

The logo for Oracle Academy 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

Java Foundations

9-2 Cores e Formas

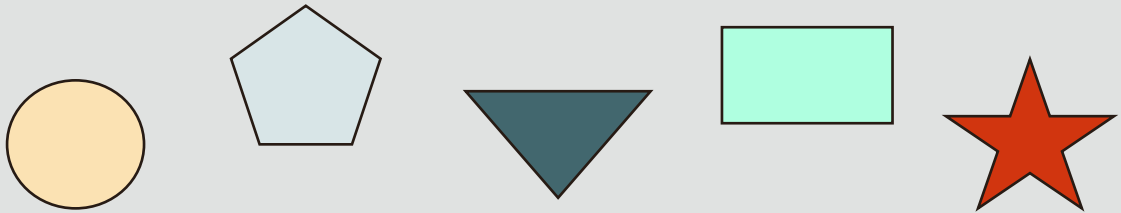
ORACLE
Academy



Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

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 à Documentação da API do JavaFX

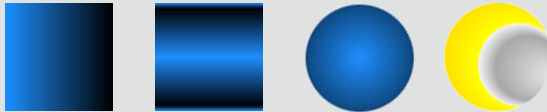


O que Posso Fazer com Cores no JavaFX?

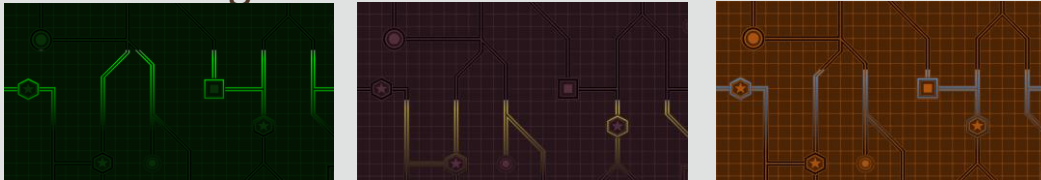
- Formas de cores



- 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...

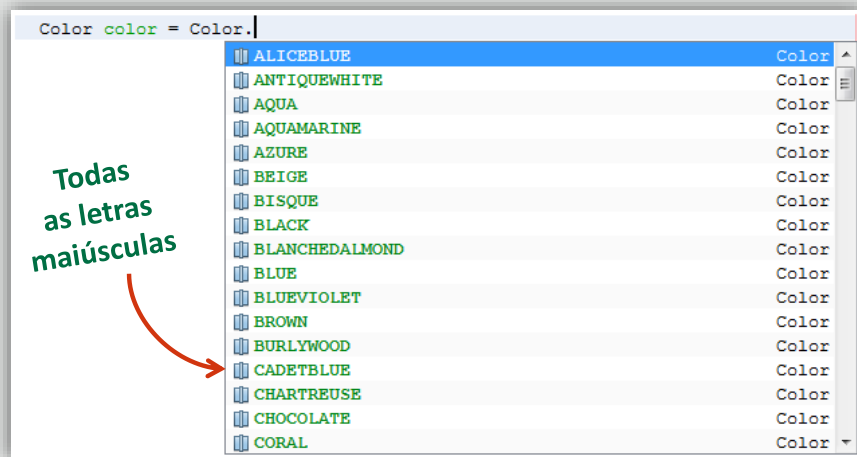
- Primeiro você precisará fazer a seguinte importação:

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

- Ignore as outras sugestões de importação de cores de seu IDE

Fazendo Referência a uma Cor

- Existem muitas cores no JavaFX
- Digitar `Color.` em seu IDE revela 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

