

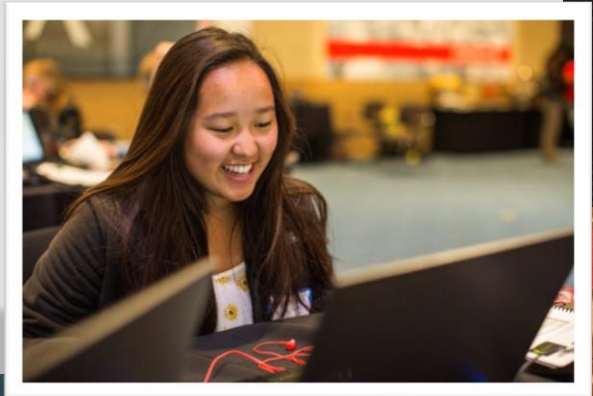


# Java Foundations

9-2

Cores e Formas

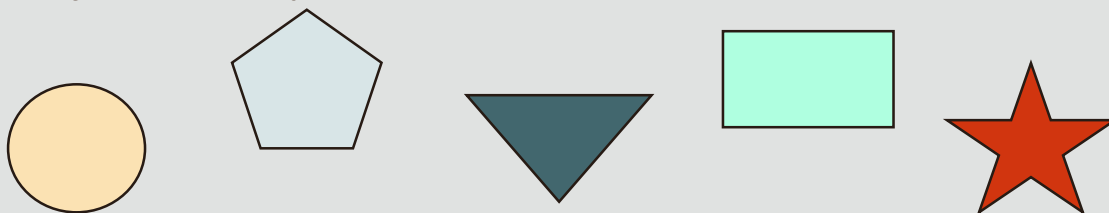
**ORACLE**  
Academy



Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

# Adivinhem, Garotada?!

- Hoje vamos aprender sobre as cores e formas!



**Sim!!!**



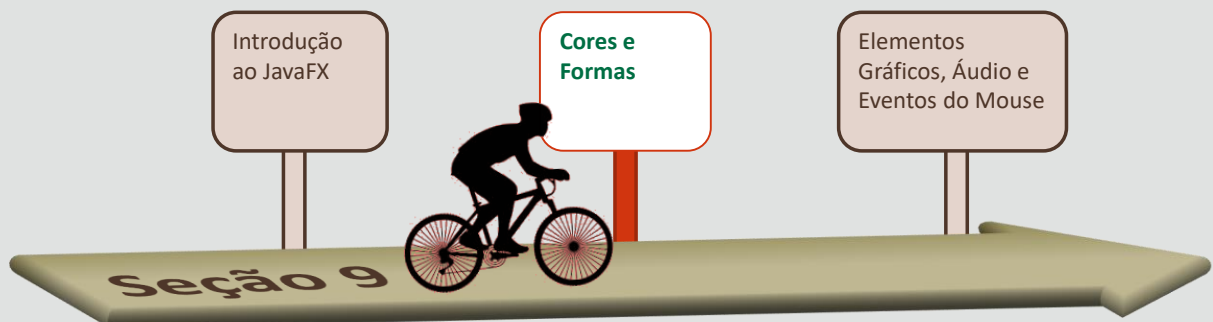
# Objetivos

- Esta lição abrange os seguintes objetivos:
  - Criar e usar cores personalizadas
  - Criar formas e explicar as respectivas propriedades e comportamentos
  - Fazer referência ao JavaFX Ensemble



# Tópicos

- **Cores**
- Formas
- O JavaFX Ensemble



**ORACLE**  
Academy

JFo 9-2  
Cores e Formas!

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

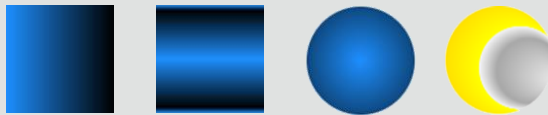
5

# O que Posso Fazer com Cores no JavaFX?

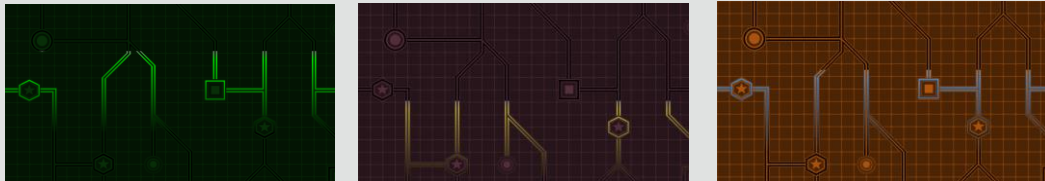
- Formas de cores



- Criar gradientes



- Colorir imagens



# O JavaFX Contém uma Classe Color

- As cores podem ser armazenadas como variáveis:

```
Color color = Color.BLUE;
```

