

# Resumo: Classes Genéricas em Java

## Resumo Teórico

O que são classes genéricas?

Classes genéricas permitem criar estruturas reutilizáveis e seguras em Java, usando tipos genéricos definidos por parâmetros como <T>, <K>, <V>, etc.

Exemplo básico:

```
class Caixa<T> {  
    private T conteudo;  
    public void guardar(T item) { conteudo = item; }  
    public T pegar() { return conteudo; }  
}
```

Uso:

```
Caixa<String> caixa1 = new Caixa<>();  
caixa1.guardar("Texto");
```

```
Caixa<Integer> caixa2 = new Caixa<>();  
caixa2.guardar(123);
```

Vantagens:

- Reutilização de código para diferentes tipos de dados
- Segurança de tipo em tempo de compilação
- Evita castings desnecessários

Outros usos comuns:

- Métodos genéricos: `public <T> void imprimir(T valor)`
- Interfaces genéricas: `Comparable<T>`, `Iterable<T>`
- Coleções: `List<String>`, `Map<Integer, String>`

## Resumo: Classes Genéricas em Java

Tipos multiplos:

```
class Par<K, V> { ... }
```

Tipos limitados (bounded types):

```
class CaixaNumerica<T extends Number> { ... }
```

Atencao:

- Generics nao funcionam com tipos primitivos (int, double, etc) - use wrappers (Integer, Double)

### Exercícios

Exercicios para fixar:

1. Crie uma classe generica chamada "Mochila<T>" que armazene um item.
2. Implemente metodos "guardar" e "retirar".
3. Use a classe com String, Integer e uma classe sua personalizada (ex: Livro).
4. Crie um metodo generico que recebe um array e imprime os elementos.
5. Crie uma classe Par<K, V> que armazena dois objetos relacionados.