

Seção 07: Outros Tópicos

Restrições para nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não usar acentos ou til
- Não pode ter espaço em branco
- Sugestão: use nomes que tenham um significado

Convenções

- **Camel Case: lastName**
 - pacotes
 - atributos
 - métodos
 - variáveis e parâmetros
- **Pascal Case: ProductService**
 - classes

```
package exercicios_propostos.s9_pilares_poo;

public class ContaBancaria {
    private long numero;
    private String titular;
    private double saldo;

    public ContaBancaria(long numero, String titular, double depositoInicial) {
        this.numero = numero;
        this.titular = titular;
        depositar(depositoInicial);
    }

    public long getNumero() {
        return numero;
    }

    public String getTitular() {
        return titular;
    }
}
```

Operadores bitwise

Operador	Significado
&	Operação "E" bit a bit
	Operação "OU" bit a bit
^	Operação "OU-exclusivo" bit a bit

Operações com Strings

```

package anotacoes;

public class Anotações {
    public static void main(String[] args) {
        String texto = "O rato roeu a roupa do rei de roma"; // texto original

        // Converte para minúsculas -> saída: o rato roeu a roupa do rei de roma
        String lower = texto.toLowerCase();

        // Converte para maiúsculas -> saída: O RATO ROEU A ROUPA DO REI DE ROMA
        String upper = texto.toUpperCase();

        // Remove espaços em branco no início e fim da string
        String trim = texto.trim();

        // Extraí substring a partir do índice 2 -> saída: rato roeu a roupa do rei de roma
        String subString = texto.substring(2);

        // Extraí substring do índice 2 até 12 -> saída: rato roeu a r
        String subString2 = texto.substring(2,13);

        // Substitui todas ocorrências de 'r' por 'p' -> saída: O pato poeu a poupa do pei de po
        String replace = texto.replace('r', 'p');

        // Retorna o índice da primeira ocorrência do caractere 'a' -> saída: 4
        int indice = texto.indexOf('a');

        // Retorna o índice onde começa a substring "oma" -> saída: 33
        int indice2 = texto.indexOf("oma");

        // Retorna o índice da última ocorrência do caractere 'r' -> saída: 25
        int indice3 = texto.lastIndexOf('r');

        // Divide a string em um array usando espaço como separador
        // saída: ["O", "rato", "roeu", "a", "roupa", "do", "rei", "de", "roma"]
        String[] strSplit = texto.split(" ");
    }
}

```

Funções

- Representam um processamento com significado.
- Exemplos: `Math.sqrt(double)` e `System.out.println(string)`.
- Vantagens: modularização, delegação e reaproveitamento.
- Podem receber dados de entrada (parâmetros) e podem ou não retornar um resultado.
- Em orientação a objetos, funções dentro de classes são chamadas de "métodos".