- As cores podem ser passadas em métodos:

```
Scene scene = new Scene(root, 300, 250, Color.BLACK);
```

- Este exemplo faz com que o plano de fundo da cena seja preto

- Mas antes de usar qualquer Cor...

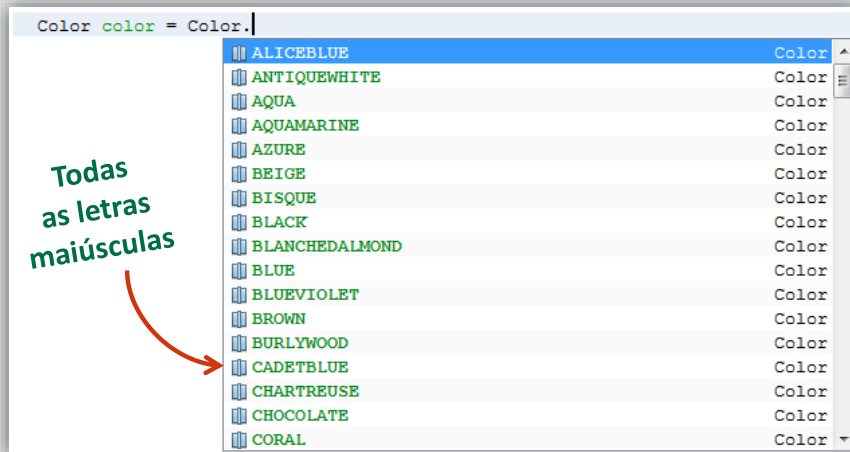
- Primeiro você precisará fazer a seguinte importação:

```
import javafx.scene.paint.Color;
```

- Ignore as outras sugestões de importação de cores do NetBeans

## Fazendo Referência a uma Cor

- Existem muitas cores no JavaFX
- Digitar Color. no NetBeans exibe a lista inteira de cores possíveis





## Personalizando uma Cor

- Se você não estiver satisfeito com as cores que o JavaFX fornece, existem maneiras de personalizar sua própria cor
- A classe `Color` contém métodos para fazer isso:

```
Color customColor = Color.rgb(  
    rgb(int i, int i1, int i2) Color  
    rgb(int i, int i1, int i2, double d) Color
```

vermelho verde azul opacidade

- Personalize uma cor misturando os componentes vermelho, verde e azul
- A opacidade também pode ser controlada

O JavaFX fornece outros métodos para criar cores. Você está convidado a usá-los em seus programas, mas analisaremos apenas o método RGB (red-green-blue, vermelho, verde, azul) nesta lição.

# O Intervalo de Componentes de Cores

```
Color customColor = Color.rgb|
    rgb(int i, int i1, int i2) Color
    rgb(int i, int i1, int i2, double d) Color
```

vermelho verde azul opacidade

Componente	Intervalo de valores
Vermelho	0–255
Verde	0–255
Azul	0–255
Opacidade	0.0–1.0

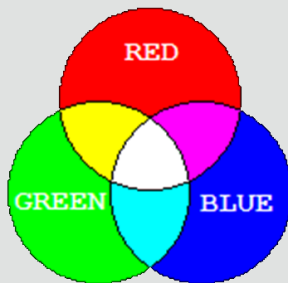
## Exemplo de Cor

- Neste exemplo, a cor resultante contém...

```
Color color = new Color.rgb(255, 255, 20);
```

- A quantidade máxima de Vermelho possível
  - A quantidade máxima de Verde possível
  - Somente um pouco de Azul
- 
- A cor resultante é muito próxima do amarelo
    - Mas como sabemos isso?
    - Na maioria dos casos, encontrar a cor perfeita é uma questão de "tentativa e erro", mas existem princípios de orientação

