

به نام خدا

## چگونه یک بکارگیری نرم افزار ParsGn2

با اجرای نرم افزار ابتدا گزینه مورد نظر خود را از بین گزینه های CLR , LALR , SLR انتخاب نمایید. سپس با فشردن دکمه جستجوی سمت راست، فایل BNF را انتخاب کنید. در این لحظه دکمه Start روشن شده که با فشردن آن برنامه شروع به کار می کند. در پایان برنامه فایل های خروجی را با نامهای زیر تولید می کند:

۱. فایل \*.tbl : این فایل حاوی اطلاعات جدولی است که Parser طبق آن کار خود را انجام می دهد. بنابراین وجود این فایل در کنار برنامه اجرایی شما پس از تولید لازم می باشد. در صورتی که این فایل در کنار فایل اجرایی وجود نداشته باشد برنامه پیام مناسب را صادر می کند. پس نگران این فایل نباشید!

۲. فایل \*.log : در این فایل اطلاعات مربوط به مراحل ساخت Parser و خطاها و warning ها قرار دارد. در نتیجه نگران این فایل هم نترسید!

۳. فایل FrmParser.cs : این فایل اصلی برنامه می باشد که شما می بایست داخل آن را کامل نمایید.

در ابتدا برای تست، این برنامه نیاز به هیچ تغییری نداشته و می توان فایل FrmParser.cs را در پروژه RunParser قرار داده و اجرا کرد. برای این کار فایل FrmParser.cs را در فولدر RunParser و کنار Program.cs قرار دهید. بعد از آن فایل RunParser.sln را اجرا کرده و برنامه را کامپایل نمایید. برنامه اجرایی در فولدر \bin\debug ساخته می شود. حال باید فایل \*.tbl را در کنار برنامه اجرایی قرار دهید تا این برنامه بتواند کار خود را انجام دهد.

برنامه اجرایی RunParser.exe را اجرا نمایید. در فرم باز شده ابتدا فایل حاوی کلمات برای اسکنر آزمایشی را توسط دکمه جستجو انتخاب نمایید. بعد از انتخاب فایل، دکمه Run را فشار داده تا برنامه اجرا شده و نتایج آن را ببینید.

فرم کلمات داخل فایل برای اسکنر آزمایشی دارای قوانینی است که آنها را باید رعایت کرد. اول اینکه کلمات باید تنها و تنها یک فاصله از هم داشته باشند. این فاصله باید برای تمام کلمات و علائم خاص برنامه رعایت شود حتی برای پرانتزها و گوشه ها. فاصله بین ابتدای خط تا شروع اولین کلمه و آخرین کلمه تا پایان خط مهم نمی باشد. در زبانهایی که در آنها Enter پایان خط را تشکیل می دهد شما نیز باید آخرین کلمه هر خط را کلمه کلیدی "Enter" به عنوان Token قرار دهید (پنجمین برای "EOF"). تمام کلمات باید مطابق کلمات کلیدی درون فایل BNF، نوشته شوند (از نظر املا). مثالی در ادامه آورده شده است. ابتدا خطوط فایل BNF و سپس خطوط فایل حاوی کلمات را می بینید:

```
<tokens>
```

```
"other" "if" "then" "else" "true" "false"
```

```
<bnf>
```

```
Stm ::= "other" @do_other |
```

```
    "if" ("true" @process_true | "false" @process_true)
```

```
    "then" @start_if Stm @end_if ["else" @correct_if Stm @end_if] .
```

```
-----
```

```
if true then
```

```
    other
```

```
else
```

```
    if false then
```

```
        other
```

```
    else
```

```
        other
```

در این مثال (کلمات اسکنر) می بینید که بین تمامی کلمات و برای مثال در خط اول بین کلمه کلیدی "if" و "true" یک فاصله وجود دارد و همچنین در خطوط دوم و چهارم و پنجم و ششم و هفتم نیز می بینید که شروع کلمات از ابتدای خطوط نمی باشد. در نهایت برای تغییرات خود نیز باید تابع اسکنر خود را جایگزین تابع اسکنر آزمایشی نمایید. همچنین باید درون توابع مربوط به کنش های مفهومی را نیز کامل کنید. در آخر نیز باید تکه کد زیر را هم از درون کدهای دکمه Run حذف کنید تا تابع اسکنر شما به صورت مستقل فایل مورد نظرش را استفاده نماید.

```
if (!File.Exists(txt_ScannerFileName.Text))
{
    MessageBox.Show("نمایید وارد را کلمات حاوی فایل نام");
    return;
}
ScannerFilename = txt_ScannerFileName.Text;
ScannerIndex = -1;
```

تذکره ۱: این برنامه تحت NET 2 نوشته شده و کار می کند.

تذکره ۲: مطالعه Help1 نیز ضروری می باشد.

موفق باشید

سید امین سراج فرد