

# Programação Orientada a Objetos

Thiago Nunes de Sousa  
thiagonunes.tns@gmail.com



# Apresentação

- **Thiago Nunes de Sousa**
  - [thiagonunes.tns@gmail.com](mailto:thiagonunes.tns@gmail.com)
  - <https://github.com/thiagotns>



# Programação Orientada a Objetos

- **Declaração de Variáveis**
- **Tipos Primitivos**
- **Operadores**
- **Cast**
- **Decisão**
- **Laços**
- **Escopo de Variável**
- **Exercícios**



# Declaração de Variável

- **Toda variável precisa ser declarada antes de ser utilizada**
- **Como declarar ?**
  - tipo nomeVariavel;
    - int ano;
  - tipo nomeVariavel = valor;
    - int ano = 2017;



# Identificadores

- **Os nome de uma variável (identificador):**
  - sempre começa por letra, sublinhado ou cifrão
  - não pode começar com números
  - não pode ser uma palavra reservada
  - são “case sensitive”
    - Diferenciam maiúscula de minúscula
    - nomeVariavel <> nomevariavel
  - do segundo caractere em diante, pode conter qualquer sequência de letras, dígitos, sublinhados ou cifrões



# Palavras Reservadas

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while



# Identificadores

- **Válidos**

- nomeALuno
- \_nome
- \$nome
- \_\$nome\_aluno
- Tênis
- Nome

- **Inválidos**

- :nome
- -nome
- Idade#
- 4g
- short



# Padrão de Nomes

- **Convenção de Código Java**

- Padronizar a nomenclatura de diferentes sistemas e bibliotecas
- Facilitar o desenvolvimento

- **Identificadores**

- Sempre iniciar com minúscula
- CamelCase para nomes compostos
- Utilizar nomes curtos e significativos
- Não utilizar \_





# Padrões de Nomes

- **Exemplos de bons nomes de identificadores**
  - nomeCompleto
  - dataNascimento
  - prazoMaximoDeEnvio



# Tipos Primitivos

- **Java é fortemente tipada**
  - Toda variável tem que ter um tipo

Tipos	Primitivo	Valores possíveis		Valor Padrão	Tamanho	Exemplo
		Menor	Maior			
Inteiro	byte	-128	127	0	8 bits	byte ex1 = (byte)1;
	short	-32768	32767	0	16 bits	short ex2 = (short)1;
	int	-2.147.483.648	2.147.483.647	0	32 bits	int ex3 = 1;
	long	-9.223.372.036.854.770.000	9.223.372.036.854.770.000	0	64 bits	long ex4 = 1L;
Ponto Flutuante	float	-1,4024E-37	3.40282347E + 38	0	32 bits	float ex5 = 5.50f;
	double	-4,94E-307	1.79769313486231570E + 308	0	64 bits	double ex6 = 10.20d; ou double ex6 = 10.20;
Caractere	char	0	65535	\0	16 bits	char ex7 = 194; ou char ex8 = 'a';
Booleano	boolean	false	true	false	1 bit	boolean ex9 = true;



# Operadores

- **Mais usados**

- Adição (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)
- Resto (%)
- Atribuição com operação
  - += -= \*= /= %=



# Operadores

- **Incremento**

- ++a ou --a
  - Incrementa ou decrementa antes de usar a variável
- a++ ou a--
  - Incrementa ou decrementa depois de usar a variável



# Operadores

- **Relacionais**

- ==
- !=
- < e <=
- > e >=

- **Lógicos**

- && → E (and)
- || → Ou (or)
- ! → Negação (not)



# Cast

- **“Trocar” o tipo de uma variável**
- **Só é permitido entre tipos compatíveis**
  - Feita automaticamente do menor para o maior
  - Caso contrário, é preciso “forçar”



# Cast

- **Implícito**

```
int a = 10;
```

```
long b = a;
```

- **Explícito**

```
double a = 3.14;
```

```
int b = (int) a;
```

- `//b==3` (o valor flutuante é truncado)



# Mão na Massa





# IF ELSE

```
if(condicao){  
    codigo  
} else {  
    outro codigo  
}
```

```
if(valor==1){  
    valor++;  
} else{  
    valor--;  
}
```



# While

```
int idade = 15;  
while (idade < 18) {  
    System.out.println(idade);  
    idade = idade + 1;  
}
```



# FOR

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



# Como ler uma entrada?

```
Scanner scanner = new Scanner(System.in);
```

```
System.out.println("Entre com o número");
```

```
int entrada = scanner.nextInt();
```

```
System.out.println("Entre com o texto");
```

```
String str = scanner.next();
```

```
System.out.println("Número digitado: " + entrada);
```

```
System.out.println("Texto digitado: " + str);
```



# Mão na Massa

- **Imprimir todos os números de 1 a 1000**
- **Imprimir todos os números de 150 a 300**
- **Imprimir todos os números múltiplos de 3 entre 1 e 100**
- **Calcular Fatorial**
- **Calcular Fibonacci**



# Mão na Massa

- **Imprimir a soma de todos os números múltiplos de 3 e de 5 que são menores que 1000.**
- **Ler 3 números e imprimi-los em ordem crescente**



# Mão na Massa

- **Faça um programa para calcular o Máximo Divisor Comum entre 2 números passados como argumentos para a execução do programa. Observe as seguintes propriedades do MDC:**
  - $\text{MDC}(x, y) = \text{MDC}(x - y, y)$ , se  $x > y$ ;
  - $\text{MDC}(x, y) = \text{MDC}(y, x)$ ;
  - $\text{MDC}(x, x) = x$ .
- **Exemplo:**  
 $\text{MDC}(3, 5) = \text{MDC}(5, 3) = \text{MDC}(2, 3) = \text{MDC}(3, 2) = \text{MDC}(1, 2) = \text{MDC}(2, 1) = \text{MDC}(1, 1) = 1$



# Vetores (arrays)

- `int[] valores = new int[10];`
- `valores[1] = 5;`
- `valores.length` → tamanho do vetor
  
- `int[][] matriz = new int[10][10];`





# Vetores (arrays)

- **for(int i=0; i<10; i++)**  
    System.out.print(valores[x]);
- **for(int x : valores)**  
    System.out.print(x);



# Mão na massa

- **Dado um vetor de inteiros, verificar se um número informado pelo usuário está no vetor;**
- **Dado um vetor, imprimir todos os seus números pares;**
- **Criar programa para ler do teclado e armazenar o valor de 10 números inteiros em um vetor (e imprimi-los)**



# Mão na Massa

- **Dado um vetor de 10 números inteiros, verificar se ele está ordenado**
- **Dado um vetor de char, verificar se este é um palíndromo**
- **Inverter os elementos de um vetor**



# Programação Orientada a Objetos

Thiago Nunes de Sousa  
thiagonunes.tns@gmail.com