# Regras de Combinação Aditiva de Cores



Exemplos:

Código	Cor
<code>Color.rgb(255, 0, 0 );</code>	vermelho
<code>Color.rgb(0, 255, 0 );</code>	verde
<code>Color.rgb(0, 0, 255);</code>	azul
<code>Color.rgb(255, 255, 0 );</code>	amarelo
<code>Color.rgb(0, 0, 0);</code>	preto
<code>Color.rgb(255, 255, 255);</code>	branco

Vermelho  
puro  
Verde puro  
Azul puro  
Sem azul  
Sem cor  
Todas as  
cores

**ORACLE**  
Academy

JFo 9-2  
Cores e Formas!

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

12

As cores são criadas de acordo com as regras de combinação aditiva de cores.

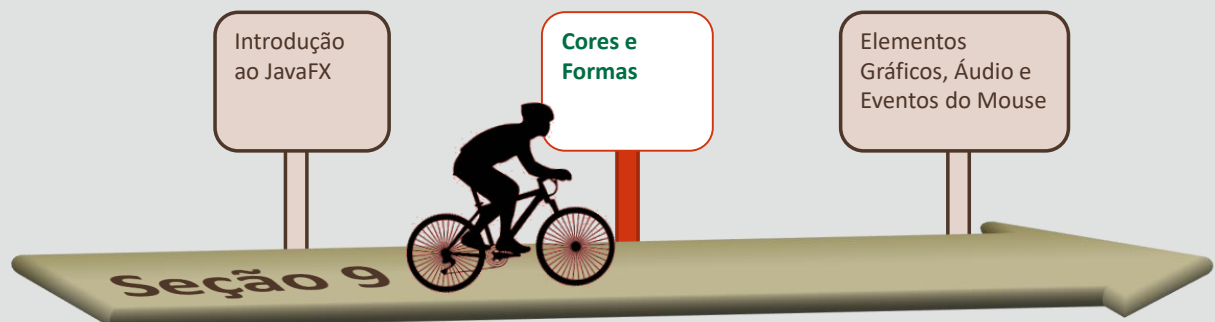


## Exercício 1

- Crie um novo projeto JavaFX
  - Altere o Nó Raiz para um tipo Group
  - Remova o botão e qualquer outro código necessário relacionado a ele
- Teste cores personalizadas
  - Crie algumas cores personalizadas
  - Veja suas cores personalizadas no plano de fundo da cena, fornecendo um argumento Color quando a Scene for instanciada

# Tópicos

- Cores
- **Formas**
- O JavaFX Ensemble



**ORACLE**  
Academy

JFo 9-2  
Cores e Formas!

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

14

# Isto é um Retângulo

- É assim que se instancia um Retângulo JavaFX:



```
Rectangle rect = new Rectangle(20, 20, 100, 200);
```

Posição de x  
Posição de y  
largura  
altura

- Primeiro você precisará fazer esta importação:

```
import javafx.scene.shape.Rectangle;
```

- Ignore as outras sugestões de importação de Retângulo do NetBeans

## Métodos Importantes para Retângulos

- Podemos ter uma ideia das propriedades de um Retângulo no construtor e nos métodos a seguir:
  - setX(double d)
  - setY(double d)
  - setWidth(double d)
  - setHeight(double d)
  - setFill(Paint paint)
  - setStroke(Paint paint)
  - setStrokeWidth(double d)
- Mas o que exatamente esses métodos fazem?

Eles podem aceitar uma cor como argumento





## Exercício 2

- Continue editando o projeto JavaFX que você criou no exercício anterior
- Crie um Retângulo e adicione-o ao Nó Raiz
- Chame cada método descrito no slide anterior
- Você consegue imaginar o que cada método faz?

