

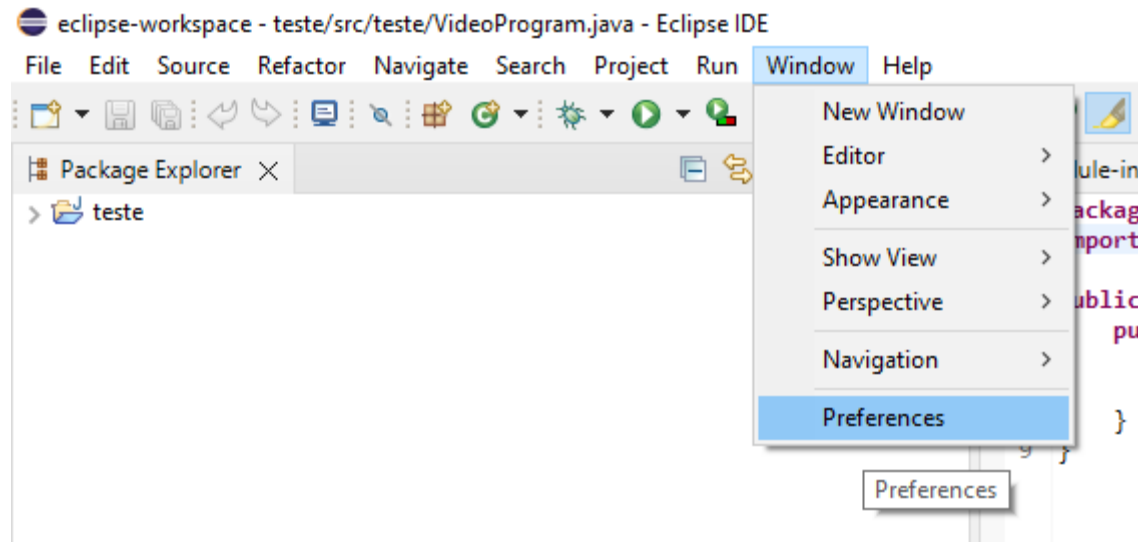
# Programação Orientada a Objetos

Aula 2 – Professor Diogo Orlando Nunes de Almeida

# Baixando o eclipse

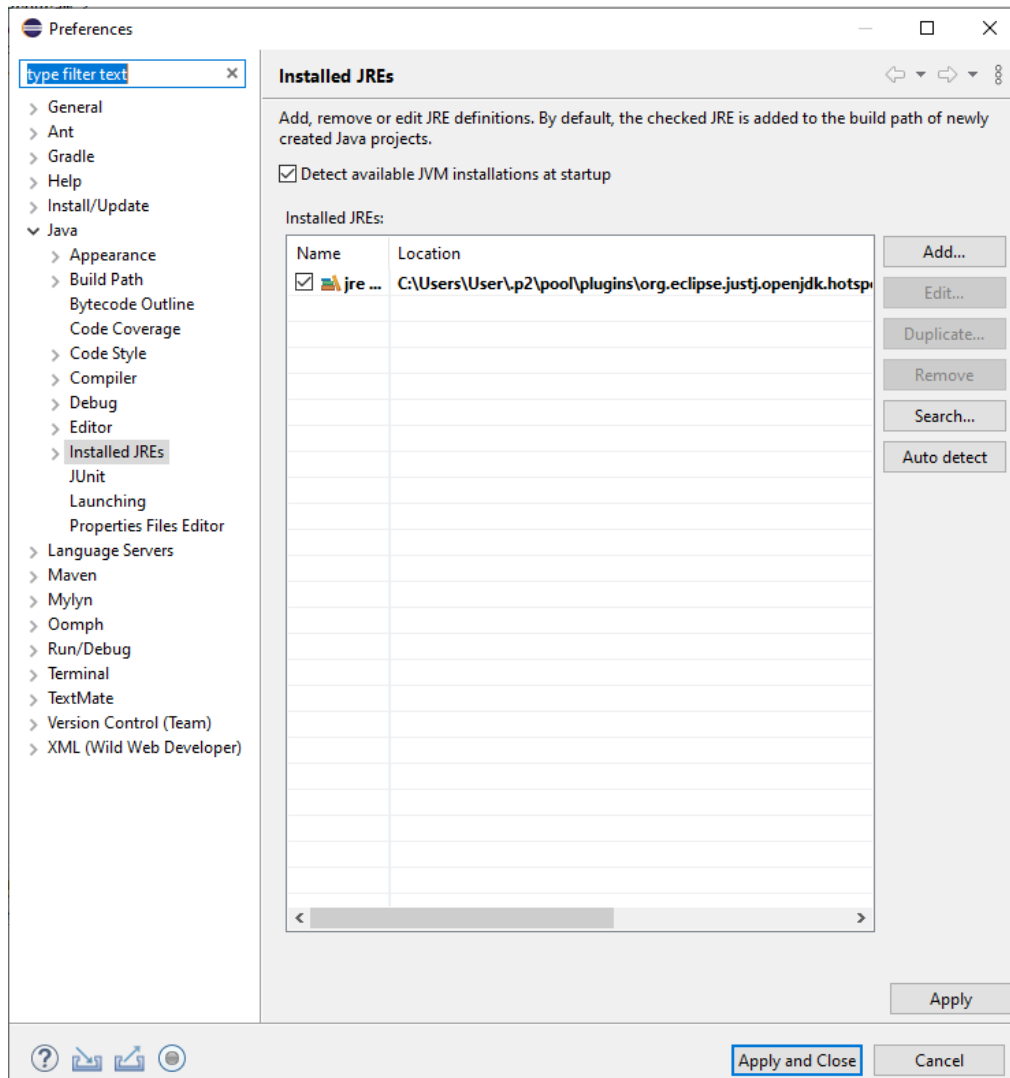
- Para baixar o eclipse: <https://eclipseide.org/>. Escolha IDE para Java Developers.
- Para baixar a SDK: <https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>
- Usaremos a versão do JDK 8 por ser melhor e ter as bibliotecas de uso geral.
- Existe versões mais atualizadas, e essas atualizações tem seu lado positivo como mais segurança e eficiência e negativo como alteração de biblioteca e alteração de linguagem.
- Ela é difícil de configurar.

