

# Linguagem de Programação I



Prof. Edson Kaneshima

# Atributos e Métodos Estáticos

A series of horizontal lines in teal and light blue colors, with varying lengths and thicknesses, extending across the width of the slide.

Aula 01

# Atributos Estáticos

- Se definirmos um atributo como static, então só haverá um atributo desses por classe. Em contraste, cada objeto tem sua própria cópia de todos os atributos de instâncias.
- Ex.: suponha que atribuímos um número de identificação singular para cada empregado. Adicionamos um atributo id e um atributo estático proximalId à classe Empregado:

```
public class Empregado {  
    private int id;  
    private static int proximalId = 1;  
}
```

# Atributos Estáticos

- Cada objeto Empregado agora tem seu próprio atributo id, mas há somente um atributo proximalId, que é compartilhado entre todas as instâncias da classe.
- Ex.: se houver 1000 objetos da classe Empregado, então há 1000 atributos id, um para cada objeto. Porém, há somente um campo estático proximalId.
- Mesmo se não houver qualquer objeto Empregado, o atributo estático proximalId está presente. **Assim, ele pertence à classe, não a qualquer objeto individual.**

# Atributos Estáticos

- Implementando um método simples:

```
public void setId( ) {  
    id = proximaId;  
    proximaId++;  
}
```

- Suponha que você defina o número de identificação para rodrigo:

```
rodrigo.setId( );
```

- Então, o atributo id de rodrigo está definido, e o valor do atributo estático proximaId é incrementado.

# Constantes Estáticas

- Variáveis estáticas são bastante raras. No entanto, constantes estáticas são mais comuns. Ex.:

```
public class Util {  
    ...  
    public static final double PI = 3.14159265358979323846;  
    ...  
}
```

- A constante pode ser acessada como Util.PI.
- Se a palavra-chave static tivesse sido omitida, então PI seria um atributo da classe Util. Ou seja, seria necessário um objeto da classe Util para acessar PI, e todo objeto Util teria sua própria cópia de PI.

# Métodos Estáticos

- Métodos que não operam sobre objetos. Ex.: método `pow` da classe `Math` é um método estático.

`Math.pow(x, a)`

- Calcula a potência  $x^a$ . Não usa qualquer objeto `Math` para executar sua tarefa.
- Pelo fato de os métodos estáticos não operarem em objetos, você não pode acessar atributos a partir de um método estático. Porém, métodos estáticos podem acessar atributos estáticos de suas classes.

# Métodos Estáticos

```
public static int getProximaId() {  
    return proximaId; // retorna um campo estático  
}
```

- Para chamar esse método, devemos fornecer o nome da classe:

```
int n = Empregado.getProximaId();
```



# Métodos Estáticos

- Utiliza-se métodos estáticos em duas situações:
  - Quando um método não precisa acessar o estado do objeto porque todos os parâmetros necessários foram fornecidos como parâmetros explícitos (exemplo: *Math.pow*)
  - Quando um método somente precisa acessar atributos estáticos da classe (exemplo: *Empregado.getProximaId*)