Lógica de Programação

Aula 03 - Tipos de Dados

Apresentação da Aula

Olá, seja bem-vindo(a)! Na aula passada você conheceu as instruções primitivas de entrada e saída do Potigol e a partir delas se viu os tipos de dados (inteiro, real e texto/cadeia de caracteres). E, nesta aula você vai explorar ainda mais o uso dessas instruções, aprenderá o que são expressões e como construí-las, verificando seu funcionamento e além disso, irá identificar a estrutura de decisão **se...então**, importante recurso utilizado pelas linguagens de programação para criar algoritmos com desvios de fluxo.

Objetivos

Construir expressões;

Conhecer e utilizar os Operadores Aritméticos, Relacionais e Lógicos;

Identificar e construir estruturas de decisão **se...então**.

Você já sabe como definir variáveis (espaços de memória para armazenar algum dado) e como preencher as variáveis através dos comandos de entrada e de atribuição? Não!? Pra relembrar, veja o exemplo abaixo:

Conteúdo interativo, acesse o Material Didático.

Como você deve ter observado, foi realizada a definição de quatro variáveis, uma do tipo inteiro, uma do tipo texto, uma do tipo real e outra de tipo inteiro. Na primeira e na terceira variáveis foram atribuídos valores iniciais, já na segunda e quarta variáveis foram utilizados os comandos de entrada para, na execução do algoritmo, informar os valores que irão preencher essas variáveis.

Veja o código abaixo:

Conteúdo interativo, acesse o Material Didático.

Na execução do código acima, foi realizada a operação 5+3, e o resultado dessa operação foi armazenado na variável **resultado** e ao final a impressão da frase "O resultado é 8", pois o conteúdo da variável **resultado** foi inserido no local indicado no comando **escreva**.

O que você acha em utilizar o exemplo acima e testar com as outras operações matemáticas básicas? Experimente realizar as operações de subtração, multiplicação e divisão e veja o que acontece.

Expressões

Antes que você fique com a expressão de preocupação, rsrsrs... entendeu o trocadilho? O que você fez no exemplo anterior foi uma **expressão** que, em linguagens de programação, pode ser definido como uma associação de valores, variáveis, operadores ou funções (não se preocupe em saber o que são funções agora, você irá conhecer mais adiante em nossas próximas aulas) que serão avaliadas (processadas) de acordo as regras particulares da linguagem e, em seguida, um valor é produzido.

Em outras palavras pode-se dizer que uma **expressão** tem como objetivo final obter um valor após o seu processamento (cálculo), seja ele lógico, matemático ou textual. Assim, constantemente, você estará construindo expressões nos algoritmos. Através delas você realizará os procedimentos necessários para a resolução de um problema utilizando a programação.

Nas expressões são empregados diversos operadores e nesta aula você irá conhecer três tipos que são utilizados na programação de computadores:

- Operadores Aritméticos;
- Operadores Relacionais; e
- Operadores Lógicos.

Operadores Aritméticos e suas expressões

As expressões aritméticas são construídas utilizando-se os operadores aritméticos, esses se assemelham aos que são aplicados para realizar cálculos matemáticos. Nas expressões aritméticas são utilizados números e/ou variáveis associados através dos operadores matemáticos. Na linguagem Potigol têm-se os seguintes operadores aritméticos (Quadro 01).

Quadro 01 - Operadores aritméticos e exemplos de expressões aritméticas

Operador	Operação	Exemplos de Expressões
+	Adição/Soma	5 + 3 5 + a

Operador	Operação	Exemplos de Expressões
-	Subtração	10 - 2 a - b
*	Multiplicação	c * d 2 * 36
/ div	Divisão	10 / 2 x / y
mod	Resto da divisão	10 mod i 10 mod 4
^	Potência	2^3 e^f

Os **operadores aritméticos** são um conjunto de símbolos que representam operações matemáticas que permitem realizá-las nos algoritmos.