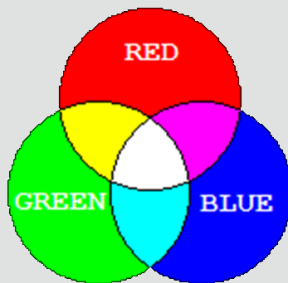
Exemplo de Cor

- Neste exemplo, a cor resultante contém...

```
Color color = 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 © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

11

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

Exercício 1

- Crie um novo projeto JavaFX usando o `JavaFXMainEx1.java`
 - `JavaFXMainEX1.java` é uma cópia de `JavaFXMain.java`
 - 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

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 de seu IDE

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)
- Eles podem aceitar uma cor como argumento
- (Existem muitos outros métodos de Retângulos além desses sete)
 - Mas o que exatamente esses métodos fazem?

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?

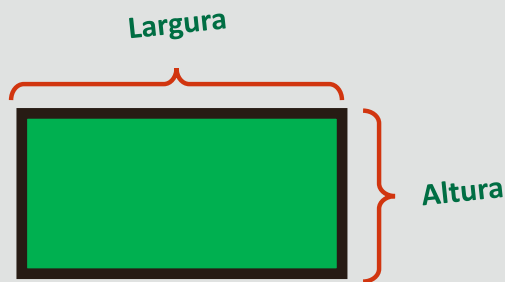
Descrições de Métodos, Parte 1

- `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

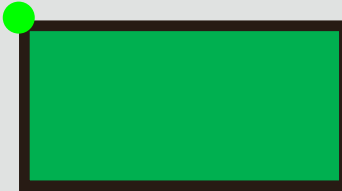
- Vimos várias maneiras de mudar a posição de um nó...mas qual é a maneira preferida?
- `setX(double d)`
- `setY(double d)`
 - Essas são as preferidas na maioria dos casos
- `setLayoutX(double d)`
- `setLayoutY(double d)`
 - Utilize-as se seu Nó estiver bloqueado em um painel de Layout, como `FlowPane`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

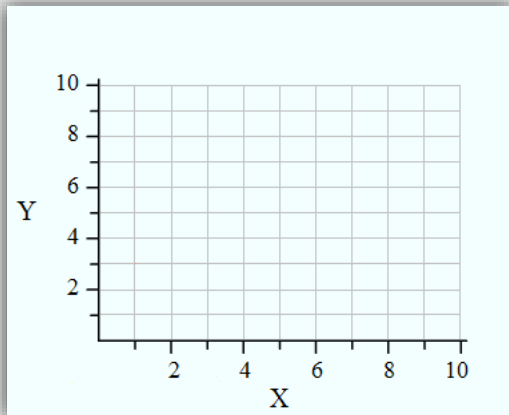
- A maioria dos Nós são posicionados em relação ao respectivo canto superior esquerdo
 - E não em relação ao seu centro geográfico
- Se você chamar `setX(100)` em um Nó...
 - A posição de x do canto superior esquerdo do Nó será definida como 100

(100, 0)



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

Sistemas de Coordenadas

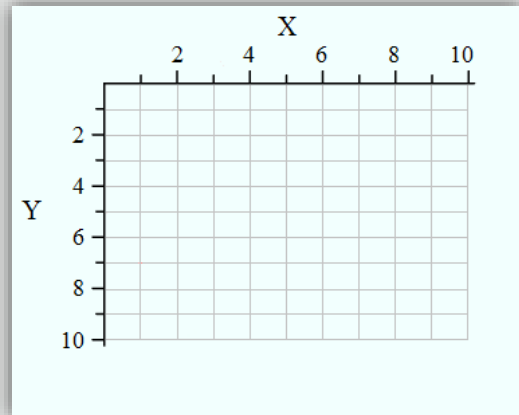


Sistema de Coordenadas Matemáticas

- A origem está localizada no canto inferior esquerdo

ORACLE
Academy

JFo 9-2
Cores e Formas!