# Depois de instalar a JDK e o eclipse



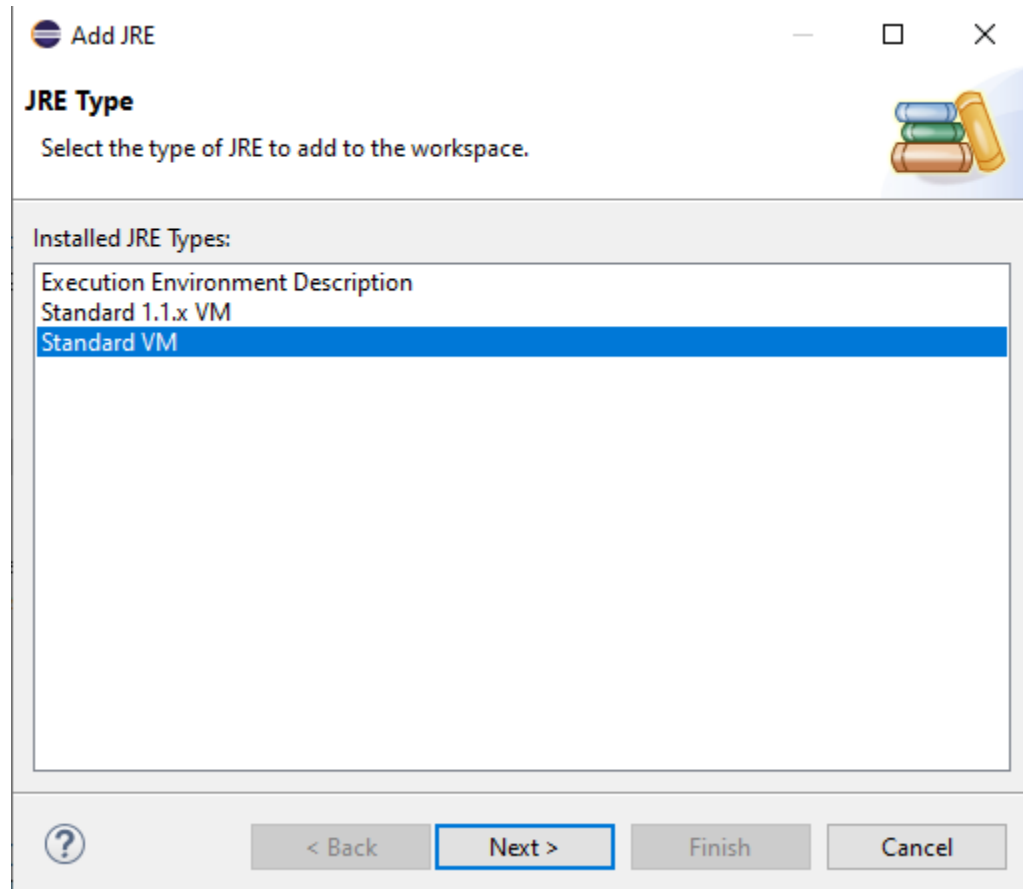
- Vá em Window -> Preferences.

# Depois de instalar a JDK e o eclipse



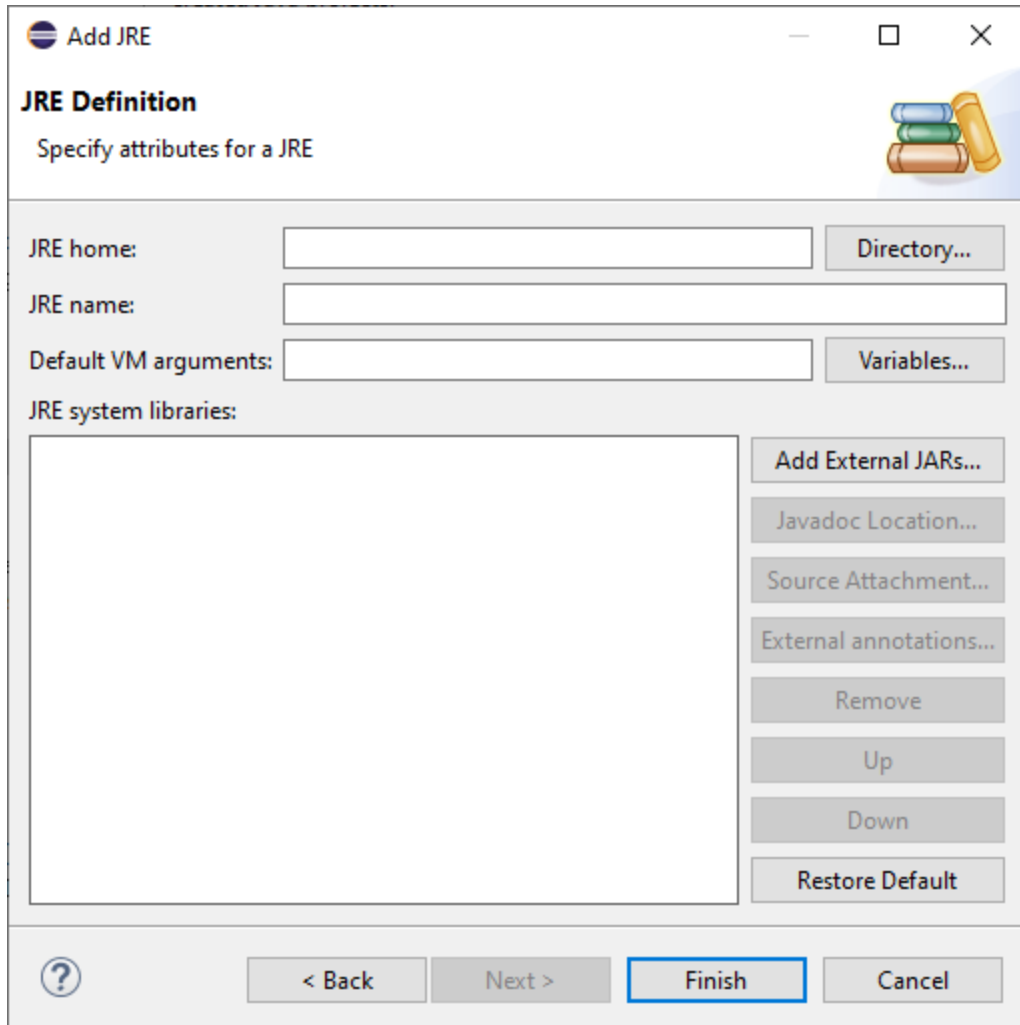
- Escolha JAVA e Installed JREs no menu a esquerda.
- Na tela que abrir clique em “Add...” na direita.

