

# Java — Exemplos por Tópico

Documento com exemplos práticos e código formatado para cada tópico pedido. Cada seção contém uma breve explicação e um exemplo Java compilável (quando aplicável).

---

## 1. Hello World

**Explicação:** Programa mínimo que imprime "Hello, World!".

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

## 2. Variáveis e Tipos de Dados

**Explicação:** Declaração de variáveis primárias e operadores de entrada/saída simples.

```
public class VariaveisTipos {  
    public static void main(String[] args) {  
        int idade = 30;  
        double salario = 2500.75;  
        boolean ativo = true;  
        char inicial = 'C';  
        String nome = "Caio";  
  
        System.out.println(nome + " (" + inicial + ") tem " + idade + " anos  
e ganha R$ " + salario);  
        System.out.println("Ativo? " + ativo);  
    }  
}
```

## 3. Operadores

**Explicação:** Operadores aritméticos, relacionais e lógicos.

```
public class Operadores {  
    public static void main(String[] args) {  
        int a = 10, b = 3;  
        System.out.println("Soma: " + (a + b));  
    }  
}
```

```

        System.out.println("Divisão inteira: " + (a / b));
        System.out.println("Resto: " + (a % b));
        System.out.println("Maior? " + (a > b));
        System.out.println("E lógico: " + (a > b && b > 0));
    }
}

```

## 4. Estruturas de Controle (if, else, switch)

```

public class EstruturasControle {
    public static void main(String[] args) {
        int nota = 85;
        if (nota >= 90) {
            System.out.println("A");
        } else if (nota >= 75) {
            System.out.println("B");
        } else {
            System.out.println("C");
        }

        String dia = "Terça";
        switch (dia) {
            case "Segunda":
                System.out.println("Início da semana");
                break;
            case "Terça":
                System.out.println("Continua");
                break;
            default:
                System.out.println("Outro dia");
        }
    }
}

```

## 5. Loops (for, while, do-while)

```

public class Loops {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            System.out.println("for: " + i);
        }

        int j = 0;
        while (j < 3) {
            System.out.println("while: " + j);
        }
    }
}

```

```

        j++;
    }

    int k = 0;
    do {
        System.out.println("do-while: " + k);
        k++;
    } while (k < 2);
}
}

```

## 6. Arrays

```

import java.util.Arrays;

public class ArraysExample {
    public static void main(String[] args) {
        int[] numeros = {1, 2, 3, 4, 5};
        System.out.println(Arrays.toString(numeros));

        for (int n : numeros) {
            System.out.println(n);
        }
    }
}

```

## 7. Métodos

```

public class Metodos {
    public static void main(String[] args) {
        System.out.println(somar(3, 4));
        imprimirMensagem("Olá métodos!");
    }

    public static int somar(int a, int b) {
        return a + b;
    }

    public static void imprimirMensagem(String msg) {
        System.out.println(msg);
    }
}

```

## 8. Classes e Objetos

```
public class Pessoa {
    String nome;
    int idade;

    public Pessoa(String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
    }

    public void apresentar() {
        System.out.println("Olá, sou " + nome + " e tenho " + idade + "
anos.");
    }

    public static void main(String[] args) {
        Pessoa p = new Pessoa("Ana", 28);
        p.apresentar();
    }
}
```

---

## 9. Herança

```
class Animal {
    public void comer() {
        System.out.println("Animal comendo");
    }
}

class Cachorro extends Animal {
    public void latir() {
        System.out.println("Au au");
    }
}

public class HerancaDemo {
    public static void main(String[] args) {
        Cachorro c = new Cachorro();
        c.comer();
        c.latir();
    }
}
```

---

## 10. Polimorfismo

```
class Veiculo {
    public void mover() {
        System.out.println("Veículo se movendo");
    }
}

class Carro extends Veiculo {
    @Override
    public void mover() {
        System.out.println("Carro dirigindo");
    }
}

public class PolimorfismoDemo {
    public static void main(String[] args) {
        Veiculo v = new Carro(); // referência de superclasse para subclasse
        v.mover(); // chama o método sobrescrito
    }
}
```

