Aula 01

Tipos de dados

Rogério Pereira Junior

rogeriopereirajunior@gmail.com

1 de abril de 2024

SENAC
Santa Catarina
Florianópolis

Tipos de dados

Introdução

- Na base de qualquer linguagem de programação estao seus tipos de dados e operadores, e Java nao e exceção
- Esses elementos definem os limites de uma linguagem e determinam o tipo de tarefas as quais ela pode ser aplicada
- Os tipos de dados sao especialmente importantes em Java porque essa e uma linguagem fortemente tipada. Ou seja, todas as operacoes tem a compatibilidade de seus tipos verificada pelo compilador.
- Vocês conseguem somar duas palavras?
- **Nao ha o conceito de uma variavel “sem tipo”**

Tipos de dados

Um tipo de dados estabelece um conjunto particular de valores que podem ser representados dentro de um programa e para os quais podem ser definidas operações específicas

Comentários

- Em Java, você pode adicionar comentários ao seu código para fornecer informações adicionais sobre o propósito e funcionamento do código.
- Existem dois tipos principais de comentários em Java: de uma linha e de várias linhas.
- De uma linha

```
// Este é um comentário de uma linha
String idade = 25; // Este é um comentário ao lado de uma instrução
```

- Comentários de Múltiplas Linhas:

```
/*
   Este é um comentário de várias linhas.
   Pode se estender por várias linhas e ser usado para explicar
   partes mais extensas do código.
*/
```

- Dica: **você pode comentar uma linha usando CTRL + /**

Saída de dados

- Em Java, você pode usar a classe **System.out** para imprimir saída de dados no console.

```java
// Imprime no console e pula uma linha
System.out.println("Olá, Mundo!");
System.out.println("Linha 1");
System.out.println("Linha 2");
System.out.println("Linha 3");


// Imprime no console e pula uma linha
System.out.print("Linha 4");
System.out.print("Ainda linha 4")
```

- Usando a dica de autocompletar: **digite syso** em seguida CTRL + SPACE
- Temos também o **Printf** que é utilizado para formatar e imprimir strings no console

```java
// Usando printf para formatar e imprimir
System.out.printf("Nome: %s, Idade: %d%n", "Rogério", 28)
```

- Neste exemplo:
  - %s é um marcador de posição para uma string.
  - %d é um marcador de posição para um número inteiro.
  - %n é um marcado para pular linha

Variável

- Nós declamaos os tipos de dados e atribuímos valores posteriormente através de **variáveis**

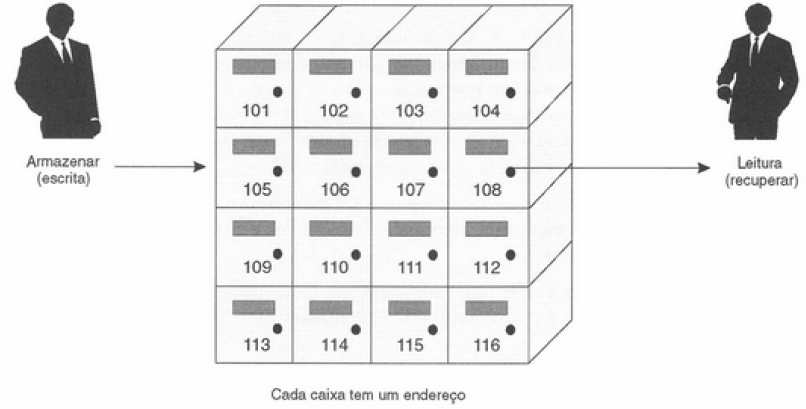
Variável

Uma variável é um espaço de armazenamento nomeado que é usado para representar um valor em um programa de computador.