# Depois de instalar a JDK e o eclipse



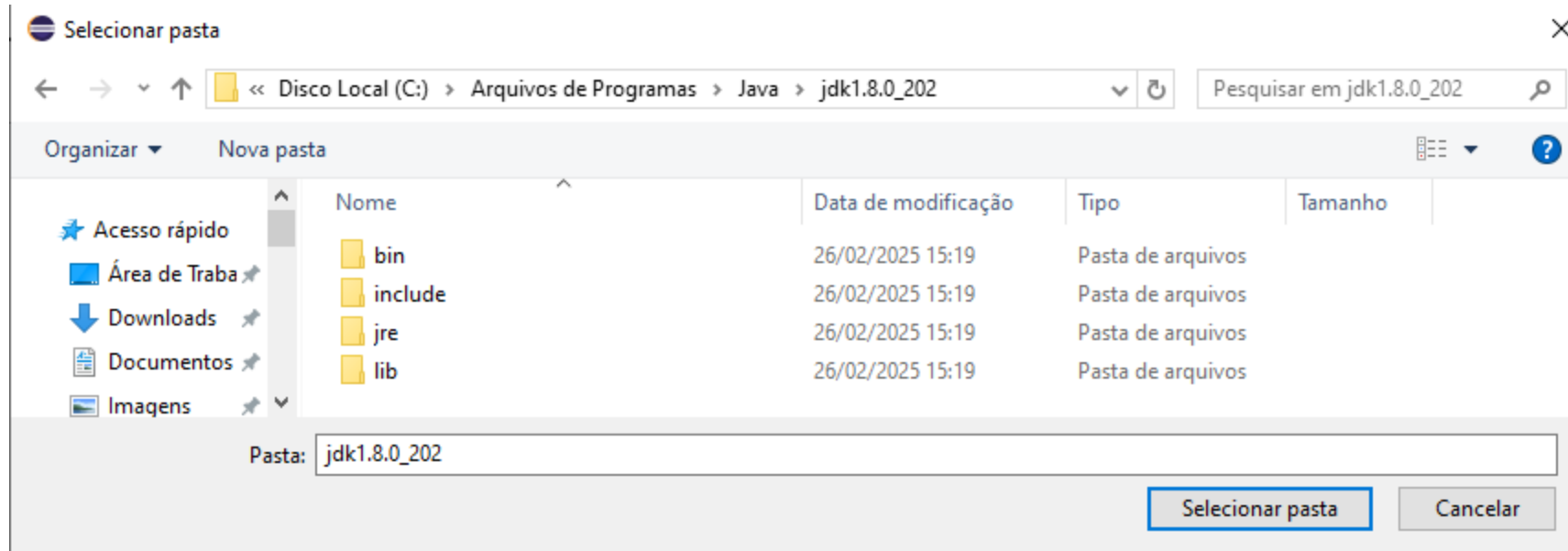
- Clique em Standard VM.

# Depois de instalar a JDK e o eclipse



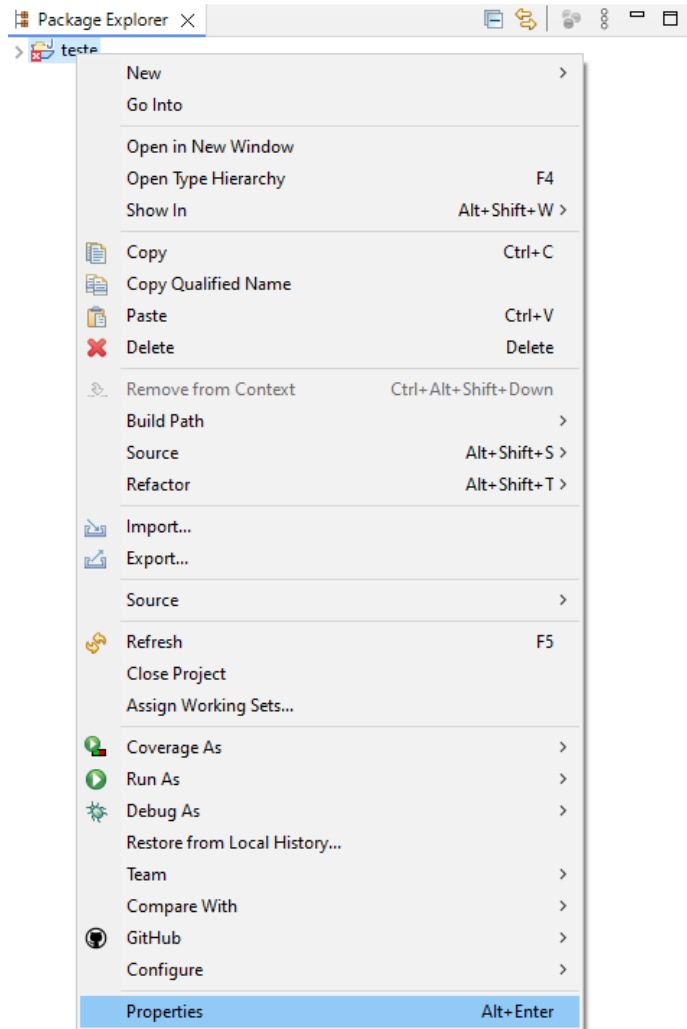
- Clique em Directory.

