



INTERFACES JAVA DESKTOP

by Rafa

rcalves@senacrs.com.br



02

Sintaxe Java

Revisão

01

Tipos

02

Laços

03

Classes

04

Estruturas
de dados

05

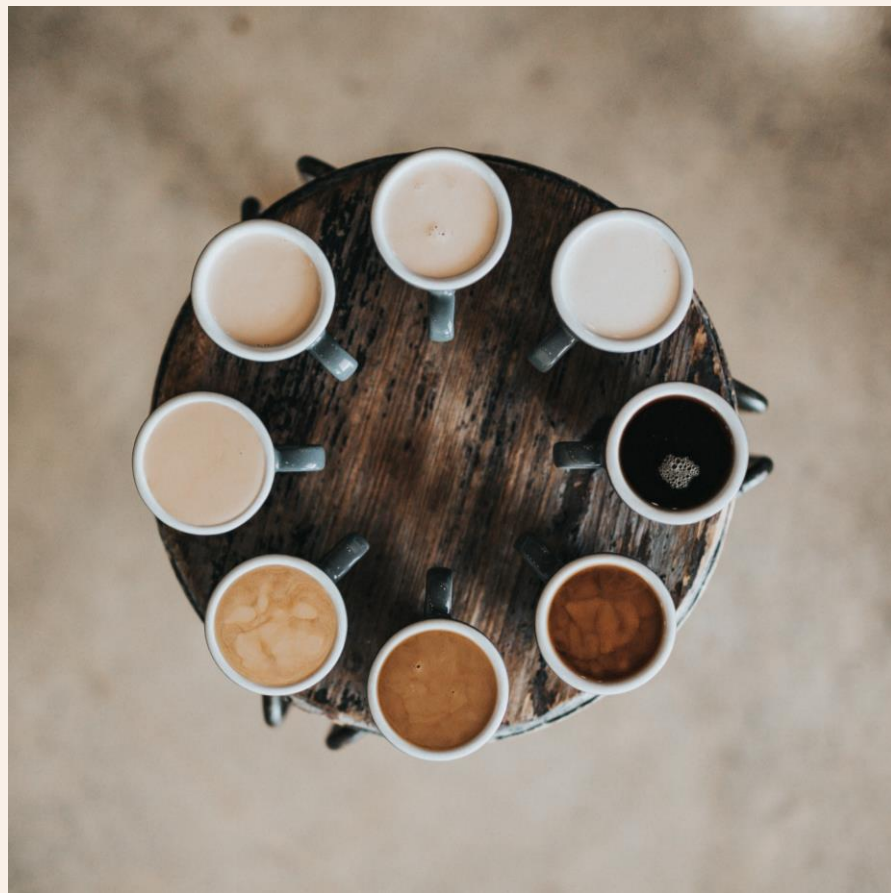
Modificadores
de acesso

06

Entradas
e saídas

Tipos

01



TIPOS PRIMITIVOS

boolean	Armazena valores lógicos: Verdadeiro ou Falso	false	1 bit
char	Armazena um único caractere Unicode	'\u0000'	2 byte
byte	Armazena números inteiros	0	1 byte
short	Armazena números inteiros	0	2 byte
int	Armazena números inteiros	0	4 byte
long	Armazena números inteiros	0L	8 byte
float	Armazena números em ponto flutuante com 8 decimais de precisão	0.0f	4 byte
double	Armazena números em ponto flutuante com 16 decimais de precisão	0.0 ou 0.0d	8 byte

TIPOS PRIMITIVOS

```
//BOOLEAN
boolean booleanVar = true;

//INT
int intVar = 1;

//DOUBLE
double doubleVar = 3.14;
```

TIPOS DE REFERÊNCIA

```
//OBJECT
Object objectVar = new Object();

//STRING
String stringVar = "String";

//ARRAY
String[] stringArray = new String[10]();
int[] intArray = {0,1,2};

//CLASSE CUSTOM
Classe classeVar = new Classe();
```