- As variáveis são fundamentais em programação, pois permitem que você armazene e manipule dados de maneira dinâmica durante a execução de um programa.
- Eu posso ter uma variavel do tipo
  - inteiro
  - decimal
  - cadeira de caracteres
  - binaria (verdadeira ou falsa)

Variável

- Valor armazenado no computador
- Associa um nome a um valor tornando mais fácil lembrar e utilizar este valor posteriormente



- Pode variar com o tempo
- Espaço de memória cujo armazenamos dados e é representado por uma palavra

Tipo char

- O tipo de dado char em Java é usado para representar caracteres individuais
- Cada valor char é armazenado como um código Unicode de 16 bits.
- A codificação de caracteres ASCII e Unicode permite que os computadores armazenem e troquem dados com outros computadores e programas.
- Unifica a representação de caracteres alfanuméricos em computadores

```
char letraA = 'A';
char letraB = 66;  // O valor ASCII de 'B'
```

- 2. Manipulação de Caracteres:

```
char primeiroCaractere = "Hello".charAt(0);  // Obtendo o primeiro caractere da string
```

Tipos de dados numéricos - Inteiros

Como a tabela mostra, todos os tipos inteiros sao valores de sinal positivo e negativo.

Tipo	Tamanho em bits	Intervalo
byte	8	–128 a 127
short	16	–32.768 a 32.767
int	32	–2.147.483.648 a 2.147.483.647
long	64	–9.223.372.036.854.775.808 a 9.223.372.036.854.775.807

- O tipo inteiro mais usado e **int**
- Variaveis de tipo int costumam ser empregadas no controle de lacos, na indexacao de arrays e na execucao de calculos de inteiros para fins gerais.
- Quando voce precisar de um inteiro que tenha um intervalo maior do que o de int, use long.

Tipos primitivos

- Java contem duas categorias gerais de tipos de dados internos: orientados a objetos e nao orientados a objetos.
- Os tipos orientados a objetos sao definidos por classes (Veremos!)
- Na base de Java, temos o termo primitivo que e usado para indicar que esses tipos nao sao objetos no sentido da orientacao a objetos e sim valores binarios comuns.
- Esses tipos primitivos nao sao objetos devido a questoes de eficiencia.
- int é um tipo de dado primitivo em Java.
- Integer é uma classe que encapsula um valor inteiro do tipo primitivo int. Ela fornece métodos úteis para operações com números inteiros

```
// int é primitivo, Integer não
Integer num = 10;
String numeroEmString = "5";
int numero = Integer.parseInt(numeroEmString);

String outroNumeroEmString = "42";
Integer numeroObjeto = Integer.valueOf(outroNumeroEmString);

// double é primitivo, Double não
String numeroEmString = "3.14";
double numero = Double.parseDouble(numeroEmString);
Double = 10.5;
```

Operadores aritméticas

- Um operador e um simbolo que solicita ao compilador que execute uma operacao matematica, logica ou relacional especifica.
- Java tem três classes gerais de operadores: aritmetico, relacional e logico.

Operador	Significado
+	Adição (também mais unário)
–	Subtração (também menos unário)
*	Multiplicação
/	Divisão
%	Módulo
++	Incremento
– –	Decremento

Expressões aritméticas - regras de precedências

Mais alta		
++ (posfixo)	– – (posfixo)	
++ (prefixo)	– – (prefixo)	
*	/	%
+	–	
>	>=	<
==	!=	
&		
^		
|		
&&		
||		
Mais baixa		

Operações relacionais

- Os operadores relacionais são utilizados para comparar valores e expressões, retornando um resultado booleano (verdadeiro ou falso) com base na relação entre esses valores.
- Esses operadores são essenciais para a construção de lógica de controle de fluxo em programas Java.

Operador	Significado
= =	Igual a
!=	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Operações relacionais - Exercício

- Complete a tabela abaixo com True ou False. Considere a = 4, b = 10, c = 5.0, d = 1 e f = 5.

Expressão	Resultado
a == c	❍ True ❍ False
a < b	❍ True ❍ False
d > b	❍ True ❍ False
c! = f	❍ True ❍ False
a == b	❍ True ❍ False
c < d	❍ True ❍ False
b > a	❍ True ❍ False
c >= f	❍ True ❍ False
f >= c	❍ True ❍ False
c <= c	❍ True ❍ False
c <= f	❍ True ❍ False

Operações lógicas

- Os operadores lógicos em Java são utilizados para realizar operações booleanas entre expressões, produzindo resultados que são verdadeiros (true) ou falsos (false).
- Esses operadores são fundamentais para criar condições lógicas e controlar o fluxo de execução em programas Java.

