

Variáveis e Constantes e e Seus Tipos	
Objetivo Geral: Compreender o uso de Variáveis e Constantes	
<i>Conteúdo:</i>	
1.	Conhecer a Função das variáveis e Constantes;
2.	Compreender o uso das Variáveis e Constantes nos Programas;
<i>Metodologia e Estratégia:</i>	
1.	Aula expositiva dialogada com apoio de tutorial;
2.	Exercícios de aplicação.

Variáveis e Constantes e seus Tipos

Variáveis e constantes tem a função de guardar os valores que são manipulados durante todo o algoritmo. Eles são os responsáveis por dar todo o dinamismo que temos dentro das linguagens de programação. Isso porque, ao invés de utilizar o valor diretamente, fazemos o uso das variáveis ou constantes. Observe o exemplo:

```
double distanciaPercorrida = 250.0;
```

A variável acima nos ajuda a fazer referência para o valor – 250. Daí não precisamos ficar escrevendo o número 250 em vários lugares. Só escrevemos o 250 no momento da declaração da variável. O melhor é que, se o número mudar de 250 para 251, nós precisamos alterar somente no momento da declaração.

Agora, qual a diferença entre uma variável e uma constante? Na verdade, a diferença está no nome de cada um: uma variável pode variar seu valor e uma constante não pode. Na prática, tudo que declaramos até aqui - como a variável `distanciaPercorrida` - foram variáveis, e uma constante você declara assim:

```
final double distanciaPercorrida = 250.0;
```

Reparou a diferença? Foi somente o uso da palavra final no início da declaração. Caso você insista, conscientemente ou não, em mudar o valor de uma constante, e crie um programa como abaixo:

```
public class UmaConstante {  
  
    public static void main(String[] args) {  
        final double euSouConstante = 10.0;  
        euSouConstante = 25.0; // Isso aqui não pode!  
    }  
}
```

ERRO?

Executando um programa com uso de variáveis

```
public class Variaveis {  
    Run | Debug  
    public static void main(String[] args) {  
        System.out.println(x: "=====");  
        // Altere o valor da variável abaixo para realizar testes  
        double precoProduto = 115.0;  
        System.out.println("O preço do produto é de "  
            + precoProduto + " reais");  
        double percentualDesconto = 10.0;  
        System.out.println("O desconto será de "  
            + percentualDesconto + "%.");  
        // Regra de três para descobrir qual o valor  
        // do desconto.  
        double desconto = precoProduto * percentualDesconto / 100  
        // Preço do produto menos o desconto para  
        // sabermos o valor do produto com o  
        // desconto que foi calculado na linha anterior.  
        double precoComDesconto = precoProduto - desconto;  
        System.out.println("Com desconto, o produto sai "  
            + "por R$" + precoComDesconto + ".");  
        System.out.println(x: "Fim!");  
        System.out.println(x: "=====");  
    }  
}
```

Tipos de Dados Existentes

No Java nós temos vários e vários tipos de dados, mas dentre esses muitos tipos, temos alguns que são principais. Antes de mencioná-los, vou agrupá-los em 3 categorias:

- Numéricos
- Textos
- Lógicos

Dentro dos numéricos nós ainda temos dois subgrupos:

- Inteiros
- Decimais

Quanto aos tipos inteiros, nós temos:

- byte (armazena valores de -128 até 127)
- short (vai de -32768 até 32767)
- int (ocupa o intervalo de -2³¹ a 2³¹ -1)
- long (intervalo de -2⁶³ a 2⁶³ -1)

Já os decimais, são:

- float (7 dígitos)
- double(16 dígitos)

Com relação ao grupo de tipos que armazenam texto, temos:

- char (armazena um carácter por vez)
- String (armazena textos de qualquer tamanho)

Dentre esses dois, o String é muito mais útil para nós trabalharmos durante os nossos algoritmos. Isso é simplesmente porque o char só consegue armazenar um carácter por vez.

Por fim, no grupo referente ao tipo lógico, temos somente o tipo:

- boolean

Esse é um tipo importante porque nos ajuda com o gerenciamento das decisões que nosso algoritmo tem ou pode tomar. O tipo boolean pode assumir somente dois valores: ou ele é verdadeiro ou falso. O valor para verdadeiro é representado como true e o falso como false.

Escolhendo bons nomes para variáveis e constantes

E qual seria o motivo de nos preocuparmos com os nomes de nossas variáveis? O primeiro motivo é que nossa memória pode falhar, ou seja, mesmo que você crie um algoritmo que só você vai mexer, caso você termine ele e precise voltar um mês depois para alguma alteração, provavelmente, terá dificuldades em lembrar de algumas coisas, o que inclui o porquê de ter declarado determinadas variáveis dentro dele.

O segundo e principal motivo é que muitas vezes você não irá criar um algoritmo sozinho e, muito menos, um software todo sozinho. Claro que é possível e você pode fazer isso, mas em boa parte das vezes - eu diria, a maioria - você vai criar software juntamente com outras pessoas - ainda mais se for trabalhar como funcionário em uma empresa onde, quase certo, fará parte de uma equipe com outros programadores.

Para que tenha uma referência a seguir, procure não abreviar as palavras caso o nome da sua variável contenha 3 palavras ou menos. Se passar disso, caso você queira, pode abreviar algumas delas, mas sempre com muito cuidado para não prejudicar o entendimento do que sua variável representa no algoritmo como um todo.

Como última dica, use o padrão de nomes chamado camelCase. Esse é um padrão que diz que, em nome de variáveis compostas por mais de uma palavra, devemos, a partir da segunda palavra, iniciá-la com a primeira letra em

maiúscula, como foi o caso daquelas que declaramos em nosso algoritmo. Veja alguns outros exemplos:

```
double distanciaPercorrida = 0.0;
```

```
double tempoGasto = 0.0;
```

```
int idadeMinimaParaTirarCarteiraDeMotorista = 18;
```

```
String nomeDoProduto = "Caneca de alumínio 500ML";
```

Procure dar bons nomes para suas variáveis sempre. Não fique com preguiça só porque você acha que o nome ficará grande. Você vai ver, principalmente quando começar a criar programas maiores, que vale a pena.