

STEINAXT ©

Quando se programa em Steinaxt é importante estar ciente que o fim de comando é reconhecido pela quebra de linha portanto não fazer o mesmo comando em linhas separadas ou por “;” no final da linha.

Declaração e atribuição de variável:

As variáveis são declaradas da seguinte forma tipo seguido de um ou mais espaços em seguida o nome da variável e caso de mais que uma os nomes separados por virgulas e valor separado do nome por um “=”, o tipo é escrito em minúsculo e as opções são “real” para representar ponto flutuante, “int” para representar inteiro e “text” para representar string. Para que as variáveis sejam declaradas com sucesso, não se pode usar palavras reservadas nem caracteres especiais em seu nome.

Importante lembrar que para declaração de uma variável do tipo real com atribuição de um valor se usa ponto e não vírgula, e se faz necessário o uso desse ponto mesmo se puder ser representado por um inteiro como no exemplo “raio = 5.0”.

exemplos de declaração:

```
real pi = 3.141592, raio = 5.0, diametro
int n = 10, i, j
```

exemplos de atribuições:

```
raio = 5.0
i = n
j = n
```

Controle de fluxo:

Para o uso de controladores de fluxo ou laço, deve se usar if ou loop seguidos da condição e [para que sejam executados os comandos a partir da próxima linha até o fim do escopo delimitado por]. É permitido o uso de aninhamento, ou seja, if dentro de if ou loop ou vice-versa.

```
if “condição”[
    comandos
    .
    .
    .
]
```

Laço:

```
loop "condição" [  
    comandos  
    .  
    .  
    .  
]
```

Os operadores de condição são:

== para igual

!= para diferente

>= para maior ou igual

<= para menor ou igual

< para menor

> para maior

Comando de saída:

O comando de saída é “show” após digitar a palavra reservada se coloca os argumentos separados por virgula, os argumentos que começam com “*” serão impressos até a próxima virgula ou até a quebra de linha caso não tenha mais argumentos, se não tiver “*” vai ser procurado uma variável com o nome daquele argumento e impresso o valor da mesma.

exemplo:

```
int var = 9  
show *valor eh:, var, *entendeu?
```

Saída será :valor eh:9entendeu?

Comando de entrada:

Até o momento a linguagem suporta duas maneiras de entradas, entrada de um valor inteiro ou real, o comando usado para as mesmas é: inputint e inputreal, após o comando se digita a variável de destino, tendo que respeitar o tipo de entrada.

exemplos:

```
int lado  
real raio  
inputint lado  
inputreal raio
```

Operações aritméticas:

add (soma)
sub (subtração)
mul (multiplicação)
div (divisão)
mod (modulo)

Exemplos:

```
diametro = mul raio& 2.0
diametro = add raio& raio
raio2 = mul raio& raio
area = mul raio2& 3.141592
```

lista de palavras reservadas:

int, real, text, add, sub, mul, div, mod, if, loop, show, inputint, inputreal.

Exemplos:

Imprime os números primos no intervalo de 1 até limite:

```
int n=2, ehprimo, res,p, limite
show    *digite o limite para verificar primos
inputint    limite
loop    n<=limite[
    ehprimo = 1
    p=2
    loop p<n[
        res=mod n & p
        if    res ==0 [
            ehprimo=0
        ]
        p=add p & 1
    ]
    if    ehprimo==1[
        show n,* eh um numero primo
    ]
    n= add n&1
]
```

média entre dois números:

```
real    n1, n2
show *Digite dois numeros para fazer a media
inputreal n1
inputreal n2
real media=add n1&n2
media = div media & 2
show *A media entre , n1 , * e , n2, * eh , media
```

Calcula a área e o perímetro de uma circunferência:

```
real area, raio
show *Digite o raio da circonferenceia
inputreal raio
real raio2 =mul raio & raio
area =mul raio2 & 3.141592
real perimetro = mul 3.141592  & 2
perimetro = mul perimetro & raio
show *A area eh , area,*um²,* O perimetro eh , perimetro, *um
```

Imprime as coordenadas de uma matriz quadrada de tamanho n, quando a coordenada esta na diagonal principal imprime d, d:

```
int i=0, j, n
show *Digite o tamanho da matriz quadrada
inputint n
```

```
loop i<n[
    j=0
    loop j<n[
        if i==j[
            show *d d
        ]
        if i!=j[
            show i, * , j
        ]
        j=add j & 1
    ]

    i=add i & 1
]
```

Autores: Leonardo Bianchini leonardobianchini7@gmail.com

João Carlos Becker joaoc.becker@hotmail.com