

تحلیل معنایی و تولید کد – زبان VB.NET

می خواهیم مفسری برای قسمتهایی از زبان VB طراحی کنیم که با توجه به کد برنامه از کاربر اطلاعاتی را دریافت و نتایجی را روی صفحه نمایش چاپ کند. قالب کلی برنامه هایی که باید آن را تفسیر کرد به صورت زیر است:

```
class a
sub b
dim x,y as integer , z as boolean
x = -1 + ((+6+--1))+7           ' x = 13
z = x = 7 or not 3 >= 4 and true 'z = true
x \= 4+1-1                       'x = 3
call input( y)                   'assume y = 1
dim r as boolean
r = false
if not z
  call print( x + 1)             'writing 4
elseif true then
  call print( x + y)
  y *= x
end if
while y < 3 and CBool((1*2) mod 3) 'y < 3 and true
  y = y +2 + CInt(not r)         'y = 4
end while
call print( y)                   'writing 4
end sub
end class
```

۱- قسمت های پررنگ شده در مثال فوق در همه برنامه ها تکراری است و دستور العمل های داخل روال قابل تغییر است.

۲- عبارات ریاضی فقط به دو شکل boolean و integer هستند.

- اعمال محاسباتی ممکن عبارتند از: + و - دو عملوندی و تک عملوندی، ضرب و تقسیم صحیح \، باقیمانده mod. عملوندهای این اعمال فقط از نوع صحیح باید باشند.

- اعمال مقایسه عبارتند از: > < <= >= <>

دو طرف این علائم باید هم تایپ باشند. برای مقادیر منطقی فقط دو عمل = و <> برای مقایسه قابل استفاده است.

- اعمال منطقی and, or, xor و not. عملوندهای این اعمال باید boolean باشند.

- برای تبدیل یک عبارت منطقی به عدد صحیح از تابع CInt استفاده می شود که true را به 1 و false را به 0 تبدیل می کند. تابع CBool نیز برای تبدیل صحیح به منطقی استفاده می شود به طوری CBool(x) با گزاره (x <> 0) هم ارز است.

- به منظور سادگی فرض کنید که اعداد ثابت فقط به شکل دهدهی بوده و فاقد حروف و علائم ویژه هستند.
(مثلاً 345 ممکن است اما 345L و &hA4 نداریم.)

۳- در قسمت دستور العمل ها چندین دستور العمل آورده می شود که دارای انواع زیر است:
الف) تعریف متغیر محلی:

این دستور با dim شروع شده و پس از آن چند تعریف آورده می شود و هر تعریف حاوی چند متغیر که پس از آن as integer و یا as boolean آورده می شود.

ب) دستور انتساب:

در سمت چپ دستور یک متغیر و در سمت راست آن یک عبارت ریاضی دلخواه آورده می شود. دستور انتساب علاوه بر شکل ساده خود یعنی = ممکن است به شکل های =* ، =\ ، += یا -= باشد که عملکردی مشابه زبان C دارند. (نکته پیاده سازی: نماد = هم برای عمل مقایسه و هم عمل انتساب مورد استفاده است. در صورتی که این عمل در ریشه درخت معنا قرار گیرد، انتساب بوده وگرنه مقایسه است. البته در قسمت شرط دستورات کنترلی معنای مقایسه خواهد داشت.)

ج) دستورات I/O که به صورت فراخوانی روالهای زیر است:

call input(متغیر)

call print(عبارت)

این دو دستور عملکرد مشابهی با دو دستور read و write در زبان پاسکال دارند. تایپ آرگومانهای این دو تابع باید از نوع صحیح بوده و در صورتی که از نوع boolean باشند، پیغام خطا باید صادر شود.

د) دستور if :

این دستور به دو شکل تک خطی و بلاکی دیده می شود که از لحاظ منطق تفاوت چندانی با هم ندارند، اما در حالت چند خطی امکان قرار دادن بلاک های کد در قسمت then و elseif و else وجود دارد. نمونه ای از دستور if بلاکی در بالا دیده می شود. نمونه ای از دستور تک خطی به شکلهای زیر است:

```
if x > 2 then else call print(x)
```

```
if x <= 2 then call print(x)
```

```
if x < y then min = x : max = y else min = y : max = y
```

ه) دستور while :

عملکرد این دستور کاملاً مشابه زبان پاسکال است و نمونه ای از آن در مثال صفحه اول دیده می شود.

۴- انواع خطاهای منطقی که باید گزارش شده و سپس برنامه متوقف گردد:

- در صورتی که متغیری مورد استفاده قرار گیرد ولی تعریف نشده باشد.

- در حالتی که یک متغیر دوباره تعریف شود نیز گزارش خطا لازم است. نامهای a و b و input و print را رزرو شده فرض کنید و بنابراین از آنها برای اسامی متغیرها نمی توان استفاده کرد.

- اعمال I/O فقط برای اعداد صحیح تعریف شده که در صورت عدم رعایت این موارد نیز پیغام خطا لازم است.

- عدم سازگاری تایپ عملوندهای اعمال محاسباتی و شرطی و منطقی و عمل انتساب.

- نبودن تایپ قسمت شرط دستورات if و while از نوع Boolean.

خواسته های پروژه:

قسمت سوم (موعد تحویل: ۹۴/۲/۲۲)

در این قسمت فقط کنش های معنایی که جهت تحلیل معنایی و اجرای دستورات فوق لازم است را تشخیص و در فایل BNF درج کرده و مشابه با قسمت دوم پروژه، برنامه ای کامل به همراه کلیه فایل های ساخته شده را تحویل دهید. نام برنامه باید Vb.exe بوده و یک مسیر ورودی را دریافت کرده و معادل با اجرای هر کنش به صورت نمادین نام آن را چاپ کند. مثلاً:

```
Vb.exe inp1.txt
@make_id
@process_dim
@ make_id
@ make_num
@process_exp
success
```

قسمت چهارم (موعد تحویل: ۹۴/۳/۱۲ به صورت حضوری)

در این بخش، بدنه کنش های معنایی مرحله قبل را پیاده سازی کرده به طوری که برنامه یک فایل ورودی را دریافت کرده و آن را تفسیر و اجرا کند. مثلاً:

```
Vb.exe inp1.txt
1←
4
4
```

قسمتهای اختیاری با نمره اضافی:

در صورت انجام قسمت های اصلی و داشتن وقت کافی می توانید چندین بخش از امکانات زیر را پیاده سازی کنید. توضیح جزئیات این دستورات فقط برای افراد علاقه مند در ساعات مشاوره قابل انجام است.

۱- پیاده سازی دستور exit while برای خروج از داخلی ترین حلقه while و دستور exit sub برای خروج از روال

۲- پیاده سازی دستور for برای مقادیر صحیح با قابلیت step

۳- پیاده سازی دستور select فقط برای انواع صحیح

۴- پیاده سازی آرایه یک بعدی

۵- پیاده سازی آرایه دو بعدی