# Depois de instalar a JDK e o eclipse



- Escolha a pasta de instalação do JDK. C:\Program Files\Java\jdk1.8.0\_202.
- Na tela anterior clique em finish e na outra tela Apply and Close.

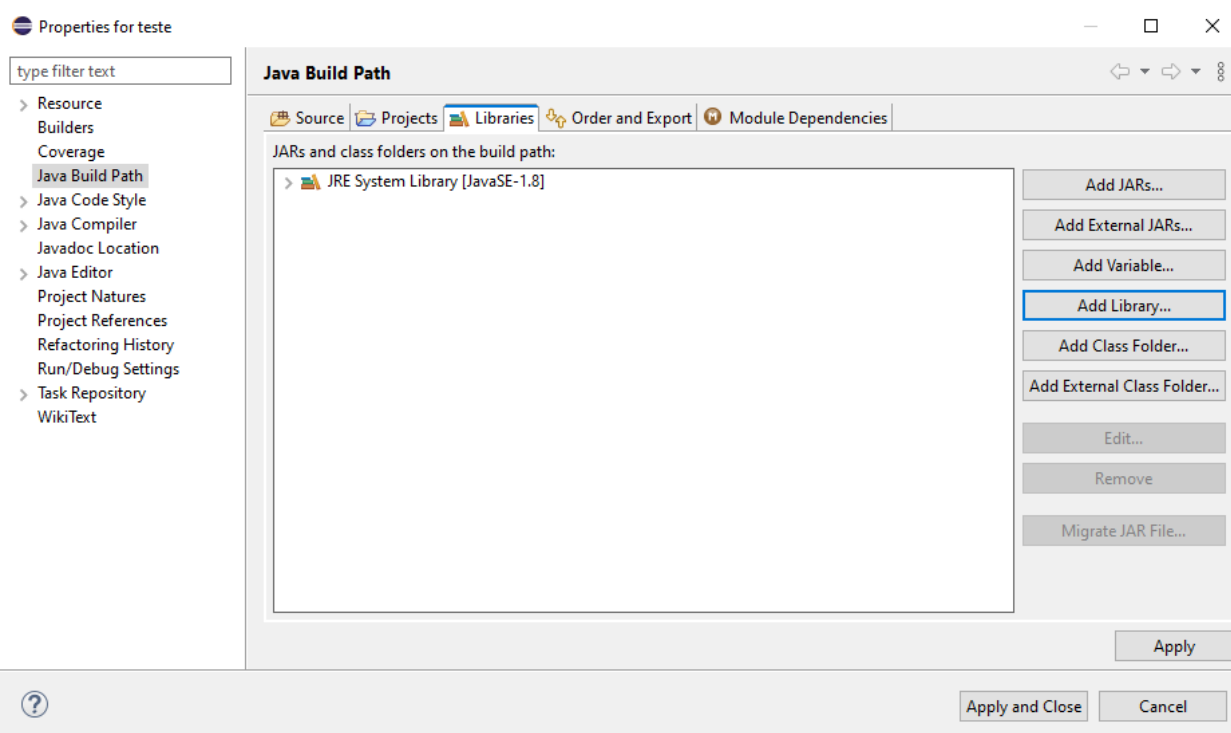
# Depois de instalar a JDK e o eclipse



- Terminando a configuração do JDK, clique em propriedades do projeto criado (nesse caso teste) em Package Explorer da tela principal.

