

Linguagem de programação Krauts

Manual do Programador

Krauts é uma linguagem de programação utilizando notação polonesa e uma sintaxe similar a de linguagens de montagem (Assembly).

Krauts é interpretada linha por linha; assim, o final de cada comando é reconhecido pelo caractere fim-de-linha, e não há como ter mais de um comando na mesma linha.

Os comandos da linguagem Krauts são **case-insensitive**, ou seja, não diferenciam letras maiúsculas e minúsculas. Nomes de variáveis e textos, contudo, são **case-sensitive**.

Por que notação polonesa?

A maioria de nós está acostumado com a notação infixa, que segue o modelo **operando1 operador operando2**, ou $A + B$, por exemplo. Isto implica trabalhar com parênteses e prioridade de operadores para evitar expressões ambíguas.

A notação polonesa, também conhecida como notação prefixa, segue o modelo **operador operando1 operando2**, ou $+ A B$, assim evitando a necessidade de parênteses ou prioridade de operadores para resolver expressões.

Exemplo

Expressão em notação infixa: `var = (5 - 6) × 7`

Expressão em notação prefixa: `= var * - 5 6 7`

Expressão em Krauts: `ATR var MUL SUB 5 6 7`

Comentários

Comentários em Krauts são indicados pelo caractere "til" (`~`). Exemplo:

```
prt add 10 5 ~ Este é um comentário de linha
```

```
~ 0 valor impresso pela linha acima foi 15
```

Sintaxe da linguagem Krauts

Segue um resumo dos comandos da linguagem e suas respectivas sintaxes.

Comando	Utiliza-se no início da linha?	Resumo	Sintaxe
new	Sim	Declara uma nova variável do tipo double.	NEW a new b
atr	Sim	Atribui à variável dest o resultado da expressão orig .	ATR a ADD 2 2 atr b a
prt	Sim	Imprime na saída padrão o resultado de uma expressão aritmética ou booleana, de acordo com o primeiro argumento passado. Caso o primeiro argumento tenha como prefixo o caractere aspas duplas (“), imprime todo o resto da linha como uma mensagem na tela.	prt “Ola Mundo! prt MOD 10 3 prt eq a b prt gt 10 5 ~True
if	Sim	Controle de fluxo: se a expressão for verdadeira, executa o código das linhas que antecedem o comando fi .	if <expr...>
fi	Sim	Encerra o escopo do último comando if .	fi
while	Sim	Laço de repetição: executa o código das linhas que antecedem o comando done enquanto a expressão for verdadeira.	while <expr...>
done	Sim	Encerra o escopo do último comando while .	done
add	Não	Retorna a soma do resultado de duas expressões.	ADD i 10 add a b
sub	Não	Retorna a subtração do resultado de duas expressões.	Sub op1 op2
mul	Não	Retorna o produto do resultado de duas expressões.	Mul op1 op2
div	Não	Retorna o quociente da divisão do resultado de duas expressões. Programa encerra em erro caso op2 seja equivalente a zero.	Div op1 op2
mod	Não	Retorna o resto da divisão do <u>chão</u> do resultado de duas expressões. Programa encerra em erro caso op2 seja equivalente a zero.	Mod op1 op2
eq	Não	Comparação de igualdade. Como todas as comparações, serve para ser colocado em um if ou while .	Eq op1 op2 if eq a b
lt	Não		
	Não		
	Não		
	Não		

	Não		
	Não		
	Não		
	Não		