

Programação Orientada a Objetos - POOS3

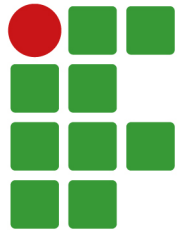
1

Tecnologia em Análise e Desenvolvimento de Sistemas

Aula 7

Utilizando o Generics

2º semestre de 2018



INSTITUTO FEDERAL

São Paulo

Câmpus Araraquara

Problema

- O exercício avaliativo 6 propõe a implementação de um `ArrayCoisa` que armazena um array de objetos que implementam a interface `ICoisa`.
- Como fazer com que nosso `ArrayCoisa` armazene qualquer tipo de objeto, independentemente da interface?

```

public class ArrayConjuntoGeneric<T> implements IConjunto<T>{
    private final int DEFAULT_SIZE = 20;
    private Object dados[];
    private int tamanhoMaximo;
    private int qtdElementos;

    public ArrayConjuntoGeneric() {
        this.tamanhoMaximo = DEFAULT_SIZE;
        init();
    }

    public ArrayConjuntoGeneric(int tamanho) {
        if(tamanho > 0) {
            this.tamanhoMaximo = tamanho;
        }else
            this.tamanhoMaximo = DEFAULT_SIZE;
        init();
    }

    private void init() {
        this.qtdElementos = 0;
        this.dados = new Object[this.tamanhoMaximo];
    }

    @Override
    public boolean add(T coisa) {
        boolean deuCerto = false;
        if(coisa != null)
            if(this.qtdElementos == this.tamanhoMaximo) {
                dobrarArray();
                dados[qtdElementos] = coisa;
                qtdElementos++;
                deuCerto = true;
            }
        return deuCerto;
    }
}

```

```

    private void dobrarArray() {
        Object novo[] = new Object[this.tamanhoMaximo*2];
        for(int i=0; i<this.tamanhoMaximo; i++)
            novo[i] = dados[i];
        this.tamanhoMaximo *= 2;
        this.dados = novo;
    }

    @Override
    public boolean contains(T coisa) {
        boolean existe = false;
        for(int i=0; i<qtdElementos && !existe; i++) {
            if(dados[i].equals(coisa))
                existe = true;
        }
        return existe;
    }

    @Override
    public T get(int posicao) {
        Object coisa = null;
        if(posicao >= 0 && posicao < qtdElementos)
            coisa = dados[posicao];
        return (T)coisa;
    }

    @Override
    public T remove(int posicao) {
        Object coisa = null;
        if(posicao >= 0 && posicao < qtdElementos) {
            coisa = dados[posicao];
            for(int i=posicao; i < qtdElementos-1; i++)
                dados[i] = dados[i+1];
            qtdElementos--;
        }
        return (T) coisa;
    }
}

```

```

public class Main {
    public static void main(String args[]) {
        ArrayConjuntoGeneric<Nada> arrayNadas;

        arrayNadas = new ArrayConjuntoGeneric<>();

        for(int i=0; i < 10; i++) {
            arrayNadas.add(new Nada());
        }

        while(!arrayNadas.isEmpty()) {
            System.out.println(arrayNadas.remove(0).toString());
        }
    }
}

```

```

public class Nada {

    @Override
    public String toString() {
        return "Sou um nada!";
    }
}

```

```

public class Main2 {

    public static void main(String[] args) {

        ArrayConjuntoGeneric<String> mensagens = new ArrayConjuntoGeneric<>(10);

        for(int i=0; i < 100; i++ ) {
            mensagens.add("Nada a declarar em " + i);
        }

        while(!mensagens.isEmpty()) {
            System.out.println(mensagens.remove(0));
        }
    }
}

```

Classe Genérica

- Analisando o exemplo da ArrayCoisaGeneric:
 - observa-se que o código da classe não mudaria se os elementos do array fossem Strings ou Pets ou qualquer outro tipo de objeto.
 - o que mudaria seria o tipo de entrada do add() e o tipo de retorno do get() e remove().
- Para melhorar o **reuso** de software, a partir do Java 5, foi disponibilizado o uso de Classe Genérica, que apoiada pelo polimorfismo (compatibilidade de tipos) permite maravilhas na implementação, reduzindo a quantidade de **cast** que deve ser realizada.



- **Leitura Obrigatória**

- Winder, R.; Roberts, G. Desenvolvendo Software em Java. 3 ed. Rio de Janeiro: LTC, 2009.
 - Capítulos 6.3
- <https://www.devmedia.com.br/usando-generics-em-java/28981>
- <http://argonavis.com.br/cursos/java/j100/java-5-generics.pdf>
- <https://javaprogramando.blogspot.com.br/2009/02/artigo-classes-gerenericas.html>

Trabalhando

- **Exercício avaliativo**
 - 9