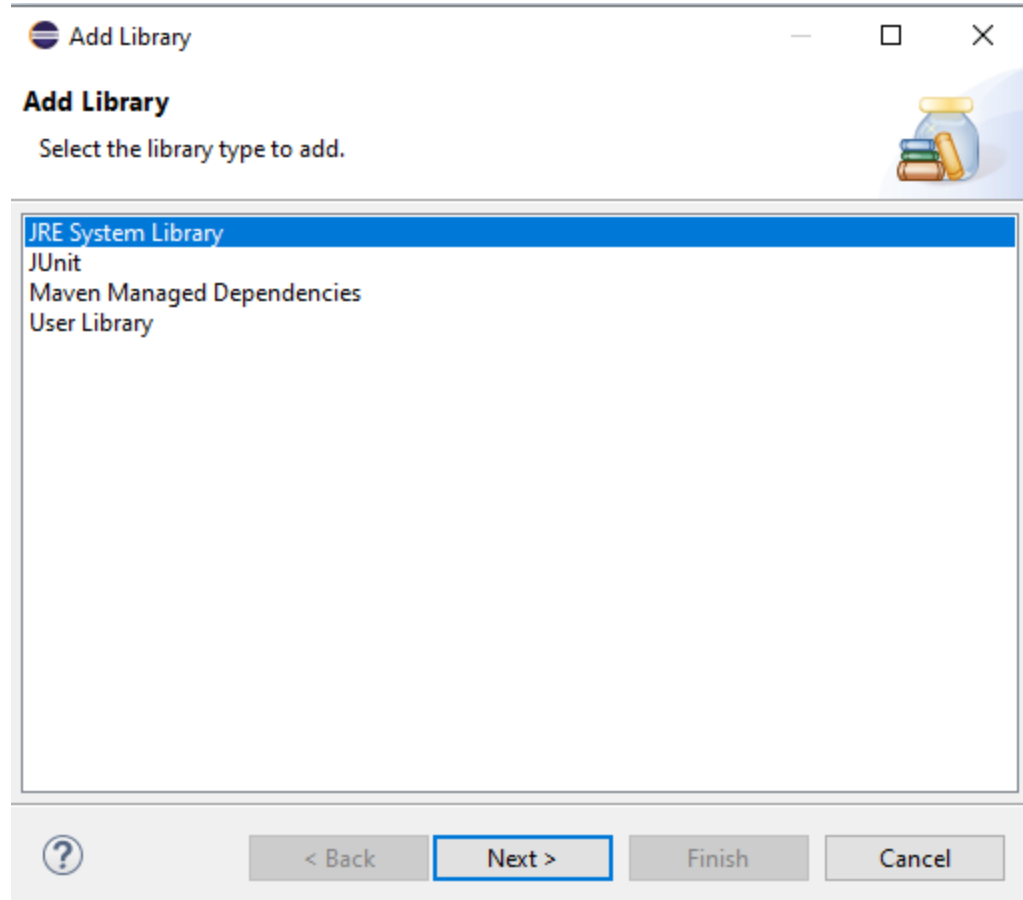


# Depois de instalar a JDK e o eclipse



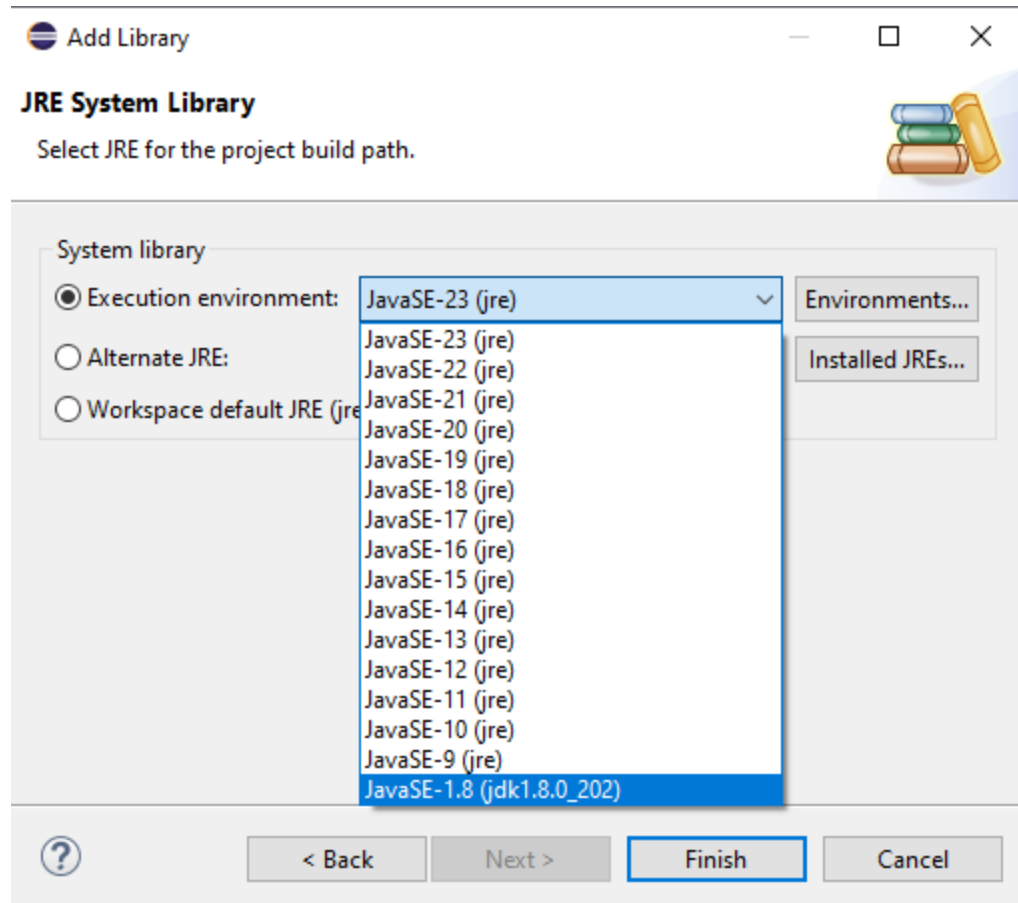
- Na esquerda escolha Java Build Path.
- Na tela a direita, remova tudo que existir, selecionando e clicando em remove, e depois clique em Add Library...

# Depois de instalar a JDK e o eclipse



- Clique em JRE System Library e depois Next.

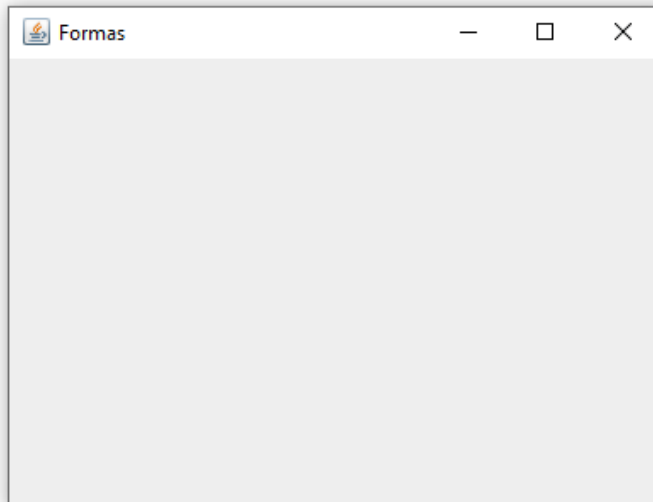
# Depois de instalar a JDK e o eclipse



- Escolha JavaSE-1.8 e clique em Finish e depois na outra janela, Apply and Close.