CLASSES WRAPPER

```
//BOOLEAN
Boolean booleanVar = true;
booleanVar = Boolean.TRUE;

//INTEGER
Integer integerVar = 1;
integerVar = null;

//DOUBLE
Double doubleVar = 3.14;
```




Laços

02

FOR FOREACH

```
//FOR
for(int i = 0; i < 10; i++){
    // bloco de código
}

//FOREACH
int[] numeros = {1, 2, 3, 4, 5};
for (int numero : numeros) {
    // bloco de código
}
```

Classes

03



CLASSE CUSTOM

```
public class Pessoa {  
    private String nome;  
    public Pessoa(String nome) {  
        this.nome = nome;  
    }  
  
    public String getName() {  
        return this.nome;  
    }  
}
```



Estruturas de dados 04

LISTA

```
//LIST
List<String> listaDeNomes = new ArrayList<>();
listaDeNomes.add("João");
listaDeNomes.add("Maria");

listaDeNomes.remove(0);
listaDeNomes.remove("Maria");
```

SET



```
//SET - Funcionamento semelhante à lista  
//      - armazena apenas valores únicos
```

```
Set<String> listaDeNomes = new HashSet<>();  
listaDeNomes.add("João");  
listaDeNomes.add("Maria");  
  
listaDeNomes.remove("Maria");
```

FILA

```
//QUEUE - FIFO
//      - primeiro a entrar, primeiro a sair

Queue<String> listaDeNomes = new LinkedList<>();
listaDeNomes.add("João");
listaDeNomes.add("Maria");

listaDeNomes.peek(); // apenas retorna
listaDeNomes.poll(); // retorna e remove
```

