

به نام خدا

## ۱- اهداف پروژه

هدف از این پروژه، پیاده‌سازی یک کامپایلر برای زبان برنامه‌نویسی ساده شده C است. کامپایلر مورد نظر می‌بایست با کمک ابزار ANTLR (Another Tool for Language Recognition) پیاده‌سازی شود.

- ورودی کامپایلر، فایل کد برنامه مبدا است که یک برنامه نوشته شده به زبان برنامه‌نویسی ساده شده C می‌باشد.
  - خروجی کامپایلر:
    - در صورت وجود خطای برنامه‌نویسی در کد برنامه مبدا، خطاها تشخیص داده شده و پیغام‌های مناسب برای هر خطا در خروجی چاپ می‌شود.
    - در صورت عدم وجود خطای برنامه‌نویسی در کد برنامه مبدا، خروجی کامپایلر یک فایل است که شامل کد میانی (دستورالعمل‌های سه آدرس) معادل با کد برنامه مبدا می‌باشد.
- مراحل پیاده‌سازی پروژه در دو فاز انجام می‌شود که در بخش ۴ به توضیح آن می‌پردازیم.

## ۲- نرم افزارها و ابزارهای مورد نیاز

- Java 7 Development Kit (JDK 7)
- Microsoft Visual Studio (2010 or newer)
- Microsoft Visual Studio 2010 (or newer) ANTLR Language Support extension (version 4)
- ANTLRWorks 2

## ۳- زبان برنامه نویسی ساده شده C

در این پروژه کد برنامه مبدا به زبان برنامه نویسی ساده شده C است. قواعد نحوی و معنایی این زبان، به شرح زیر می‌باشد:

### ۳-۱- قواعد نحوی

- این زبان شامل دستورات محاسباتی و انتساب، شرطی، while ، for (بدون اعلان متغیر شمارنده)، دستور خروجی printf با یک پارامتر ورودی و توضیحات (comment) می‌باشد.
- هر برنامه باید شامل یک تابع main باشد. void main() { ... }
- سایر توابع، دستورات ورودی و خروجی، آرایه‌ها، ساختارها، فایل‌ها و ... توسط گرامر زبان پشتیبانی نمی‌شوند.
- دستورات اعلان متغیرها نباید شامل عملگرهای انتساب و محاسباتی باشد.
- دستورات شرطی و تکرار باید ساختارهای تو در تو را پشتیبانی کنند.
- عملگرهای مجاز عبارتند از: = ، + ، - ، \* ، / ، ( ) ، ++ ، -- ، < ، <= ، > ، >= ، = و != .
- انواع داده مجاز: int (چهار بایت) و char (۱ بایت).

مثال: الگوریتم محاسبه مجموع اعداد ۱ تا ۱۰

```
void main()
{
    // compute the sum of numbers from 1 to 10
    int n,i,x;
    n=10;
    x=0;
    for(i=1;i<=n;i++)
    {
        x=x+i;
    }
    printf(x);
}
```

### ۳-۲- قواعد معنایی

- در دستورات، از متغیرهایی می‌توان استفاده کرد که از قبل اعلان (تعریف) شده باشند.
- یکتایی نام متغیرها در داخل یک بلاک {int x;int x;} نادرست {int x;int x;} درست
- سازگار بودن نوع متغیرها در دستورات محاسباتی و رابطه‌ای.

### ۴- فازهای انجام پروژه

#### ۴-۱- فاز اول: تولید گرامر و درخت تجزیه با استفاده از نرم افزار ANTLRWorks 2

نرم افزار ANTLRWorks2 یک محیط برنامه نویسی برای تولید گرامرهای ANTLR است. در فاز اول، می‌بایست گرامر زبان ساده شده C را در نرم افزار ANTLRWorks2 با استفاده از دستور زبان ANTLR 4 بنویسید و آن را با نام C و با پسوند g4. (به صورت c.g4) ذخیره کنید. سپس با استفاده از گزینه Generate Recognizer ... که در منوی Run قرار دارد، کدهای لکسر و پارسر را به طور خودکار برای گرامر c.g4 تولید کنید. پس از تولید کدهای لکسر و پارسر که به زبان جاوا نوشته شده‌اند، می‌توانید صحت گرامر نوشته شده را مورد آزمایش قرار دهید. برای این کار از منوی Run، گزینه Run in TestRig... را انتخاب کنید. سپس در بخش Input File، فایل تست و در بخش Start Rule، اسم نماد شروع گرامر را انتخاب کنید. فایل تست باید شامل یک برنامه نوشته شده به زبان ساده شده C باشد. خروجی نرم افزار، نمایش گرافیکی درخت تجزیه برای کد نوشته شده در فایل تست است که می‌توانید آن را با استفاده از گزینه png به صورت عکس ذخیره کنید. در صورتی که کد نوشته شده در فایل تست دارای خطا باشد، در پنجره output پیغام‌های خطا نمایش داده می‌شود.