# Testando a configuração

```
VideoProgram.java X
1 package teste;
2 import javax.swing.*;
3
4 public class VideoProgram {
5     public static void main(String[] args) {
6         JFrame frame = new JFrame("Formas");
7         frame.setSize(400, 300);
8         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9         frame.setVisible(true);
10    }
11 }
```



- Ao executar o código ao lado deve aparecer a janela Formas.
- Caso algum erro apareça, alguma configuração está errada.

```

package teste;
import javax.swing.*;
import java.awt.*;

public class VideoProgram {
    public static void main(String[] args) {
        //Cria Janela
        JFrame quadro_pintura = new JFrame("Formas");
        quadro_pintura.setSize(400, 300);
        quadro_pintura.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel painel = new JPanel() {
            @Override
            public void paint(Graphics pincel) {
                //Cria variaveis essenciais
                super.paint(pincel);

                //Desenha um circulo vermelho
                pincel.setColor(Color.RED);
                pincel.fillOval(50, 50, 100, 100);

                // Desenha um retângulo azul
                pincel.setColor(Color.BLUE);
                pincel.fillRect(200, 50, 100, 60);

                // Desenhar um polígono (triângulo)
                pincel.setColor(Color.GREEN);
                int[] xPoints = {150, 200, 250};
                int[] yPoints = {200, 100, 200};
                pincel.fillPolygon(xPoints, yPoints, 3);
            }
        };
        quadro_pintura.add(painel);
        quadro_pintura.setVisible(true);
    }
}

```

- Aqui tem o código para criar uma imagem a Java.
- O Java AWT é bem antigo (1995), já existe algumas bibliotecas mais modernas, mas alguns conceitos dele são comuns a outras bibliotecas.
- Criamos um quadro\_pintura para desenhar algo, criamos um painel para colocar no quadro e criamos um pincel para desenhar no painel.
- setColor é usado para escolher uma cor, fillOval desenha um circulo, fillRect desenha retângulo e fillPolygon desenha um polígono.

# Inicialização de vetor x atribuição de vetor

```
6 public class VideoProgramfx {
7     public static void main(String[] args) {
8         int[] a = {1,2,3};
9         int[] b;
10
11         b = new int[] {1,7,5};
12
13         System.out.println(a[0]);
14         System.out.println(b[0]);
15         System.out.println(a);
16     }
17 }
18
```

<

Problems @ Javadoc Declaration Console X

<terminated> VideoProgramfx [Java Application] C:\Program Files

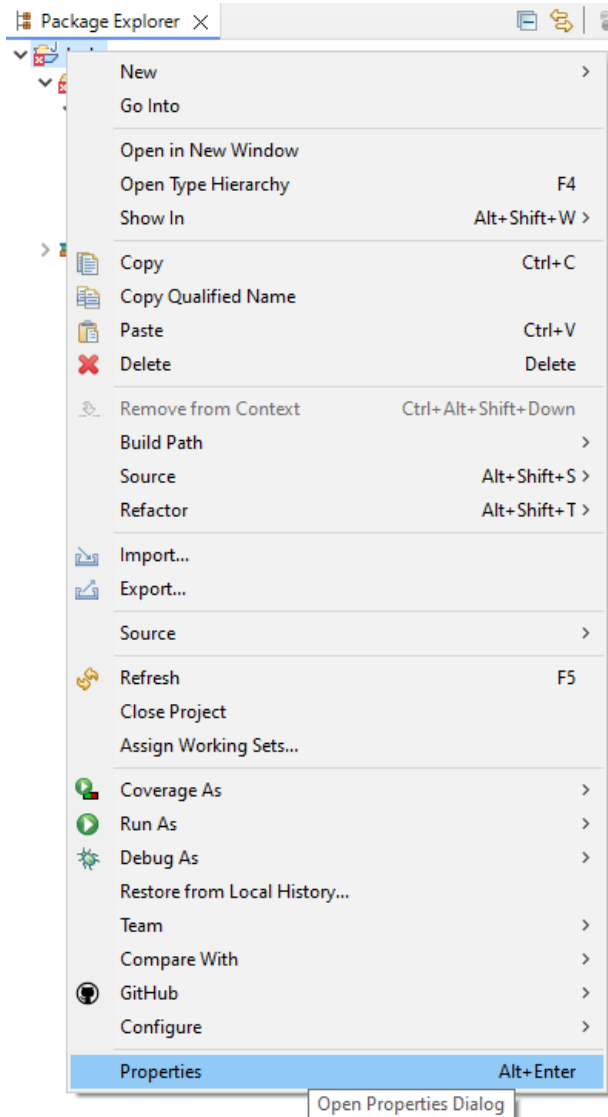
1

1

[I@15db9742

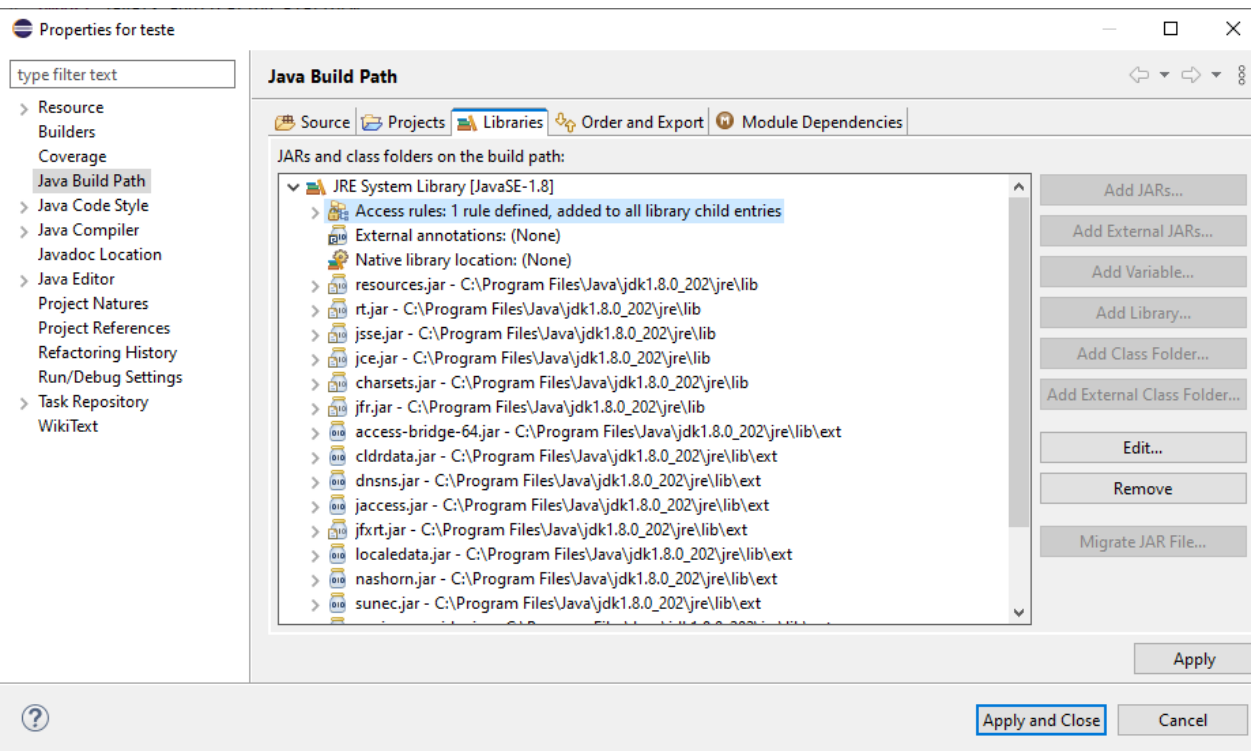
- Se for inicializado o vetor com os valores, não há a necessidade de instanciar o objeto, devido algo que será chamado de construtor.
- Um ponto interessante é que não se imprime um vetor inteiro sem uma estrutura de repetição.