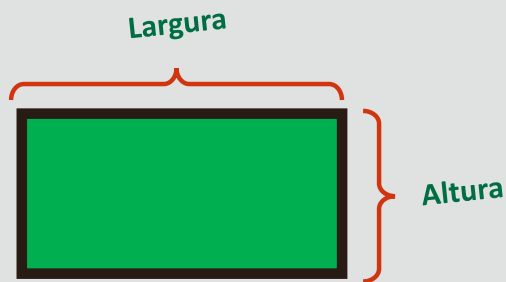
# Descrições de Métodos, Parte 1

- `setFill(Paint paint)`
  - Define a cor do Retângulo
- `setStroke(Paint paint)`
  - Define a cor do contorno do Retângulo
- `setStrokeWidth(double d)`
  - Define a largura do contorno do Retângulo



## Descrições de Métodos, Parte 2

- `setX(double d)`
- `setY(double d)`
  - Define a posição de x ou y do Retângulo
- `setWidth(double d)`
- `setHeight(double d)`
  - Define a largura ou a altura do Retângulo



## Alterando a Posição de um Nó

- Vimos várias maneiras de mudar a posição de um nó...mas qual é a maneira preferida?
- `setX(double d)`
- `setY(double d)`
  - Essas são as preferidas na maioria dos casos
- `setLayoutX(double d)`
- `setLayoutY(double d)`
  - Utilize-as se seu Nó estiver bloqueado em um painel de Layout, como `FlowPane`
  - Ou se `setX()` não estiver disponível, o que é o caso dos elementos de UI, como os Botões

*setX() definitivamente não funciona neste caso*

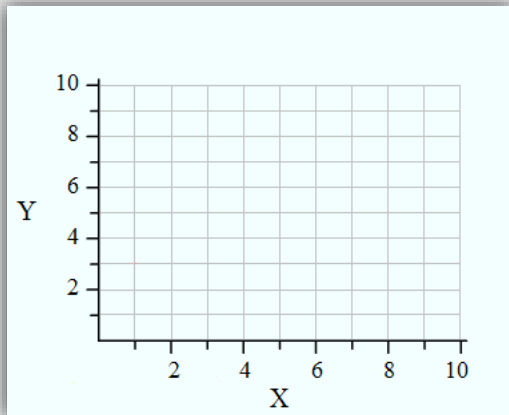
## Posicionando um Nó

- A maioria dos Nós são posicionados em relação ao respectivo canto superior esquerdo
  - E não em relação ao seu centro geográfico
- Se você chamar `setX(100)` em um Nó...
  - A posição de x do canto superior esquerdo do Nó será definida como 100



Um Círculo do JavaFX é uma exceção a essa regra. Os Círculos do JavaFX são posicionados em relação ao respectivo centro.

# Sistemas de Coordenadas

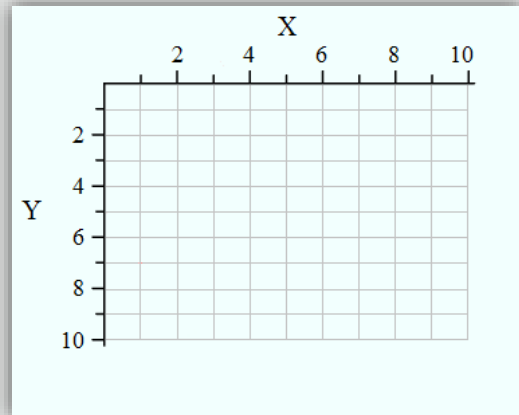


## Sistema de Coordenadas Matemáticas

- A origem está localizada no canto inferior esquerdo

**ORACLE**  
Academy

JFo 9-2  
Cores e Formas!



## Sistema de Coordenadas do JavaFX

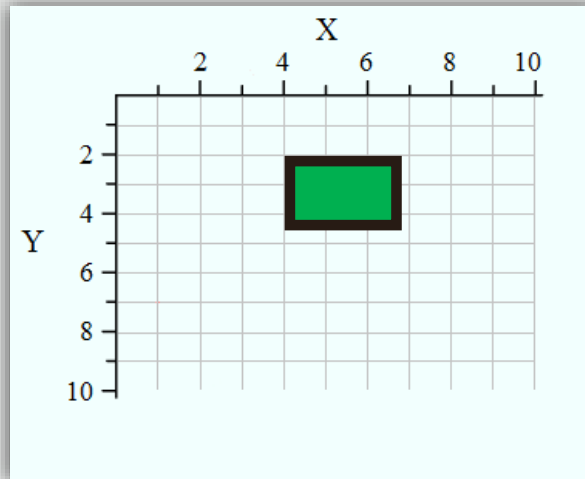
- A origem está localizada no canto superior esquerdo
- O eixo y está para trás

