

Programação Orientada a Objetos

Prof. Dr. Anderson V. de Araujo



Módulo III - Pilares da programação orientada a objetos

Unidade IV - Classes e Métodos Genéricos



Pilha de Inteiros

```
class PilhaVInt{  
    int[] v = new int[100];  
    int top = 0;  
    void push(int x) {  
        v[top++] = x;  
    }  
    int pop() {  
        return v[--top];  
    }  
}
```

```
//outros códigos...  
public static void main(String[] s) {  
    PilhaVInt p = new PilhaVInt();  
    p.push(10);  
}
```

Pilha de Strings

```
class PilhaVStr{  
  
    String[] v = new String[100];  
  
    int top = 0;  
  
    void push(String x) {  
        v[top++]=x;  
    }  
  
    String pop() {  
        return v[--top];  
    }  
}
```

//outros códigos...

```
public static void main(String[] s){  
    PilhaVStr p = new PilhaVStr();  
    p.push("asdf");  
}  
}
```

Criando suas Classes usando Generics

- É possível criar a sua própria classe e adicionar elementos genéricos, que só vão ter seu tipo definido em *runtime*;
- Só definir os tipos genéricos entre <> separados por vírgula após a definição da classe;
- Ainda é possível definir se o tipo é filho de alguma classe específica usando a palavra `extends` para cada tipo definido ou `implements` para uma interface.

Criando suas Classes usando Generics – Exemplo 1

```
class PilhaVetor<T>{  
    T[] v = (T[])new Object[100];  
    int top = 0;  
  
    void push(T x) {  
        v[top++]=x;  
    }  
  
    T pop() {  
        return v[--top];  
    }  
}
```

```
//outros códigos...  
  
public static void main(String[] args) {  
    PilhaVetor<String> p = new  
PilhaVetor<String>();  
  
    p.push("Texto");  
  
    PilhaVetor<Integer> p = new  
PilhaVetor<Integer>();  
  
    p.push(10);  
  
}  
}
```

Criando suas Classes usando Generics

– Exemplo 2

```
class MyClass<T extends Number, C>{  
    T a;  
    C b;  
  
    public MyClass(T a, C b) {  
        this.a = a;  
        this.b = b;  
    }  
}
```

```
void print() {  
    System.out.println(a);  
    System.out.println(b);  
}  
  
public static void main(String[] s) {  
    MyClass<Double, String> m = new  
    MyClass<>(.1, "asdf");  
    m.print();  
}
```

Notação *diamond*

Licenciamento



Respeitadas as formas de citação formal de autores de acordo com as normas da ABNT NBR 6023 (2018), a não ser que esteja indicado de outra forma, todo material desta apresentação está licenciado sob uma [Licença Creative Commons - Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).