# Java fx (liberando acesso)



- Por motivos de segurança o Java pode bloquear algumas bibliotecas, o Javafx está bloqueado por padrão.
- Escolha a opção propriedades.

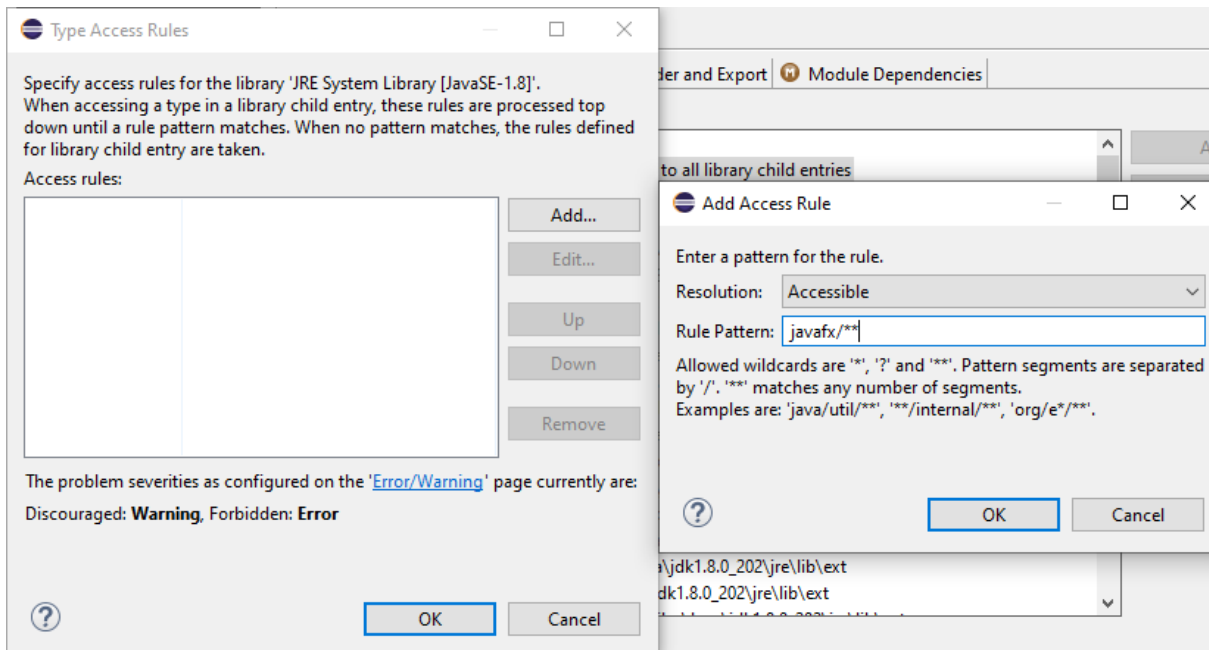
# Java fx (liberando acesso)



- Escolha Java Build Path, escolha a aba libraries, clique na flecha ao lado de JRE System Library, clique em Access rules e depois em Edit...



