

>>>>>>> 09/05/2022 <<<<<<<<

Variáveis, Tipos de Dados e Operadores Matemáticos
em Java - Prof. Thiago Leite Carvalho

1) Conceituação e criação

- Introdução

GitHub do prof.: /tlcdio (cursos e projetos)
/thiagoleitecarvalho (pessoal-prof-artigo)

Criação e manipulação de variáveis

Requisitos: logica de progr, Java 11, IntelliJ

- Conceituação e criação

conceito e criação de variaveis

Ponto de partida pra programas em Java é o método main, usado pra disparar execução do programa, onde são criadas as variáveis iniciais. O padrão de definição é:

<?visibilidade?><?modificador?>tipo nome <?=valorInicial?>;

o que está com ? é opcional. Tipo de dado deve sempre ser definido, por isso se diz que Java é fortemente tipada porque sempre devemos informar tipo da variável.

Nome da variável para usar e entender pra que serve.

Ligados à POO= V(public, protected, private), M(static)

Var. definida como final é constante, valor não varia e deve ter o valor informado/atribuído.

Variável parâmetro é ligada à assinatura do método.

"QUANTIDADE_\$_OPCOES" é definição válida de var. em Java.

Atentar pra convenções e regras que, se não atendidas, causam erros de compilação (ex.: caracteres especiais, espaço, nro no início).

Notação camelo = exceto o início da palavra, 1a. letra das outras vai em maiúscula no nome expressivo da var.

Nome de constante é todo em maiúscula, separado por _ (underline)

[O que é camelCase?](<https://coodesh.com/blog/dicionario/o-que-e-camelcase/>)

Exemplos de boas práticas:

int QuantidadeProduto; (notação camelo com 1a letra em maiusc, nao gera erro de compilação, não é boa prática).

int final numeroTentativas = 5; (faltou maiuscula no início do nome, nao gera erro e não é boa pratica).

int NUMERO_TENTATIVAS = 5; (notação camelo em variável local, não gera erro e não é boa prática).

int qtdProd (nome pouco claro dificulta entendimento, não gera erro e não é boa prática).

int i (é iterador/contador em loop de repetição, pode ser descartado no final, não gera erro, é boa prática).

Adotar boas práticas ajudam na escrita clara e facilita manutenção futura, demonstra que o programador colabora com a equipe, além de prezar pela qualidade de código.

- Exercitando

criado codigo vtoAula01

2) Tipos de dado

- Tipos de dado

o que são tipos de dados e como usá-los

Tipos >>>>>

Estatica forte obriga definir tipo de variavel durante a criação;

Dinamica (fraca) é tipo inferido na execucao da aplicação, por isso é fraca.

Primitivos são tipos basicos, atômicos, homogêneos, como dados numericos e textuais.

Composto ligado à P00, struct da lgm C é tipo de dado composto (não é 00).

Tipos de dados: textual, numeral, lógico, objeto

Tipos numericos tem capacidade de armazenamento proporcional a sua quantidade de bits.

Logicos não tem tamanho, V ou F

Objetos usados na P00.

float = nro fracionado de 32 bits, deve colocar f no uso

double = nro fracionado de 64 bits, uso opcional de d, há inferencia que seja double na falta de d.

Float e double podem armazenar nros inteiros.

Uso de caracter de forma isolada requer aspas simples.

String é classe e objeto do Java pra trabalhar com texto, não é dado primitivo. Trabalhar com caracter isolado é

mais trabalhoso, é mais usual declarar variaveis para colocar val textuais. Originalmente string é uma dado composto que pelo uso virou praticamente dado primitivo.

Booleano não tem operação executada sobre esse tipo de dado, só informa se é V ou F.

Char tem valor default = null

Cada tipo de dado tem um valor default ou valor inicial atribuído à variavel mesmo quando o programa não traz essa informação. Lgms moderna adotam o recurso pra evitar erro na inicialização de variaveis, caso contrario pode haver erro de execução por acessar sujeira na memoria. Boas práticas: usar tipo de dado adequado, pois se usar string pra tudo no programa, será necessário convertê-la antes de operações aritmeticas. Avaliar a finalidade da variavel antes de criá-la, assim se descobre qual tipo de dado é adequado, quais operaçoes poderá fazer.

[Struct em C](<http://linguagemc.com.br/struct-em-c/>)

- Exercitando

Numeral long = colocar L no final pra Java entender que é nro long e não somente um inteiro.

Criado codigo vtoAula02

3) Operadores Aritméticos

- Operadores aritméticos

Operador pós-fixado tem execução maior que pré-fixado

int i = ++k; (operacao, usa var -> i = k + 1;)

```
int j = K--; (usa var, operacao -> j = k; K = k-1;)
double d = f (atribuicao de float em double)
i += 5; (i = i + 5)
j -= 3; (j = j - 3;)
k %-2; (k recebe resto de k; k = k % 2;)
```

- Exercitando

Trabalhado codigo vtoAula03

4) Conversões (casting)

- Conversões (casting)

As capacidades de representação e armazenamento podem ser diminuídas ou aumentadas.

Upcast - implícito, promove dado menor para tipo maior, não inf tipo de dado, há elevação porque é automática.

Downcast - deve ser explícita a conversão por diminuir a capacidade de representação e armazenamento, a decl explícita previne perda de dados.

conversao i = (int) f; ==> perda de real pra inteiro, descarta casa decimal e pega só parte inteira.

Conversao gera erro de compilação: float f = double

- Encerramento

Links de documentação

- Exercitando

Trabalhado codigo vtoAula04