Sistema de Coordenadas do JavaFX

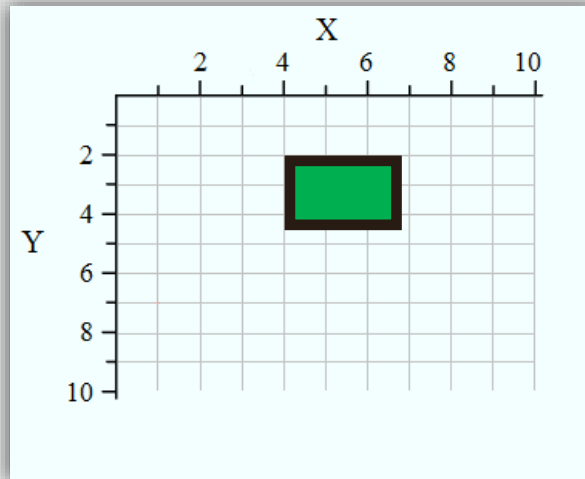
- A origem está localizada no canto superior esquerdo
- O eixo y está para trás

Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

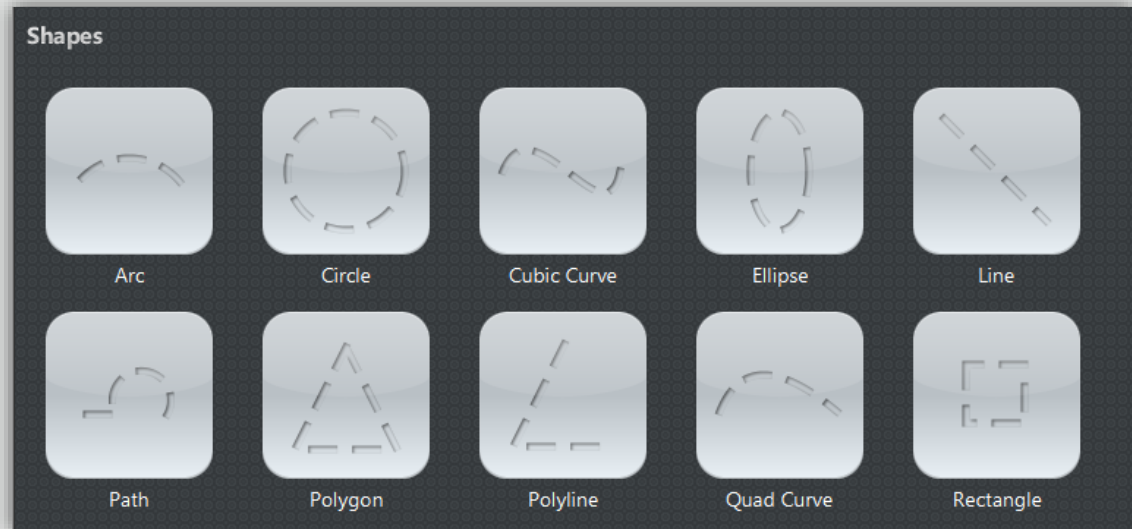
20

Exemplo de Posicionamento

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



Muitas Formas Estão Disponíveis no JavaFX



A Documentação da API do JavaFX

- Ela contém exemplos de código e informações de classe de recursos do JavaFX
- Vá para <https://openjfx.io/javadoc/17/index.html>
- O módulo Gráficos é um ótimo ponto de partida
- Você pode usar um recurso de pesquisa para localizar aulas específicas ou pode navegar pelos pacotes para obter ideias



ORACLE
Academy

JFo 9-2
Cores e Formas!

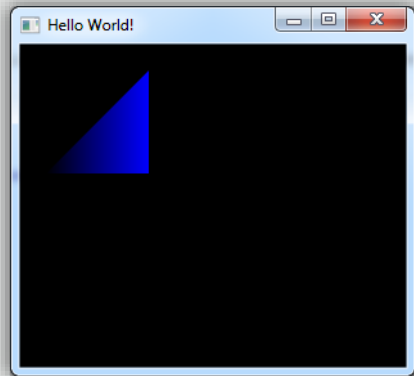
Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

23

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

Exercício 3

- Explore a documentação da API do JavaFX
- Você consegue descobrir como criar um triângulo retângulo com uma coloração gradiente usando o projeto JavaFX criado no exercício anterior?
- Dica: use a caixa de pesquisa para pesquisar primeiro por gradiente e, em seguida, por Polígono



Como explorar a documentação da API:

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

Como explorar a documentação da API:

Exemplo de polígono

- Digite polygon na caixa de texto da API; o exemplo Polygon nos mostra...
 - Como criar um polígono com base em uma matriz de pontos:

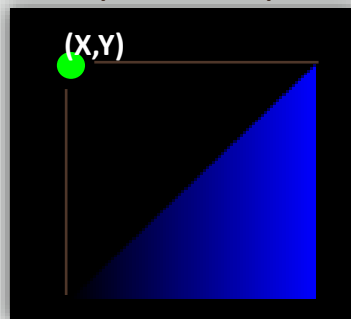


```
//Triângulo simples
Polygon polygon1 = new Polygon(new double[] {
    80.0, 10.0,
    80.0, 80.0,
    10.0, 80.0
});
```

- Combine esse com o exemplo do gradiente e você terá sua solução
 - Mas, ainda melhor, você perceberá como a documentação da API 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 à Documentação da API do JavaFX