#### ۴-۲- فاز دوم: تولید کامپایلر با استفاده از ابزار ANTLR در محیط Visual Studio .Net

پس از اطمینان از صحت گرامر و عملکرد صحیح تحلیلگرهای لغوی و نحوی در تشخیص لغات و جملات زبان مبدا، در فاز دوم می‌بایست از ابزار ANTLR در محیط Visual Studio .Net استفاده کنید و یک کامپایلر به زبان C# طراحی کنید که کدهای نوشته به زبان ساده شده C را به کدهای سه آدرس ترجمه می‌کند. به طور کلی، فاز دوم شامل مراحل زیر است:

- (۱) در محیط Visual Studio که از ANTLR پشتیبانی می‌کند، یک پروژه به نام دلخواه (برای مثال "MyProject") ایجاد کنید.
- (۲) در بخش Solution Explorer بر روی نام پروژه کلیک راست کنید و گزینه Add → Existing Item... را انتخاب کنید. در پنجره باز شده، فایل گرامر c.g4 که در فاز قبل تولید کرده‌اید را انتخاب کنید تا به پروژه اضافه شود.
- (۳) در فایل c.g4، پس از عبارت "grammar c;"، کدهای زیر را اضافه کنید. این کار باعث می‌شود که در زمان تولید کدهای لکسر و پارسر توسط ابزار ANTLR، کلاس‌های مربوطه در فضای نام پروژه (در اینجا MyProject) تعریف شوند.

```
@lexer::namespace{MyProject}
@parser::namespace{MyProject}
```

همچنین در بخش `@header`، فضاها را نام و در بخش `@members`، متغیرها و ساختارهای مورد نیاز را می‌توانید تعریف کنید.

```
@header
{
    using System;
    using System.Collections;
    ...
}
@members
{
    Hashtable symbolTable=new Hashtable();
    ...
}
```

- (۴) `action` ها و کدهای مورد نیاز برای تحلیل معنایی، تولید کد میانی و جدول نمادها (Symbol Table) را به گرامر اضافه کنید.
  - (۵) پروژه نوشته شده را `Debug` و اجرا کنید. با این کار، در مسیر `"...\MyProject \obj\Debug"` کدهای لکسر و پارسر مربوط به گرامر `c.g4`، به طور خودکار و به ترتیب با نام‌های `"cLexer.cs"` و `"cParser.cs"` ایجاد می‌شوند.
  - (۶) برای استفاده از کدهای لکسر و پارسر تولید شده توسط ابزار ANTLR، یک کلاس با نام دلخواه (برای مثال `Compiler`) را به پروژه اضافه کنید. این کلاس باید شامل یک تابع `main` باشد که در آن لکسر و پارسر تولید شده را مورد فراخوانی قرار می‌دهید تا فایل کد برنامه مبدا را به کدهای سه آدرس ترجمه کنند.
- مثال: `// code`، نماد شروع گرامر است.

```
using System;
using Antlr4.Runtime;

namespace MyProject
{
    class Compiler
    {
        static void Main(string[] args)
        {
            cLexer lexer = new cLexer(new AntlrFileStream("test.c"));
            CommonTokenStream tokens = new CommonTokenStream(lexer);
            cParser parser = new cParser(tokens);
            try
            {
                parser.code();
            }
            catch (RecognitionException e)
            {
                Console.WriteLine(e.Message);
            }
            Console.ReadLine();
        }
    }
}
```

## ۵- نکات مربوط به تحویل پروژه

- پروژه را می‌توانید به صورت گروهی (حداکثر سه نفره) انجام دهید.
- زمان‌بندی پروژه
  - فاز اول:
    - موعده تحویل: ۹۳/۹/۶
    - تحویل فایل گرامر زبان برنامه‌نویسی ساده شده C (c.g4)، کد لکسر به زبان جاوا (cLexer.java)، کد پارسر به زبان جاوا (cParser.java) و فایل اسامی توکن‌ها (c.tokens) که توسط نرم افزار ANLTRWorks2 تولید شده‌اند + تصاویر درخت تجزیه برای کدهای تست
  - فاز دوم:
    - موعده تحویل: ۹۳/۱۰/۴
    - تحویل کامپایلر (پروژه) نوشته شده به زبان C# - ورودی پروژه: یک فایل شامل کد برنامه به زبان برنامه-نویسی ساده شده C - خروجی پروژه: یک فایل شامل کدهای سه آدرسه که معادل با کد ورودی است.

موفق باشید

امامدادی