

Compiler – Assignment no1

سهیل فخرخواه

شناسه ها:

متغیرها: قواعدی که برای تعریف متغیرها پیروی از آنها اجباری می باشد به ۳ شماره محدود می شود.

۱. نام متغیرها تنها با حروف بزرگ انگلیسی شروع خواهند شد
 ۲. اعداد فقط در پایان متغیرها مجاز خواهند بود
 ۳. هیچ علائم دیگر ریاضی و منطقی (مثل \$، ^، _، - و ...) در تعریف متغیر مجاز نخواهند بود (تنها علائم مجاز حروف و اعداد در پایان هستند)
- مثال های از متغیرهای غیرمجاز:

soheil, 1num, ad\$ad, ...

توابع: تنها محدودیت های موجود در تعریف نام توابع عدم وجود هرگونه علائم ریاضی و منطقی در آنها و عدم شروع آنها با اعداد می باشد.

مثال هایی از نام توابع غیر مجاز:

1soheil, function-1, ...

کلمات کلیدی:

| کلمه | توضیح |
|-----------|------------------------|
| circum | عبارات شرطی |
| otherwise | عدم صحت شرط |
| go | تعریف حلقه |
| execute | تعریف تابع |
| backflip | برگردونن مقدار از تابع |
| var | تعریف متغیر |

اعداد مجاز:

اعداد صحیح همانند همان اعداد ریاضی و از ۰ تا ۹ هستند.
قسمت اعشاری اعداد به وسیله . از قسمت صحیح جدا می شود.

نمادها و عملگرهای مجاز:

| | |
|----|----------|
| + | جمع |
| - | تفریق |
| * | ضرب |
| / | تقسیم |
| : | مقداردهی |
| <> | مساوی |
| << | کوچکتر |
| >> | بزرگتر |
| | یا |
| && | و |

| | |
|---|----------------------|
| syntax | توضیحات |
| circum a <> 5 => (...) other circum a <> 6 => (...) otherwise => (...) | دستورات شرطی |
| go a:0 to a:10 [1] => (...) | دستورات حلقه و تکرار |
| go until a<>5 => (...) | while |
| execute sum[a, b] => (...) | روش تعریف تابع |
| sum[a, b] | فراخوانی تابع |
| backflip[a] | بازگشت مقدار از تابع |
| var Name: 5 | مقداردهی به متغیر |
| a + b | جمع دو عدد |
| a - b | تفریق دو عدد |
| a * b | ضرب دو عدد |
| a / b | تقسیم دو عدد |

- در دستورات شرطی کلمه `circum` نشان دهنده شرطی بودن بلاک کد است.
- عملیات مورد نظر در هر بلاک کد درون () نشان داده خواهد شد.
- در حلقه‌ها، پارامتر اول شروع حلقه و کلمه `to` کمک به تعیین بازه و پارامتر دوم پایان بازه میباشد. تمپو مورد نظر درون [] نوشته خواهد شد.
- تمام عباراتی که حاصل آنها درون بلاکی از کد جداسازی خواهند شد، به کمک (...) => نمایش داده خواهد شد.
- برای پیاده‌سازی مفهوم `while` در این زبان از همان حلقه `go` و کلمه ای کلیدی یعنی `until` استفاده خواهد شد.