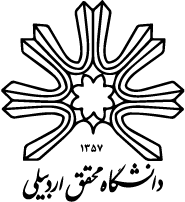
**به نام خدا**



**طراحی یک زبان برنامه نویسی و تعیین ویژگی های آن**

**مستندات مربوط به پروژه تحلیلگر نحوی**

**نام و نام خانوادگی: محمدمهدی خانی**

**رشته: مهندسی کامپیوتر**

**درس: اصول طراحی کامپایلر**

**نام زبان: Eddie**

**۱- شناسه :**

در زبان برنامه نویسی ابداعی من هر متغیر باید با \*\* شروع شود و پس از آن کاربر می‌تواند از هر حروفی برای نام گذاری متغیر استفاده نماید و مجاز است از ترکیب هر کاراکتر دلخواه بهره ببرد. همچنین برای تعیین نام متغیر گذاشتن فاصله بین حروف مجاز نمیباشد و کلمات باید با استفاده از کاراکتر‌های underline(\_) یا عدد از هم جدا شوند.

Example:

\*\*my-name , \*\*5a , \*\*my\_var , \*\*mohammad1mahdi2khani

\*نکته: بین \*\* و اسم متغیر نباید فاصله گذاشته شود.

برای تعیین نوع متغیر، کاربر میبایست بعد از تعیین نام متغیر و گذاشتن مقدار مساوی با یک سری کلمات کلیدی نوع آن را مشخص نماید که این مورد در قسمت سینتکس مجاز و کلمات کلیدی توضیخ داده میشود.

در این زبان برای تعریف تابع از کلمه کلیدی fn استفاده میشود که کاربر اسم تابع را بعد استفاده از کلمه کلیدی تعینن میکند و مجاز است از ترکیب هر کاراکتر دلخواه برای اسم تابع استفده نماید(به مانند اسم متغیر).

**۲- کلمات کلیدی :**

may , maybe , other , rep , num , alph , fn , show , get , call

**۳- اعداد مجاز :**

**اعداد صحیح : 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9**

**اعداد اعشاری :** در این زبان قسمت صحیح و اعشاری اعداد با کاراکتر (.) از هم جدا میشوند مانند :

6.5 , 4.75 , 3.25

|  |  |
| --- | --- |
| جمع | + |
| تفریق | - |
| ضرب | \* |
| تقسیم | % |
| مقداردهی | = |
| مساوی | /=/ |
| نا مساوی | /!=/ |
| کوچکتر | << |
| بزرگتر | >> |
| کوچکتر مساوی | =>> |
| بزرگتر مساوی | =<< |
| به علاوه یک | ++ |
| منهای یک | -- |

**۴- نماد‌ها و عملوند‌های مجاز :**

**۵- سینتکس مجاز در زبان :**

|  |  |
| --- | --- |
| syntax | توضیحات |
| may condition{ statement }  maybe condtion{ statement }  other{ statement } | دستورات شرطی |
| rep \*\*item=n,\*\*item<<m,\*\*item ++ { statement } | دستورات تکرار و حلقه |
| fn functionName(/arguments/){ statement } | روش تعریف یک تابع |
| call functionName(/arguments/) | فراخوانی تابع |
| varName=num(value)  varName=alph(“value”)  varName=otherVariable | مقداردهی به متغیر |
| first + second | جمع دو عدد |
| first - second | تفریق دو عدد |
| first \* second | ضرب دو عدد |
| first % second | تقسیم دو عدد |
| get(/varName/) | گرفتن ورودی |
| Show(/values/) | نمایش در خروجی |

**۶- گرامر اولیه زبان ابداعی:**

**Base Syntax | Operation | Assignment**

**Syntax** if **Condition** { **Base** } **|** else\_if **Condition** { **Base** }

**|** else { **Base** } **|** for **Condition\_for** { **Base** } **|** function id ( / id / ) { **Base** } **|** call id ( / id / ) **|** input ( / id / ) **|** output ( / **Variable** / )

