

# TI-220 Java Orientado a Objetos

---

ANTONIO CARVALHO - TREINAMENTOS

A solid blue horizontal bar spanning the width of the slide, located at the bottom.

# JavaFX Propriedades

---

# JavaFX Propriedades

---

Até o momento o uso de propriedades em Java segue o padrão JavaBean, onde o atributo é mantido como privado, e são criadas funções publicas **get()** e **set()**.

Porém o uso das propriedades convencionais não permite a **ligação** entre elas, tão pouco são **observáveis**, ou seja permitem a notificação quando ocorrem mudanças.

No exemplo seguinte quando ocorre uma mudança de valor na variável, nenhum código fica conhecendo o esta mudança

# JavaFX Propriedades

---

No exemplo abaixo quando ocorre uma mudança de valor na variável, nenhum código pode observar esta mudança.

```
public class Aluno {  
    private String nome;  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public static void main(String[] args) {  
        Aluno a = new Aluno();  
        a.setNome("João");  
        a.setNome("José");  
    }  
}
```

# JavaFX Propriedades

---

Para resolver esta necessidade a biblioteca **JavaFX** conta com um vasto suporte a propriedades observáveis pelo pacote **javafx.beans.property**

É possível criar propriedades com base nas classes abstratas abaixo:

<b>BooleanProperty</b>	<b>IntegerProperty</b>	<b>FloatProperty</b>	<b>SetProperty</b>
<b>ObjectProperty</b>	<b>LongProperty</b>	<b>DoubleProperty</b>	<b>ListProperty</b>
<b>StringProperty</b>			<b>MapProperty</b>

# JavaFX Propriedades

---

A criação da propriedade é feita instanciando um objeto de acordo com a classe abstrata da propriedade desejada.

## □ Criar uma propriedade do tipo **String**

### ■ Sintaxe:

```
StringProperty <propriedade> = new <Classe extends  
StringProperty>();
```

### ■ Exemplo:

```
StringProperty nome = new SimpleStringProperty("João");
```

ou

```
StringProperty nome = new SimpleStringProperty();  
nome.set("João");
```

# JavaFX Propriedades

---

O uso de propriedades do **JavaFX** permite que as mudanças de valores sejam observadas, por objetos do tipo **ChangeListener**

- Observar uma propriedade do tipo **String**

- Sintaxe:

```
class <NomeClasse> implements ChangeListener<T> {  
    public void changed(  
        ObservableValue<? extends T> observable, T oldValue, T newValue) {  
    }  
}
```

- Exemplo:

```
class Observador implements ChangeListener<String> {  
    public void changed(ObservableValue<? extends String> observable,  
        String oldValue, String newValue) {  
        System.out.printf("Mudou o valor de %s para %s\n", oldValue, newValue);  
    }  
}
```

# JavaFX Propriedades

---

Com a classe do tipo **ChangeListener** criada, é possível usá-la para observar as mudanças de valores nas propriedades do tipo **javafx.beans.property**

- Criar uma propriedade do tipo **String**, e registrar o observador

- Sintaxe:

```
ChangeListener<T> <objeto> = new <Classe implemets  
ChangeListener<T>>();
```

- Exemplo:

```
ChangeListener<String> observador = new Observador();  
StringProperty nome = new SimpleStringProperty("João");  
nome.addListener(observador);
```



# Bibliografia

---

**DEA, C., GRUNWALD, G., PEREDA, J., PHILLIPS, S., HECKLER, M.,** JavaFX 9 by Example 3rd Edition, Apress, 2017

**EBBERS, H.,** Mastering JavaFX 8 Controls, McGrawHill Education, 2014

**HOMMEL, S.,** JavaFX Properties and Binding Tutorial Release 8, Oracle Press, 2014

**MORRIS, S.,** JavaFX in Action, MANNING Greenwich, 2010

**ORACLE,** JavaFX 2.0 Programming Model, Oracle Press, 2011

**VOS, J., CHIN, S., WEIQI, G., WEAVER, J., IVERSON, D.,** Pro JavaFX 9 - A Definitive Guide to Building Desktop, Mobile, and Embedded Java Clients 4th edition, Apress, 2018