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

22

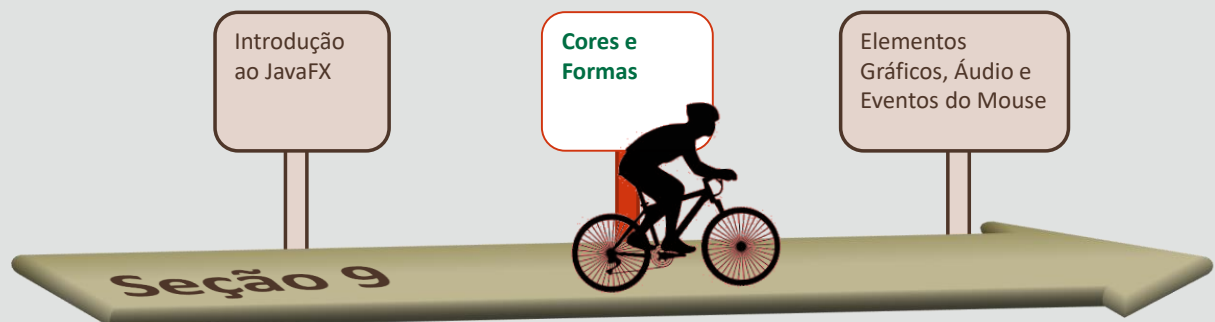
## Exemplo de Posicionamento

- Este Retângulo é posicionado em (4,2) chamando:
  - `setX(4) ;`
  - `setY(2) ;`



# Tópicos

- Cores
- Formas
- **O JavaFX Ensemble**



**ORACLE**  
Academy

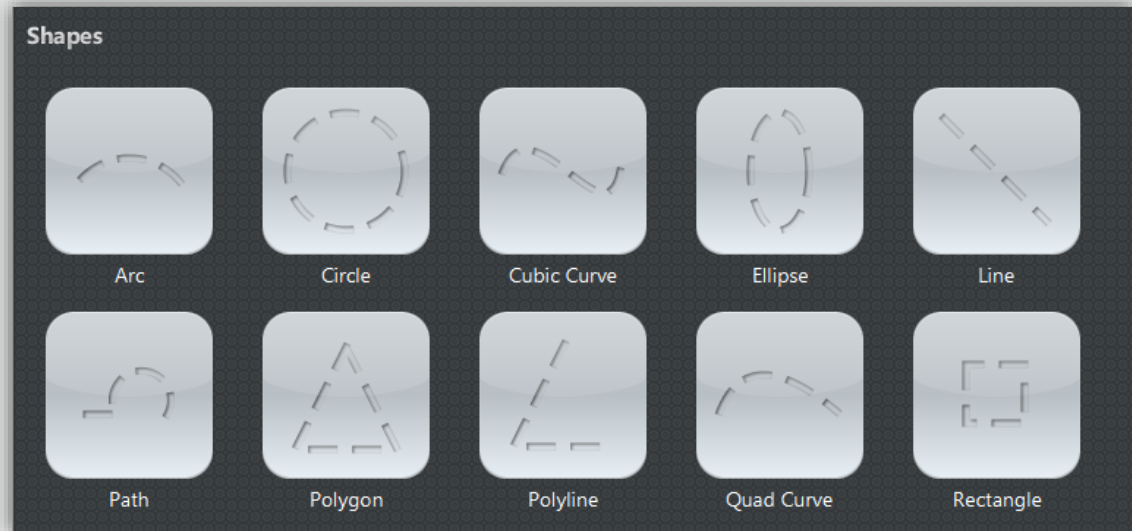
JFo 9-2  
Cores e Formas!