**Condition** id **Comparison** id **|** id **Comparison** number **|**

number **Comparison** id **|** number **Comparison** number

**Operation** id = id **Operator** id **|** id = id **Operator** number **|**id = number **Operator** id **|** number **Operator** number

**Operator** + **|** - **|** \* **|** % **|** ++ **|** --

**Comparison**  /=/ **|** /!=/ **|** << **|** >> **|** <<= **|** >>=

**Assignment** id = **Expr**

**Expr** number ( int ) **|** number ( float ) **|** alph ( “string” ) **|** id

**Variable** id **|** int **|** float **|** “string”

**Condition\_for** id = int , id **Comparison** , id **operator**

**مشکلات گرامر اولیه:**

۱- در غیر پایانه های **Condition و Oeration و Expr مشکل first/first وجود دارد.**

۲- برای **سینتکس for باید یک Condition جدا در نظر گرفته شود.**

**۷- گرامر دوم زبان ابداعی :**

**Base Syntax | Operation | Assignment**

**Syntax** if **Condition** { **Base** } **|** else\_if **Condition** { **Base** } **|**

else { **Base** }**|**for **Condition\_for** { **Base** }**|**function id ( / **Variable** / ){**Base**}

**|** call id ( / **Variable** / ) **|** input ( / id / ) **|** output ( / **Variable** / )

**Condition** id **X |** number **X**

**X Comparison T**

**Operation** id = **T Y**

**Y Operator T**

**Operator** + **|** - **|** \* **|** % **|** ++ **|** --

**Comparison** /=/ **|** /!=/ **|** << **|** >> **|** <<= **|** >>=

**Assignment** id = **Expr**

**Expr** number ( **W** ) **|** string( “string” )

**W** int **|** float

**Variable** id **|** int **|** float **|** “string”

**Condition\_for** id **Z T**

**Z** = int , id **Comparison** int , id **Operator**

**T** id **|** int **|** float

**مشکلات گرامر دوم:**

۱- در غیر پایانه **Base مشکل first/first وجود دارد برای حل این مشکل در گرامر بعدی غیر پایانه Operation را از Base حذف کرده و آن را با Assignment در ارتباط قرار میدهیم.**

**۲- برای else\_if و else باید ترتیب در نظر گرفته شود و نمیتوان آنها را قبل از if استفاده کرد.**

**۳- برای for سه غیر پایانه Variable و Operator و Condition جدا در نظر میگیریم.**

**۴- در سینتکس مربوط به Output باید این مورد در نظر گرفته شود که میتوان در خروجی به غیر از id پایانه های دیگری هم استفاده نمود.**

**۵- برای تعریف و فراخوانی توابع باید گرامر دیگری در نظر گرفته شود چون هر کدام میتوانند بیش از یک آرگومان داشته باشند.**

**گرامر سوم و نهایی همراه تصحیح مشکلات :**

**Base Syntax | Assignment**

**Syntax if Condition {** **Base } If\_continue |**

**for** **Assignment ,** **Condition\_for , id Operator\_for {** **Base** **}** **|**

**function id ( / Function\_call** **/ ) { Base** **}** **| call id ( / Function\_call** **/ )**

**input ( / id / ) | output ( / Variable** **/ )**

**If\_continue else\_if Condition {** **Base } Else\_if\_continue |**

**else { Base** **}** **| LANDA**

**Else\_if\_continue else { Base** **}** **| LANDA**

**Function\_call Variable Call\_continue**

**Call\_continue , Function\_call | LANDA**

**Operator\_for +** **Variable\_for | - Variable\_for | \*** **Variable\_for |**

**% Variable\_for | ++** **| --**

**Condition\_for id Comparison Variable\_for**

**Variable\_for id | int | float**

**Assignment id = X**

**X Expr | Operation**

**Expr number ( Y** **)** **| string ( “string” )**

**Y int | float**

**Operation Variable Operator**

**Operator +** **Variable | -** **Variable | \*** **Variable | %** **Variable | ++ | --** **| LANDA**

**Condition Variable Comparison Variable**

**Comparison /=/ | /!=/** **| <<** **| >> | <<=** **| >>=**

**Variable id | int | float | “string”**