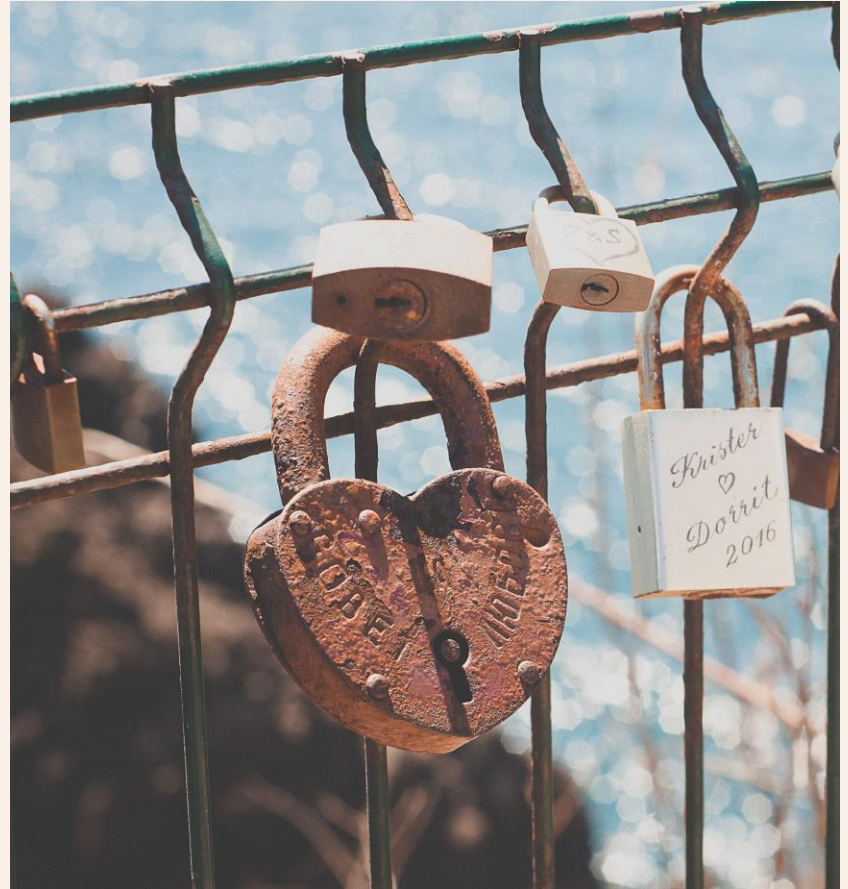

PILHA


```
//STACK – LIFO  
//      – último a entrar, primeiro a sair
```

```
Stack<String> listaDeNomes = new Stack<>();  
listaDeNomes.add("João");  
listaDeNomes.add("Maria");
```

```
listaDeNomes.peek(); // apenas retorna  
listaDeNomes.poll(); // retorna e remove
```

Modificadores de acesso 05



MODIFICADOR	ACESSO INTERNO	ACESSO PACOTE	ACESSO FILHAS	ACESSO EXTERNO
PRIVATE				
DEFAULT				
PROTECTED				
PUBLIC				



Entradas e saídas 06

IMPRESSÃO SAÍDA



```
//OUT  
System.out.println("Hello World!");  
int x = 10;  
System.out.printf("O valor é: %d\n", x);
```

SCANNER ENTRADA



```
//SCANNER  
Scanner scanner = new Scanner(System.in);  
int x = scanner.nextInt();  
String y = scanner.nextLine();
```

Revisão

01

Tipos

02

Laços

03

Classes

04

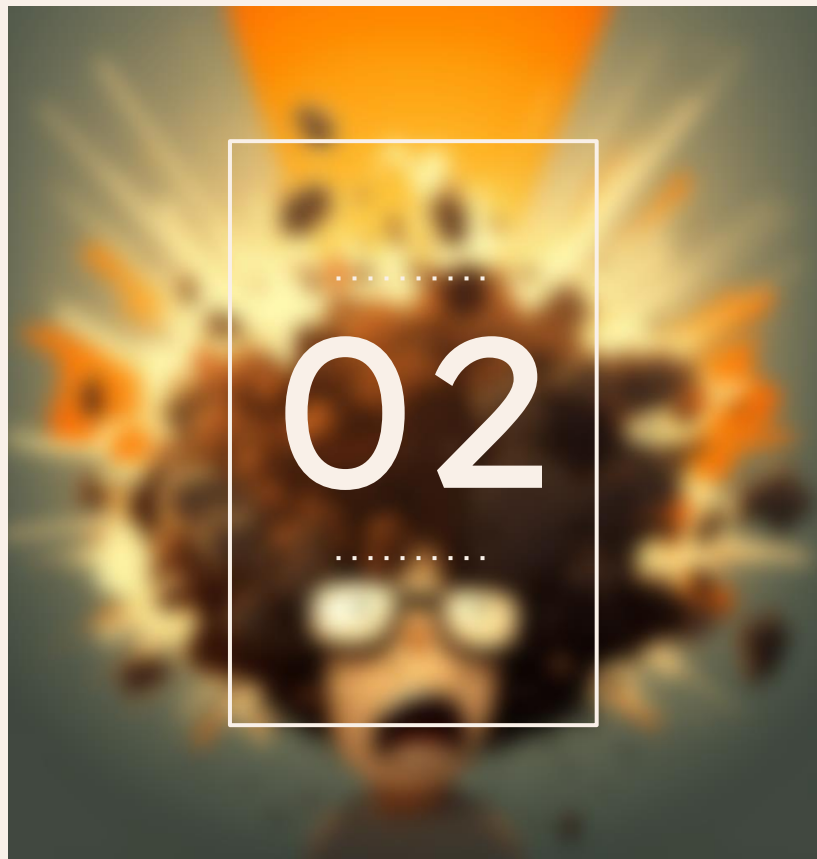
Estruturas
de dados

05

Modificadores
de acesso

06

Entradas
e saídas



**OBRIGADO PELA
COMPANHIA!**

by Rafa

rcalves@senacrs.com.br