Operador	Significado
&	AND
|	OR
||	OR de curto-circuito
&&	AND de curto-circuito
!	NOT

Operador lógico E

retorna verdadeiro somente quando ambas as condições que está conectando são verdadeiras. Se qualquer uma das condições for falsa, o resultado da operação "E"será falso.

A	B	A e B
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

- Para entrar na festa, você precisa ter mais de 18 anos e apresentar um documento de identidade válido
- O carro só funciona corretamente se houver combustível e o motor estiver em boas condições.
- Para passar na entrevista de emprego, você precisa ter as habilidades necessárias e uma boa apresentação pessoal.
- O sistema só permite o acesso se o usuário inserir o nome de usuário e a senha corretos.

Operador lógico OU

O operador lógico "OU"é utilizado para combinar duas ou mais condições em uma expressão lógica. Ele retorna verdadeiro se pelo menos uma das condições for verdadeira.

A	B	A ou B
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

- Você pode escolher estudar Medicina ou Engenharia como sua carreira universitária.
- O restaurante oferece pratos vegetarianos ou opções para quem prefere carne.
- Você pode optar por fazer exercícios aeróbicos ou levantar pesos para manter a forma.
- O aplicativo de entrega de comida permite pagar com cartão de crédito ou dinheiro em espécie.
- O evento será realizado ao ar livre ou em um local fechado, dependendo das condições climáticas.

Operações lógicas

V_1	V_2	V_1 and V_2
V	V	V
V	F	F
F	V	F
F	F	F

V_1	V_2	V_1 or V_2
V	V	V
V	F	V
F	V	V
F	F	F

p	q	p & q	p | q	!p
Falso	Falso	Falso	Falso	Verdadeiro
Verdadeiro	Falso	Falso	Verdadeiro	Falso
Falso	Verdadeiro	Falso	Verdadeiro	Verdadeiro
Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro	Falso

Operações lógicas - exercicios

- Complete a tabela a seguir utilizando a = True, b = False e c = True.

Expressão	Resultado
a and a	❍ True ❍ False
b and b	❍ True ❍ False
not c	❍ True ❍ False
not b	❍ True ❍ False
not a	❍ True ❍ False
a and b	❍ True ❍ False
b and c	❍ True ❍ False
a or c	❍ True ❍ False
b or c	❍ True ❍ False
c or a	❍ True ❍ False
c or b	❍ True ❍ False
c or c	❍ True ❍ False
b or b	❍ True ❍ False

O operador de atribuicao

```java
int x = 5;
x += 3; // Equivalente a x = x + 3;
System.out.println("x após a adição: " + x); // Saída: 8

int y = 10;
y -= 4; // Equivalente a y = y - 4;
System.out.println("y após a subtração: " + y); // Saída: 6


int z = 3;
z *= 2; // Equivalente a z = z * 2;
System.out.println("z após a multiplicação: " + z); // Saída: 6


int w = 8;
w /= 2; // Equivalente a w = w / 2;
System.out.println("w após a divisão: " + w); // Saída: 4


String texto = "Olá";
texto += " mundo"; // Equivalente a texto = texto + " mundo";
System.out.println(texto); // Saída: Olá mundo
```

Nome de variáveis

- As variáveis são representadas, geralmente, por uma palavra simples ou composta que a identifica e a torna única em seu escopo.
- Podemos definir quaisquer nomes a uma variável, porém, existem algumas poucas regras que devemos seguir
  1. Deve iniciar com uma letra minúscula
  2. Não pode conter espaços
  3. Variáveis não podem começar com números
  4. Se o nome for composto pode-se utilizar para juntar as palavras
  5. Se o nome for composto pode-se utilizar o padrão **camelCase**: primeira palavra minuscula e a segunda começando maiscula