---

## 11. Interfaces

```
interface Operacao {
    int executar(int a, int b);
}

class Soma implements Operacao {
    public int executar(int a, int b) {
        return a + b;
    }
}

public class InterfaceDemo {
    public static void main(String[] args) {
        Operacao op = new Soma();
        System.out.println(op.executar(5, 7));
    }
}
```

---

## 12. Encapsulamento

```
public class Conta {
    private double saldo;

    public Conta(double saldoInicial) {
        this.saldo = saldoInicial;
    }

    public double getSaldo() {
        return saldo;
    }

    public void depositar(double valor) {
        if (valor > 0) this.saldo += valor;
    }
}

class EncapsulamentoDemo {
    public static void main(String[] args) {
        Conta c = new Conta(100.0);
        c.depositar(50.0);
        System.out.println("Saldo: " + c.getSaldo());
    }
}
```

## 13. Exceções

```
public class Excecoes {
    public static int dividir(int a, int b) {
        if (b == 0) throw new ArithmeticException("Divisão por zero");
        return a / b;
    }

    public static void main(String[] args) {
        try {
            System.out.println(dividir(10, 0));
        } catch (ArithmeticException e) {
            System.out.println("Erro: " + e.getMessage());
        } finally {
            System.out.println("Bloco finally executado");
        }
    }
}
```

## 14. Coleções (List, Set, Map)

```
import java.util.*;

public class Colecoes {
    public static void main(String[] args) {
        List<String> lista = new ArrayList<>();
        lista.add("A"); lista.add("B");

        Set<Integer> conjunto = new HashSet<>();
        conjunto.add(1); conjunto.add(2);

        Map<String, Integer> mapa = new HashMap<>();
        mapa.put("um", 1);
        mapa.put("dois", 2);

        System.out.println(lista);
        System.out.println(conjunto);
        System.out.println(mapa.get("dois"));
    }
}
```

## 15. Generics

```
public class Caixa<T> {
    private T conteudo;

    public void guardar(T conteudo) { this.conteudo = conteudo; }
    public T abrir() { return conteudo; }

    public static void main(String[] args) {
        Caixa<String> c = new Caixa<>();
        c.guardar("Olá");
        System.out.println(c.abrir());
    }
}
```

## 16. Lambda Expressions

```
import java.util.*;

public class LambdaDemo {
    public static void main(String[] args) {
        List<Integer> nums = Arrays.asList(1, 2, 3, 4);
    }
}
```

```
        nums.forEach(n -> System.out.println(n * 2));
    }
}
```

---

## 17. Streams

```
import java.util.*;

public class StreamsDemo {
    public static void main(String[] args) {
        List<String> nomes = Arrays.asList("Ana", "Bia", "Carlos", "Ana");
        nomes.stream()
            .filter(n -> n.startsWith("A"))
            .distinct()
            .forEach(System.out::println);
    }
}
```

---

## 18. Threads

```
public class ThreadExample {
    public static void main(String[] args) {
        Thread t = new Thread(() -> {
            for (int i = 0; i < 5; i++) {
                System.out.println("Thread: " + i);
                try { Thread.sleep(200); } catch (InterruptedException e) { }
            }
        });
        t.start();

        for (int i = 0; i < 3; i++) System.out.println("Main: " + i);
    }
}
```

---

## 19. Annotations

```
import java.lang.annotation.*;

@Retention(RetentionPolicy.RUNTIME)
@interface Info {
    String autor();
    String versao() default "1.0";
}
```



```

}

@Info(autor = "Caio", versao = "1.2")
public class AnnotationsDemo {
    public static void main(String[] args) {
        Info info = AnnotationsDemo.class.getAnnotation(Info.class);
        System.out.println("Autor: " + info.autor() + ", versao: " +
info.versao());
    }
}

```

## 20. Java I/O

```

import java.io.*;

public class IOExample {
    public static void main(String[] args) {
        String nomeArquivo = "exemplo.txt";
        try (BufferedWriter w = new BufferedWriter(new
FileWriter(nomeArquivo))) {
            w.write("Linha 1\nLinha 2");
        } catch (IOException e) {
            e.printStackTrace();
        }

        try (BufferedReader r = new BufferedReader(new
FileReader(nomeArquivo))) {
            String linha;
            while ((linha = r.readLine()) != null) {
                System.out.println(linha);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

*Fim dos exemplos.*