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

24



# Muitas Formas Estão Disponíveis no JavaFX



## O JavaFX Ensemble

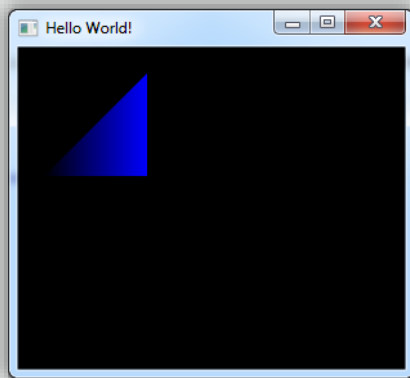
- Ele contém exemplos de código de recursos do JavaFX
- Em geral, consultamos o Ensemble ao desenvolver um Java Puzzle Ball
- Ele é uma ferramenta de grande utilidade para explorar e solucionar problemas do JavaFX



Estávamos aprendendo a usar o JavaFX ao mesmo tempo em que estávamos desenvolvendo o jogo.

## Exercício 3

- Fazer referência ao JavaFX Ensemble
- Você saberia criar um triângulo retângulo com uma cor de gradiente?



**JavaFX** Ensemble

**ORACLE**  
Academy

JFo 9-2  
Cores e Formas!

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

27

Você pode ignorar as reclamações do Ensemble sobre acesso à Internet. Fornecemos o arquivo `.jar` do Ensemble no Oracle Academy Member Hub, e uma versão do programa também pode estar disponível na Oracle em <http://download.oracle.com/otndocs/products/javafx/2/samples/Ensemble/index.html>.

# Explorando o Ensemble: Exemplo de Gradiente Linear



- O exemplo de Gradiente Linear mostra...

- Como criar um gradiente:

```
//criar um gradiente linear simples
LinearGradient gradient1 = new LinearGradient(0, 0, 1, 0, true,
CycleMethod.NO_CYCLE, new Stop[] {
    new Stop(0, Color.DODGERBLUE),
    new Stop(1, Color.BLACK)
});
```

- Como colorir uma forma com um gradiente:

```
//Primeiro retângulo
Rectangle rect1 = new Rectangle(0,0,80,80);

//definir preenchimento do retângulo
rect1.setFill(gradient1);
```

- Lembre-se de fazer as devidas importações

Verifique as guias Source Code para descobrir o código mencionado neste slide. Também pode ser muito útil pesquisar a documentação do Java para ver uma descrição dos argumentos do construtor.

# Explorando o Ensemble: Exemplo do Polígono



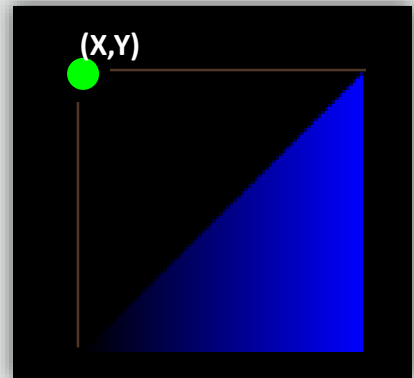
- O exemplo do Polígono mostra...
  - Como criar um polígono com base em uma matriz de pontos:

```
//Triângulo simples
Polygon polygon1 = new Polygon(new double[]{
    45.0, 10.0,
    10.0, 80.0,
    80.0, 80.0,
});
```

- Combine esse com o exemplo do gradiente e você terá sua solução
  - Você perceberá como o Ensemble é um recurso valioso
  - Isso pode ser muito útil quando você faz o conjunto de problemas

# O Polígono

- O Polígono tem métodos semelhantes aos de um Retângulo
  - Os nós compartilham os mesmos métodos
- Se você testar com `setLayoutX()`...
  - Perceberá que o Polígono está posicionado em relação ao local em que seu canto superior esquerdo deveria estar



## Sobre o Java Puzzle Ball

- Desenhamos linhas e polígonos para detecção de colisão
  - Mas essas linhas estão ocultas na versão mais atual



- Também desenhamos octógonos ao redor de cada bumper
  - Um octógono interno trata da detecção da colisão
  - Um octógono externo detecta se a bola foi longe o bastante para fazer com que o bumper gire
- Foi preciso fazer um trabalho extra para posicionar e girar os Nós da forma como queríamos

Estávamos aprendendo a usar o JavaFX ao mesmo tempo em que estávamos desenvolvendo o jogo.

# Resumo

- Nesta lição, você deverá ter aprendido a:
  - Criar e usar cores personalizadas
  - Criar formas e explicar as respectivas propriedades e comportamentos
  - Fazer referência ao JavaFX Ensemble





The Oracle Academy logo is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

# ORACLE

## Academy