Agora, aproveite para testar as expressões dos exemplos do Quadro 01 de operadores aritméticos. Lembrando que você irá utilizar o editor abaixo para realizar os testes. Para que todas as expressões funcionem corretamente você terá que definir os valores iniciais das variáveis, portanto antes de escrevê-las, na prática, você deverá declarar as variáveis (se você ainda tiver dúvidas sobre a definição e declaração de variáveis, consulte a aula anterior.). Segue uma sugestão de valores iniciais:

- a = 1
- b = 2
- c = 3
- d = 4
- x = 6

- y = 3
- i = 9
- e = 2
- f = 3

Conteúdo interativo, acesse o Material Didático.

Agora que você viu e testou as expressões apresentadas, espero que tenha conseguido compreender o que são essas expressões aritméticas a partir do uso dos operadores aritméticos.

Operadores Relacionais

Você irá conhecer os operadores relacionais e como construir suas expressões. Eles são utilizados para estabelecerem a comparação entre operandos, sejam numerais ou variáveis, e podem ser associados com diversos tipos de dados: numéricos, lógicos ou texto/literais.

Quando uma expressão que utiliza os operadores relacionais é avaliada (calculada) o seu resultado, após a avaliação, pode assumir apenas dois valores: '**true**' (verdadeiro) ou '**false**' (falso). Veja o exemplo abaixo:

Conteúdo interativo, acesse o Material Didático.

No exemplo o valor da variável **'a'** é '**true**', pois 3 é menor do que 5. Experimente alterar o operador relacional para '**>**' (maior que) e observe que o valor de **a** será diferente, assumindo um novo valor, qual?

O Quadro 02 apresenta os operadores relacionais presentes na linguagem Potigol. A maioria das linguagens de programação faz uso dos mesmos operadores relacionais.

Quadro 02 - Operadores relacionais e exemplos de expressões relacionais

Operador Relacional	Descrição	Exemplo
<	Menor que	5 < 3 a < b
<=	Menor ou igual	10 <= 2 a <= b
>	Maior que	c > d 2 > 36
>=	Maior ou igual	10 >= 2 x >= y
==	Igual	10 == i 10 == 4
<>	Diferente	2 <> 3 3 <> 3

Que tal verificar qual o resultado de cada um dos exemplos? Declare e defina o valor inicial dessas variáveis. Abaixo elenquei algumas sugestões de valores, mas fique à vontade para experimentar valores diferentes e então avaliar o seu resultado.

Você deverá criar um quadro (ver Figura 01) numa folha de papel com três colunas, na primeira escreva a expressão, na segunda os valores das variáveis e, na terceira o resultado que a expressão deve gerar, se vai ser **verdadeiro** ou **falso**. Só então, você faz na prática e verifica se o resultado coincidiu com as suas respostas. Vamos lá!?

Figura 01 - Modelo de quadro na folha de papel com resultados

Conteúdo interativo, acesse o Material Didático.

Como você deve ter percebido, uma expressão relacional ou é **verdadeira** ou é **falsa**. Elas (as expressões relacionais) serão muito importantes nas próximas etapas dessa disciplina.

Você conseguiu acertar o resultado de todas as expressões dos exemplos apresentados? Se não, recomendo que você realize uma revisão nas expressões antes de prosseguir para o próximo conteúdo.

Operadores Lógicos

Conheça agora o terceiro grupo de operadores que será utilizado nessa disciplina, os operadores lógicos. Com eles, as expressões relacionais podem ser combinadas. Com essa combinação, você poderá construir expressões com operadores relacionais e lógicos. Essas expressões, assim como as construídas apenas com os operadores relacionais, quando avaliadas, possuem os mesmos valores possíveis: **true** (verdadeiro) ou **false** (falso).

Na linguagem Potigol há três operadores lógicos: '**|ou|**', '**|e|**' e '**|não|**'. O operador '**|e|**' é considerado o de conjunção, já o '**|ou|**' é o inverso, ele é o de disjunção. No terceiro, '**|não|**' é um operador lógico de negação. Acredito que você já se deparou com esses operadores na disciplina de Matemática Aplicada, então, você deve estar familiarizado.

