیره می‌شود.	(EQ, S1, S2, D)
محتوای S بررسی می‌شود و در صورتی که false باشد، کنترل به L منتقل می‌شود.	(JPF, S, L,)
کنترل به L منتقل می‌شود.	(JP, L, ,)
اگر S1 کوچکتر از S2 باشد، در D مقدار true و در غیر این صورت، مقدار false ذخیره می‌شود.	(LT, S1, S2, D)
عملوند اول در عملوند دوم ضرب می‌شود و حاصل در D قرار می‌گیرد.	(MULT, S1, S2, D)
نقیض محتوای عملوند S در D قرار می‌گیرد.	(NOT, S, D,)
محتوای S بر روی صفحه چاپ می‌شود.	(PRINT, S, ,)

- از روش‌های نشانی‌دهی^۱ مستقیم (مانند t)، غیر مستقیم (مانند @t) یا مقدار صریح (مانند #5) می‌توانید در کد میانی استفاده کنید. توجه کنید که در خروجی نهایی باید به جای t، نشانی مکان این متغیر در حافظه قرار گیرد.
- برای سادگی فرض کنید آدرس متغیرها به صورت ایستا^۲ تخصیص می‌یابد.
- میزان فضای در نظر گرفته شده برای هریک از متغیرها (مانند t) ۴ بایت است.
- دستور PRINT تنها محتوای از گونه عدد صحیح را چاپ می‌کند.

ملاحظات لغوی

- کامنت‌ها همچون زبان جاوا به دو صورت `/* Comment */` و `// Comment` هستند و می‌توانند پس از هر توکنی بیایند.
- کلیدواژه‌ها^۳ رزرو شده هستند و نمی‌توانند به عنوان شناسه استفاده شوند.
- تعریف توکن‌های identifier و integer به صورت زیر است:

$\text{letter} \leftarrow [\text{A-Za-z}]$

$\text{digit} \leftarrow [0-9]$

$\text{identifier} \leftarrow \text{letter} (\text{letter} \mid \text{digit})^*$

$\text{integer} \leftarrow (+|-|\epsilon) (\text{digit})^+$

نمونه‌ی ورودی و خروجی

توجه کنید که خروجی کامپایلر شما باید کاملاً مطابق قالب فوق باشد زیرا صحت خروجی کد شما توسط مفسر کد میانی بررسی می‌شود. (مفسر کد میانی اواخر ترم در کوئرا بارگذاری خواهد شد.)

نمونه‌ی ورودی

```
// test case 1
class Cls{
    static boolean d ;
    public static int test( int a , boolean b ) {
        int c ;
        c = 1 ;
        b = true ;
        return a + c ;
    }
}

public class Cls2 {
    public static void main( ) {
        int b;
        b = 3;
        b = Cls.test( b, false );
        System.out.println(b);
    }
}

EOF
```

نمونه‌ی خروجی

توجه کنید که خانه‌های بلوک برنامه از صفر شروع می‌شود.

```
0 (JP, 5, , )
1 (ASSIGN, #1, 516, )
2 (ASSIGN, true, 512, )
3 (ADD, 508, 516, 1004)
4 (JP, @2000, , )
5 (ASSIGN, #3, 520, )
6 (ASSIGN, 520, 508, )
7 (ASSIGN, false, 512, )
8 (ASSIGN, #10, 2000, )
9 (JP, 1, , )
10 (ASSIGN, 1004, 520, )
11 (PRINT, 520, , )
```

انجام پروژه

- پروژه باید به صورت انفرادی و یا گروه‌های دو نفره انجام شود.
- در صورتی که هر گونه سوالی در رابطه با تعریف پروژه دارید، آن را در کوئرا مطرح نمایید.
- کد پروژه باید در قالب یک فایل با پسوند java. یا cpp. یا py. در کوئرا بارگذاری شود. (پروژه را با هر زبان برنامه‌نویسی دیگری نیز می‌توانید انجام دهید، ولی در نهایت بایستی سورس کد پروژه در قالب یک فایل متنی واحد با پسوند متناسب با زبان استفاده شده بارگذاری شود.)
- مهلت بارگذاری سورس کد پروژه، ساعت ۸:۰۰ روز شنبه ۷ بهمن است.
- تحویل حضوری روز ۱۰ بهمن است که زمان‌بندی دقیق آن متعاقبا اعلام خواهد شد. (توجه کنید که حضور هر دو عضو گروه‌های دو نفره در جلسه‌ی تحویل الزامی است.)

موفق باشید.