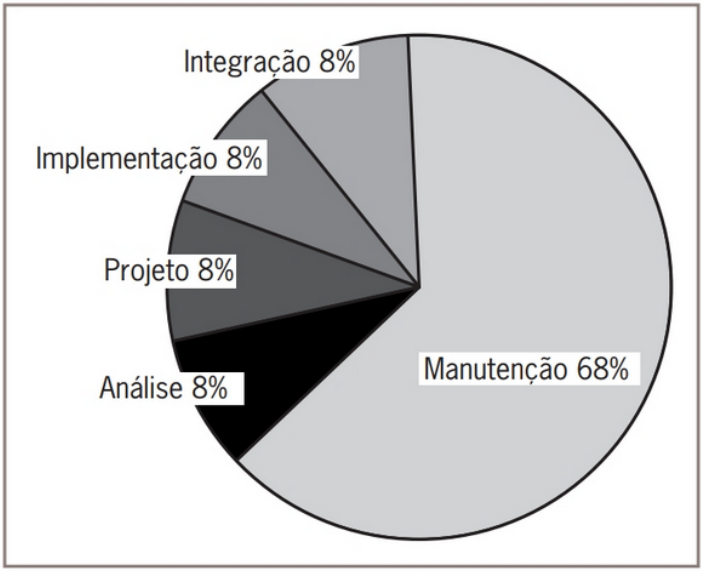
Nome de variáveis

```
int idadeDoUsuario;
double salarioMensal;


// Não tão bom
int x;

// Melhor
int idade;

// Melhor ainda
double salarioMensal;


// Não tão bom
int qty;

// Melhor
int quantidadeProdutos;


// Não permitido (palavra reservada)
int class;
```

Desenvolvimento de software

1. **Pedido do cliente** — Nessa fase, os programadores recebem uma descrição geral de um problema que é potencialmente passível de uma solução computadorizada. Essa etapa também chama-se fase de requisitos do usuário.
2. **Análise** — Os programadores determinam o que o programa fará. Isso às vezes é visto como um processo de esclarecimento das especificações do problema.
3. **Projeto** — Os programadores determinam como o programa realizará sua tarefa.
4. **Implementação** — Os programadores escrevem o programa. Essa etapa também chama-se fase de codificação.
5. **Integração** — Grandes programas têm muitas partes. Na fase de integração, essas partes são reunidas em um todo que funciona harmoniosamente, o que geralmente não é uma tarefa fácil.
6. **Manutenção** — Os programas geralmente têm uma vida longa; uma vida útil de 5 a 15 anos é comum para software. Durante esse período, os requisitos mudam, erros são detectados e modificações menores ou maiores são realizadas.

Desenvolvimento de software



Desenvolvimento de software - Estudo de caso

Solicitação

- O cliente solicita um programa que calcula o imposto de renda de uma pessoa.

Análise

- A análise geralmente exige que o programador aprenda algumas coisas sobre o domínio do problema, neste caso, a legislação tributária relevante.
- Simplificando
  - De todos os contribuintes é cobrada uma taxa de imposto fixa de 20%.
  - Todos os contribuintes têm direito a uma dedução padrão de R$ 10.000.
  - Para cada dependente, o contribuinte tem direito a uma dedução adicional de R$ 3.000.

```
Insira a receita bruta: 150000.00
Insira o número de dependentes: 3
O imposto de renda é de $ 26200,0
```

Projeto

- Durante a análise, especificamos o que um programa fará. Na próxima fase, projeto, descrevemos como o programa fará isso.
- Em geral, isso envolve escrever um algoritmo.

```
Aceite como entrada a renda bruta e o número de dependentes
Calcule o lucro tributável utilizando a fórmula
Renda tributável = renda bruta - 10.000 - (3.000 * número de dependentes)
Calcule o imposto de renda utilizando a fórmula
Imposto = renda tributável * 0,20
Imprima o imposto
```

Desenvolvimento de software - Estudo de caso

Testes

- O verdadeiro desafio é criar conjuntos de entradas que possam revelar um erro. Um erro neste ponto, também chamado erro de lógica ou erro de projeto, é uma saída inesperada.
- Testar um programa com todas as entradas possíveis é uma tarefa geralmente impraticável
- Um programa correto produz a saída esperada para qualquer entrada legítima.
- Importante encontrar padrões de possíveis erros

Número de dependentes	Renda bruta	Imposto esperado
0	10.000	0
1	10.000	–600
2	10.000	–1.200
0	20.000	2.000
1	20.000	1.400
2	20.000	800

- Observe que as saídas negativas não são consideradas erros. Veremos como evitar esses cálculos.