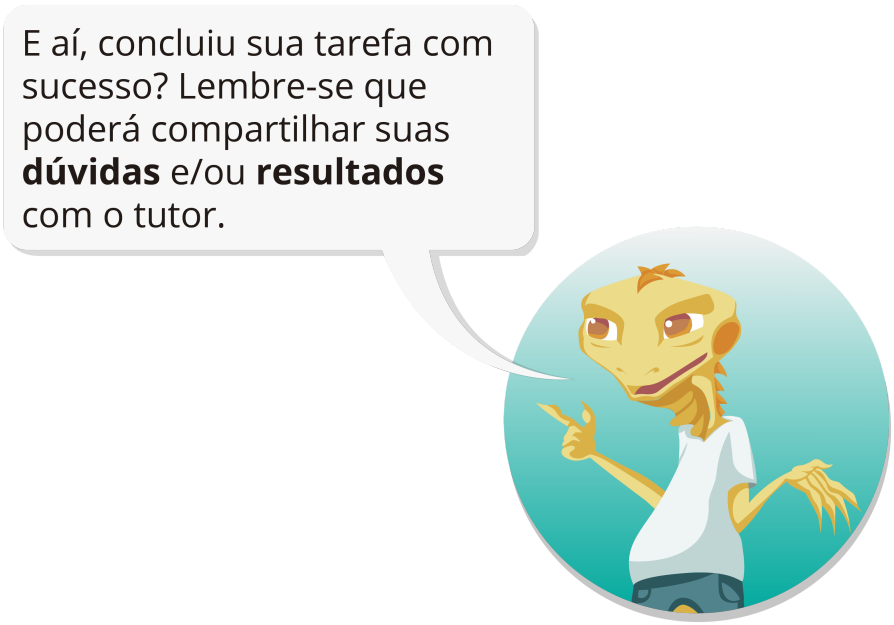
Pratique um pouco! Utilize esses operadores em alguns exemplos. Assim, ficará mais fácil para você compreendê-los. Considere o Quadro 03 abaixo e construa um semelhante em uma folha de papel.

Quadro 03 - Operadores lógicos

Nº Exp.	Valor de A	Valor de B	Operador '|e|'	Operador '|ou|'	'|Não|' A	'|Não|' B
1	Falso	Falso				
2	Verdadeiro	Falso				
3	Falso	Verdadeiro				
4	Verdadeiro	Verdadeiro				

Seu próximo passo, será construir as expressões e realizar o teste no simulador. Em seguida, anote o resultado obtido em cada expressão e, então, preencha os espaços vazios no quadro que foi construído por você. O editor abaixo apresenta, como exemplo, a expressão de número 1 do Quadro 03. Altere os valores das variáveis **a** e **b** e também a operação para obter os demais resultados.

Conteúdo interativo, acesse o Material Didático.



Precedência de Operadores

Você recorda que na Matemática algumas operações devem ser realizadas antes de outras? Observe a expressão matemática abaixo e encontre o valor de x.

$$x = 2 + 3 * 4 - \frac{10}{2}$$

Qual o valor de **x**? 5 ou 9 ou 15?

É provável que você tenha encontrado a resposta correta para essa equação matemática, pois deve recordar-se que algumas das operações possuem **precedência** em relação às outras. Por esse motivo você deve ter realizado, primeiro, a multiplicação, e em seguida a divisão (fração). Só então procedeu com a soma e, por último, a subtração, correto?

Essa precedência também existe nas operações realizadas pelo computador. Assim, como na matemática, o computador realizará primeiro as operações de exponenciação, em seguida, multiplicação e divisão para, só então, realizar as operações de adição e subtração. O Quadro 04 apresenta um resumo sobre as precedências dos operadores aritméticos.