# Java fx (liberando acesso)



- Em type Access Rules, clique em Access Rules, clique em Add..., escolha resolution como Accessible, digite me Rule Pattern: javafx/\*\*, depois aperte ok, ok e Apply and Close.

```

package teste;
import javafx.embed.swing.JFXPanel;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;
import javafx.application.Platform;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class VideoProgramfx {
    public static void main(String[] args) {
        new JFXPanel(); // Necessário para iniciar a Thread do JavaFX

        Platform.runLater(() -> {
            Stage palco = new Stage();
            palco.setTitle("Janela Hello World");

            Circle circulo = new Circle(75, 75, 25); //(x,y,raio)
            circulo.setFill(Color.RED);

            Rectangle retangulo = new Rectangle(200, 50, 100, 60);
            retangulo.setFill(Color.BLUE);

            Polygon poligono = new Polygon();
            Double[] pontos = {150.0, 200.0, 200.0, 100.0, 250.0, 200.0};
            poligono.getPoints().addAll(pontos);
            poligono.setFill(Color.GREEN);

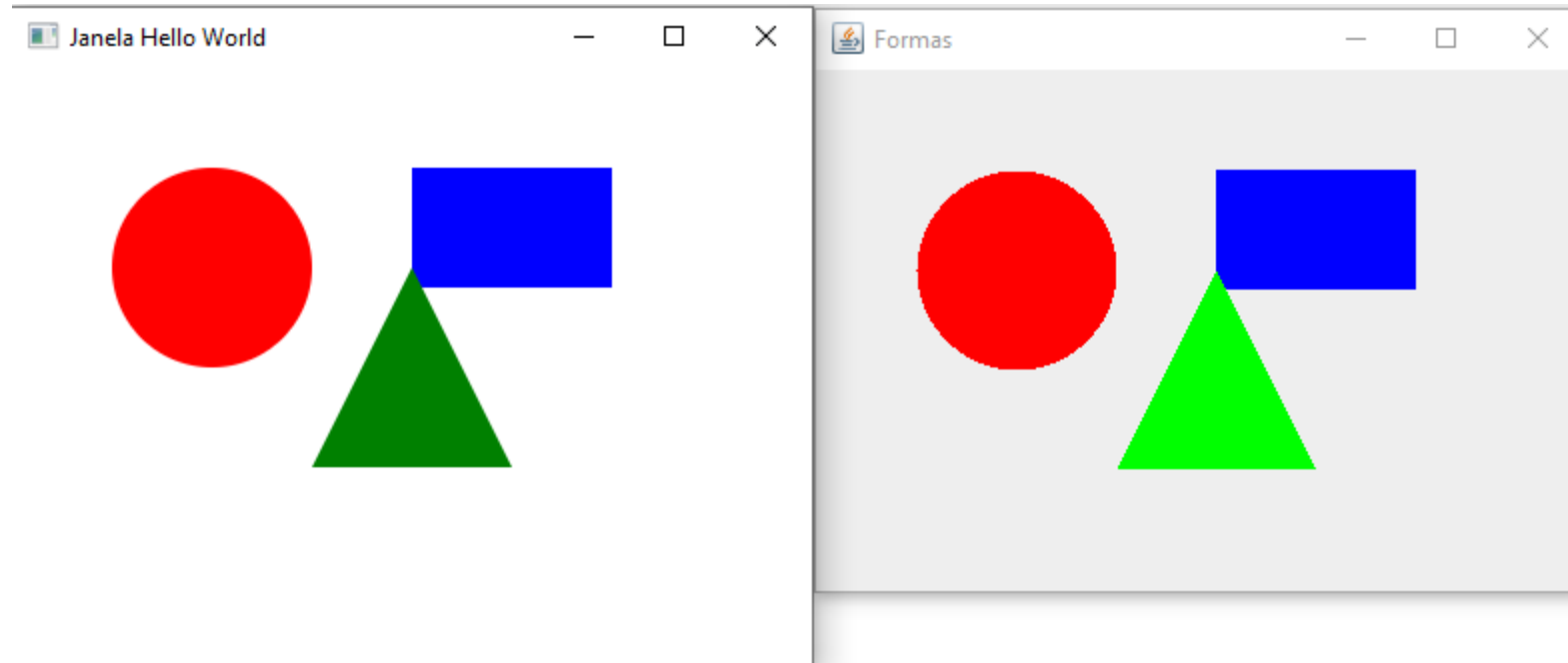
            Pane painel = new Pane();
            painel.getChildren().add(circulo);
            painel.getChildren().add(retangulo);
            painel.getChildren().add(poligono);

            Scene cena = new Scene(painel, 400, 300);
            palco.setScene(cena);
            palco.show();
        });
    }
}

```

- Das diversas opções existe o javafx, que é mais moderna (2008).
- Aqui se ativa a Thread do JavaFX, depois cria um palco, nesse palco se cria os atores (formas geométrica), os atores são colocados no painel, esse painel é colocado em uma cena que é mostrado no palco.
- Existe bibliotecas externas que podem ser melhores do que essas em certas funções.

# Comparando JavaFX e o AWT



- A esquerda JavaFX e a direita o AWT.

# Criando uma Classe com FX de Base

```
10 class Retangulo{
11     double[] pontos;
12     Color color; //Recebe Color.RED, Color.BLUE
13
14     Stage criar_palco() {
15         Stage new_stage = new Stage();
16         new_stage.setTitle("Minha primeira classe");
17         return new_stage;
18     }
19
20     Rectangle monta_retangulo() {
21         Rectangle new_rectangle = new Rectangle(pontos[0],pontos[1],pontos[2],pontos[3]);
22         new_rectangle.setFill(color);
23         return new_rectangle;
24     }
25
26     Scene mostra_retangulo(Rectangle retangulo, int tam_x, int tam_y) {
27         Pane painel = new Pane();
28         painel.getChildren().add(retangulo);
29
30         Scene new_Scene = new Scene(painel, tam_x, tam_y);
31         return new_Scene;
32     }
33
34 }
```

- Essa classe é uma classe bem básica, sem um vetor de retângulos que poderiam ser desenhados na tela, desenhando um retângulo por vez em uma tela.
- Se eu instanciar dois retângulos ele vai criar duas telas com cada uma com o seu retângulo.

# Criando uma Classe com FX de Base

```
public class VideoProgramfx {  
    public static void main(String[] args) {  
        new JFXPanel(); // Necessário para iniciar a Thread do JavaFX  
        Retangulo r1 = new Retangulo();  
        r1.pontos = new double[] {70, 70, 100, 100};  
        r1.color = Color.ORANGE;  
  
        Retangulo r2 = new Retangulo();  
        r2.pontos = new double[] {70, 70, 100, 100};  
        r2.color = Color.YELLOW;  
  
        Platform.runLater(() -> {  
            Stage palco = r1.criar_palco();  
            Rectangle new_Rectangle = r1.monta_retangulo();  
            Scene cena = r1.mostra_retangulo(new_Rectangle, 600,600);  
            palco.setScene(cena);  
            palco.show();  
  
            palco = r2.criar_palco();  
            new_Rectangle = r2.monta_retangulo();  
            cena = r2.mostra_retangulo(new_Rectangle, 600,600);  
            palco.setScene(cena);  
            palco.show();  
        });  
    }  
}
```

- Aqui de dois retângulos criados em telas diferentes.

# Exercícios

- Crie uma classe para desenhar triângulo usando qualquer uma das bibliotecas.
- Crie uma classe para desenhar círculos usando qualquer uma das bibliotecas.