

Programação de Aplicativos

Revisão de Lógica de
Programação

Java

Tipos de Datos

Primitivos:

integer || long

double || float

char

boolean

```
// tipos primitivos  
int a = 1;  
double b = 2.0;  
char c = '9';  
boolean d = true;
```

Tipos de Dados

Compostos:

String

Object

Array

ArrayList (lista de n tipos)

```
// tipos compostos
String e = "um conjunto de caracteres";

// objeto em Javascript
// let f = {
//     chave01: 1;
//     chave02: "valor";
// };

// objeto em java
class f {
    int chave01 = 5;
    double chave02 = 6.2;
}
```

Tipos de Dados

Compostos:

String

Object

Array

ArrayList (lista de n tipos)

```
// array || "ArrayList" em Javascript
// let teste = [10, 20, "texto", 4.0, true]

// array em Java
// Cria um array de inteiros com 5 elementos
int[] numeros = new int[5];

// Atribui valores
numeros[0] = 10;
numeros[1] = 20;
numeros[2] = 30;
numeros[3] = 40;
numeros[4] = 50;

// ArrayList em Java
// Características:
// Tamanho dinâmico
// Pode conter qualquer tipo de objeto
// (use generics <Tipo> para especificar)
import java.util.ArrayList;
ArrayList<String> frutas = new ArrayList<>();
frutas.add("Maçã");
frutas.add("Banana");
frutas.add("Laranja");
```

Tipos de Dados

Atribuição de Valores aos tipos:

Input / Output de dados

```
// atribuição de valores
import java.util.Scanner;

// cria o leitor do terminal
Scanner entrada = new Scanner(System.in);

int g;
System.out.print("Digite um valor para 'g': ");
// lê o valor digitado e armazena em 'g'
g = entrada.nextInt();

System.out.println("O valor de 'g' é: " + g);

// System.out.print(); - Não pula linha
// System.out.println(); - Pula linha
```

Operadores

Aritméticos:

Soma (+) - Atenção: Pode ser
Adição OU Concatenação

Subtração (-)

Multiplicação (*)

Divisão (/)

Módulo (%)

Operadores

Relacionais:

Igual (== OU ===)

Atenção: apenas um "=" serve para atribuição

Diferente (!=)

Maior que (>)

Menor que (<)

Maior ou igual a (>=)

Menor ou igual a (<=)

Operadores

Lógicos:

E (and - &)

OU (or - ||)

NÃO (not - ! - !=)

Estruturas de Controle

Condicionais:

if

if else

else

switch / case

```
int valorDeTeste = 10;

if(valorDeTeste < 10){
    System.out.println("entrou no if");
}
else if(valorDeTeste > 10){
    System.out.println("entrou no else if");
}
else{
    System.out.println("entrou no else");
}
```

Estruturas de Controle

Condicionais:

if
if else
else
switch / case

```
int dia = 3;

switch (dia) {
    case 1:
        System.out.println("Domingo");
        break;
    case 2:
        System.out.println("Segunda");
        break;
    case 3:
        System.out.println("Terça");
        break;
    default:
        System.out.println("Dia inválido");
}
```

Estruturas de Controle

Repetição:

- for - Repete i vezes
- while - Verifica primeiro
- do...while - Executa primeiro

```
for(int i = 0; i < 10; i++){  
    System.out.println("Contagem: " + i);  
}
```

```
int contador = 1;  
  
while (contador <= 5) {  
    System.out.println("Contagem: " + contador);  
    contador++;  
}
```

```
int numeroVariavel = 1;  
  
do {  
    System.out.println("Número: " + numeroVariavel );  
    numeroVariavel ++;  
} while (numeroVariavel <= 5);
```

Estruturas de Controle

Controle:

break - Interrompe imediatamente a execução do laço ou switch mais próximo.

continue - Pula para a próxima iteração do laço, ignorando o restante do código dentro dele.

return - Encerra a execução de uma função e, opcionalmente, retorna um valor.

```
for (int i = 1; i <= 10; i++) {  
    if (i == 5) {  
        break; // Sai do laço quando i == 5  
    }  
    System.out.println("i = " + i);  
}  
//Não irá imprimir o 5 e próximos
```

Estruturas de Controle

Controle:

break - Interrompe imediatamente a execução do laço ou switch mais próximo.

continue - Pula para a próxima iteração do laço, ignorando o restante do código dentro dele.

return - Encerra a execução de uma função e, opcionalmente, retorna um valor.

```
for (int i = 1; i <= 5; i++) {  
    if (i == 3) {  
        continue; // Pula o número 3  
    }  
    System.out.println("Número: " + i);  
}  
//Não imprime o 3, mas continua no loop
```

Estruturas de Controle

Controle:

break - Interrompe imediatamente a execução do laço ou switch mais próximo.

continue - Pula para a próxima iteração do laço, ignorando o restante do código dentro dele.

return - Encerra a execução de uma função e, opcionalmente, retorna um valor.

```
public class ExemploReturn {  
    public static void main(String[] args) {  
        System.out.println("Soma: " + somar(3, 4));  
        somarSemRetorno(3, 4); // apenas chama o método  
    }  
  
    public static int somar(int a, int b) {  
        return a + b; // Retorna o resultado da soma  
    }  
  
    public static void somarSemRetorno(int a, int b) {  
        int soma = a + b;  
        return; // Retorna "void" (opcional)  
    }  
}
```

Estruturas de Dados

Vetor (matriz unidimensional)

Martiz (jogo da velha / xadrez)

Registro (structs - Classes em Java) (tipo personalizado)

```
public class ExemploVetor {  
    public static void main(String[] args) {  
        int[] numeros = {10, 20, 30, 40, 50};  
  
        for (int i = 0; i < numeros.length; i++) {  
            System.out.println("Elemento " + i + ": " + numeros[i]);  
        }  
    }  
}
```

Estruturas de Dados

Vetor (matriz unidimensional)

Martiz (jogo da velha / xadrez)

Registro (structs - Classes em Java) (tipo personalizado)

```
public class ExemploMatriz {  
    public static void main(String[] args) {  
        int[][] matriz = {  
            {1, 2, 3},  
            {4, 5, 6}  
        };  
  
        for (int i = 0; i < matriz.length; i++) {  
            for (int j = 0; j < matriz[i].length; j++) {  
                System.out.print(matriz[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```


Exercício de Fixação

- a) Resolver alguma das sugestões
- b) Fique a vontade para desenvolver uma nova ideia



Anagrama: o jogo embaralha as letras de uma palavra e o jogador deve descobrir qual é a palavra correta dentro de n tentativas

Sorteador de dados: o jogador escolhe quantos dados lançar e o programa sorteia resultados aleatórios

Pedra, papel e tesoura: o jogador escolhe uma dentre as três opções e o computador sorteia outra, apresentando o vencedor

Adivinhe o número: o jogo escolhe um número aleatório secreto (dentro de um "range" escolhido) e o jogador tenta adivinhar com base em dicas de "maior" ou "menor"

Jogo da velha: dois jogadores tentam formar uma linha com três símbolos iguais - X ou O - em um tabuleiro 3x3 - **DESAFIO**

Genius ou Simon: o jogo mostra uma sequência de cores (nesse caso, números) em ordem aleatória e limpa o console alguns segundos depois. O jogador deve memorizar e repetir a sequência na mesma ordem - **DESAFIO**