Quadro 04 - Precedências dos operadores aritméticos

Operador	Operação	Prioridade de Precedência
^	Exponenciação	0
* / mod	Multiplicação Divisão Resto da divisão	1 1 1
+ -	Adição Subtração	2 2

Nesse quadro foram apresentados três níveis de precedências, de acordo com as operações. O nível 0 (zero) representa a precedência máxima, o nível 1, que é intermediário e o nível 2 que representa a menor precedência. As operações presentes em uma expressão serão calculadas considerando, primeiro, a prioridade de precedência (começando pela prioridade zero). Caso exista mais de uma operação com o mesmo nível de precedência, as operações serão realizadas sequencialmente da esquerda para a direita, obedecendo a ordem em que aparecem na expressão.

Veja o exemplo apresentado e informe o valor que resultou dessa expressão. Depois, substitua a soma pela operação de exponenciação. Então? Com esse novo procedimento houve alguma mudança no resultado?

Conteúdo interativo, acesse o Material Didático.

Usando parênteses

E se for necessário alterar a ordem em que as operações são realizadas em uma expressão? Por exemplo na expressão '**x := 2 ^ 3 + 4 / 2**' se quer que a soma seja realizada primeiro e, só então, a multiplicação e a divisão.

Para que isso seja possível deve-se utilizar os parênteses para definir explicitamente qual a ordem que as operações devem ser realizadas. Para o exemplo citado, a expressão ficaria assim '**x := 2 ^ (3 + 4) / 2**'. Se você testar essas duas expressões verificará que os valores finais serão bem diferentes. Experimente!

Quando se utiliza os parênteses define-se subexpressões e essas, por sua vez, deverão ser calculadas primeiro antes da expressão total ser avaliada por completo. Quando uma subexpressão é então avaliada, ela é substituída por seu valor.

Quando em uma expressão houver vários níveis de subexpressões, a mais interna será sempre avaliada primeiro. Observe novamente um exemplo no simulador, qual será o resultado obtido nessa expressão?

Conteúdo interativo, acesse o Material Didático.

Na expressão do simulador há uma subexpressão mais interna que é a soma de '3+4'. Somente após essa soma ser realizada e substituída pelo seu valor, é que a subexpressão externa (a operação de divisão) será avaliada. Quando todas as subexpressões forem avaliadas a expressão final poderá então ser calculada e seu resultado será atribuído a variável, denominada 'resultado'. Resumidamente, após a soma ser realizada, será a vez da operação de divisão e, em seguida, a de multiplicação.

Estruturas de Decisão

Desde o início dessa disciplina, até o momento, você construiu pequenos programas e todos eles funcionam de forma linear, ou seja, cada linha do seu algoritmo é executada e processada uma a uma. Essa estrutura linear é também conhecida como **Estrutura Sequencial**, pois todas as linhas do algoritmo são executadas sequencialmente uma após a outra.

Para construir algoritmos e programas que contemplem os diversos tipos de problemas as linguagens de programação possuem as estruturas de decisão. Essas estruturas são peças fundamentais em qualquer linguagem de programação e permitem criar algoritmos com desvios de fluxo.

Para que um desvio de fluxo ocorra é preciso verificar se uma condição é **verdadeira** ou **falsa**. Ou seja, se uma determinada condição for **verdadeira**, o algoritmo deverá seguir por um fluxo, se for **falsa** ele deverá, então, seguir por outro. Veja a figura a seguir, ela representa um algoritmo em **formato de fluxograma**.

Figura 02 - Algoritmo em formato de fluxograma

Você lembra dos operadores relacionais vistos no início dessa aula? E também das expressões relacionais, que quando avaliadas (calculadas), resultam sempre em um único resultado: **verdadeiro** ou **falso**? A partir de agora você irá perceber que as expressões relacionais e as lógicas são utilizadas com frequência nas estruturas de decisão.

No exemplo contido no fluxograma, a expressão relacional está verificando se o valor informado ao programa e, consequentemente, armazenado na variável **x**, é maior do que **0** (zero).

