

Java Orientado a Objetos


Tipos Genéricos



Java Orientado a Objetos

Tipos Genéricos



	
É um perigo em potencial para uma ClassCastException	Permite que uma única classe trabalhe com uma grande <u>variedade de tipos</u>
Torna nossos códigos mais poluídos e menos legíveis	É uma forma natural de eliminar a necessidade de se fazer cast
Destrói benefícios de uma linguagem com tipos fortemente definidos	Preserva benefícios da checagem de tipos

Declarando uma Classe utilizando Generics

Classe não trabalha com nenhuma referência a um tipo específico

```
public class ManipulaArray<T> { // Contem o parâmetro <?>

    private T[] array; // atributo de tipo generic

    public ManipulaArray( T[] array ) { // Construtor
        this.array = array;
    }

    public boolean existeElemento(T elementoABuscar) {
        for (T elemento : array) {
            if (elemento.equals(elementoABuscar)) {
                return true;
            }
        }
        return false;
    }

    // get e set de atributo generico
    public T[] getArray() { return array;}
    public void setArray(T[] array) {this.array = array;}
}
```

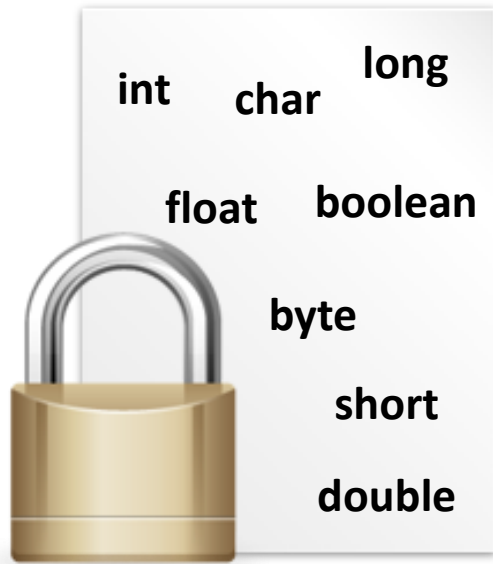
Indica que a classe declarada é uma classe Generics

Declarando Métodos Genéricos



Limitação “Primitiva”

Tipos **Generics** do Java são restritos a tipos de referência (**objetos**) e não funcionarão com tipos primitivos



Solução



Utilizar Classes
Wrapper(Empacotadoras)



Limitando Genéricos

```
public class ColecaoBichoFelino<T extends Felino> {  
    T[] animais;  
}
```



Leão



Gato



Coringa <?>

<? extends T>

Aceita T e todos os seus descendentes

<? super T>

Aceita T e todos os seus ascendentes

```
public class ColecaoBichoFelino {  
    public void addAnimal(List<? extends Felino> animais) {  
        //animais.add(new Leao()); //não pode adicionar quando e utilizado  
        //<? extends Felino>  
        for (Felino bicho : animais) {  
            bicho.fazerRuido();  
        }  
    }  
    public static void main(String[] args) {  
        List<Leao> animais = new ArrayList<Leao>();  
        animais.add(new Leao());  
        ColecaoBichoFelino colecao = new ColecaoBichoFelino();  
        colecao.addAnimal(animais);  
    }  
}
```

Aceita somente Felino
Neste caso [Leão]

