

# Coleções (Generics)

DigitalHouse>



**Certified Tech  
Developer**

The Ultimate Degree

# Índice

1. Introdução
2. Soluções utilizando Generics

# 1 | Introdução

# Introdução

Aprenderemos, nesta apresentação, o uso da programação paramétrica em **coleções**. Para começar, vamos lembrar que em todas as operações que podemos fazer em coleções, o tipo usado é **Object**.

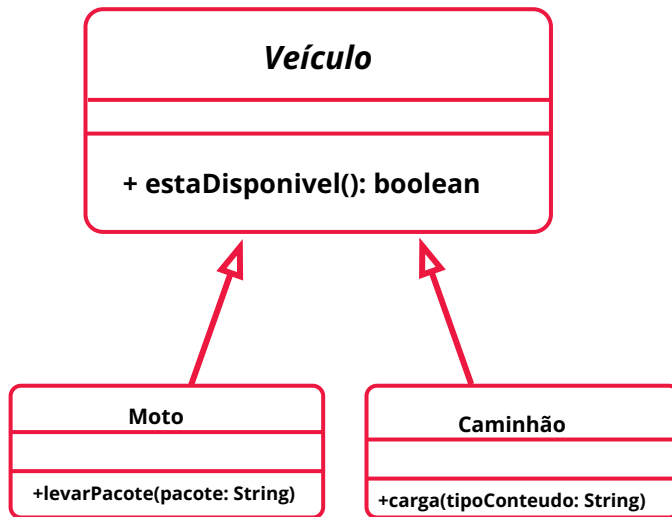
- add(**Object** o) : void
- get(int i) : **Object**
- iterator() : iterator
- hasNext() : boolean
- next() : **Object**



Como em Java, todas as nossas classes herdam de Object, então podemos misturar objetos de diferentes tipos na mesma coleção.

# Introdução

Para entender melhor o contexto, vamos supor que temos uma série de **veículos** onde nossa empresa deve **gerenciar** o que eles transportam.



# Análise

Poderíamos fazer uma lista com os veículos da nossa empresa.

```
List veiculos = new ArrayList();  
  
Moto moto = new Moto();  
Caminhao caminhao = new Caminhao();  
  
veiculos.add(moto);  
veiculos.add(caminhao);
```



# Análise

Se a qualquer momento quisermos obter um veículo da lista, devemos fazer conforme o exemplo abaixo.

```
Moto moto = (Moto) veiculos.get(0);  
Caminhao caminhao = (Caminhao) veiculos.get(1);
```



# Análise

Se quiséssemos acessar a lista de veículos e ver se eles estão disponíveis, poderíamos fazer assim:

```
for(Object o :veiculos) {  
    System.out.println(((Veiculo)o).estaDisponivel());  
}
```



# Análise

Mas, se quisermos adicionar alguma carga ao **caminhão**, é necessário inserir apenas naqueles elementos da lista que são caminhões. Caso contrário, teríamos um **erro de tempo de execução**.

```
for(Object o :veiculos) {  
    if( o instanceof Caminhao)  
        ((Caminhao)o).carga("batata");  
}
```

**2**

## **Solução utilizando Generics**

## Solução utilizando Generics

Para evitar a mistura de objetos de **diferentes tipos** em uma coleção, a partir do Java versão 1.5, todas as coleções podem receber o tipo como parâmetro, ou seja, suportam **Generics**.

```
List<Caminhao> veiculos = new ArrayList<Caminhao>();
```

## Solução utilizando Generics

Se utilizarmos as coleções parametrizando seu tipo, teremos um **controle em tempo de compilação** sobre os tipos dos objetos que adicionamos à coleção, de forma que, ao executar a coleção, não seja necessário verificar a instância de uma vez que não devemos lançar, porque não podemos "misturar" tipos de objetos.

```
for(Caminhao o :veiculos) {  
    o.carga("batatas");  
}
```

DigitalHouse>  
Coding School