Se o valor de **x** for maior do que zero a condição é então **verdadeira** e o programa segue o seu fluxo, partindo para o bloco da direita. Todos os comandos que estiverem dentro do trecho correspondente ao bloco da **condição verdadeira** serão executados e, nesse exemplo, temos o comando **escreva**, que irá imprimir na saída do programa o valor de **x**, o qual então é positivo.

Caso o valor de **x** seja negativo (ou igual a zero), o fluxo do programa será, então, direcionado para o bloco de código responsável pela condição **falsa**, e nesse programa apresentado, será impressa a mensagem de que o valor de **x** é negativo.

Por esse motivo, quando emprega-se o uso das estruturas de decisão, o programa produzido deixa de ser sequencial, pois alguns trechos podem não ser executados por causa dos desvios de fluxo.

Na linguagem Potigol a estrutura de decisão é denominada de **'se...então'**. Essa estrutura é descrita como um **comando condicional de execução em bloco** e, através dela, é possível definir um ou dois blocos de códigos que devem ser executados, um quando a condição for **verdadeira** (obrigatório) e outro quando for **falsa** (opcional).

Observe o código abaixo, ele representa um programa que verifica se um número fornecido é maior do que zero.

Conteúdo interativo, acesse o Material Didático.

Se a condição for **verdadeira**, o programa irá escrever que o número digitado é positivo. Já se for **falsa**, o comando **escreva** não será executado. Esse exemplo mostra a estrutura de decisão **se...então** em sua forma mais simples, ou seja, quando temos apenas o bloco de códigos para quando a expressão de condição for **verdadeira**.

No exemplo, se a condição for **falsa**, não há bloco de códigos para ser executado e, portanto, o algoritmo apenas irá continuar executando as linhas seguintes ao bloco **'se...então'**. Nesse caso como não há nenhuma linha código após esse bloco o programa é encerrado.

Explore o exemplo abaixo para você conhecer a sintaxe dessa estrutura.

Conteúdo interativo, acesse o Material Didático.

Você conseguiu descobrir o que faz o programa acima? Ele declara uma variável **'numero'** e em seguida lhe atribui um valor, veja na linha 1. Na linha seguinte, linha 2, temos uma expressão aritmética e relacional e que após ser avaliada tem o seu resultado armazenado na variável **misterio**. Em seguida, nas linhas de 4 a 6, temos o bloco da estrutura condicional **'se...então'**. Caso você não tenha compreendido completamente o programa, experimente alterar a expressão da linha 2 para obter novos resultados, mas lembre-se sempre que as expressões relacionais possuem o resultado **true** (verdadeiro) ou **false** (falso).

Agora observe o bloco de código abaixo, esse corresponde ao programa do fluxograma apresentado anteriormente. Sua estrutura apresenta o comando **se...então...senão** completo com os dois blocos de código:

- um para quando a condição (**x>0**) for **verdadeira**, ou seja, **x** é maior que zero, sendo o bloco de comandos inseridos entre as palavras reservadas **então** e **senão**;

- um bloco para quando a condição (**x>0**) for **falsa**, ou seja, **x** for menor ou igual a zero, sendo o bloco de comandos inseridos entre as palavras reservadas **senão** e **fim**.

Conteúdo interativo, acesse o Material Didático.

Você compreendeu a estrutura de decisão **se...então**? Ainda não!? Poste suas dúvidas no fórum designado, dessa forma outros colegas poderão compartilhar informações e seu professor mediador poderá lhe auxiliar.

Atividade 01

Conteúdo interativo, acesse o Material Didático.

Atividade 02

Conteúdo interativo, acesse o Material Didático.

Resumo

Nessa aula você conheceu o que são expressões e que elas podem ser criadas com operadores aritméticos, operadores relacionais ou lógico. Além disso, também conheceu a precedência de operadores e como definir explicitamente a precedência de operações em uma expressão.

Ao final foi trabalhada a estrutura de decisão **se...então**, que é fundamental para a maioria dos algoritmos e programas que são construídos.

Referências:

POTIGOL é... Disponível em: <http://potigol.github.io/>. Acesso em: 02 out. 2017.