

PROGRAMAÇÃO EM JAVA

MÓDULO 3

TRANSCRIÇÃO DO VÍDEO

Autor: Luiz Fernando Corcini

DETALHANDO O CÓDIGO

Conforme o vídeo, o código foi detalhado da seguinte forma:

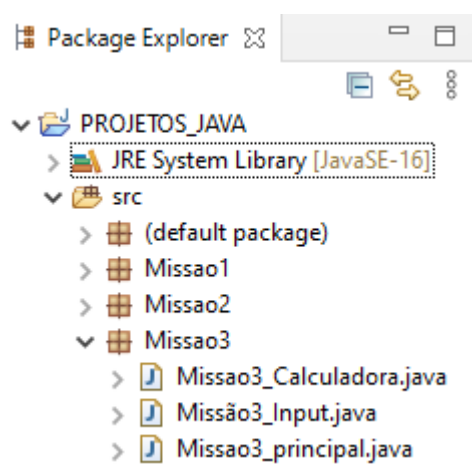
- A linha 1: **package** Missao3;

Esta linha informa que esta classe faz parte do pacote Missao3.

- A linha 3: **import** java.util.Scanner; //importar a classe Scanner

PRESTE ATENÇÃO - ESTA LINHA VOCÊ NÃO CONHECE!

A linguagem Java é útil na realização de muitos projetos com características diferentes. Por este motivo, quando instalamos o IDE ele traz consigo várias bibliotecas que podem ser importadas. Estas bibliotecas estão dentro da pasta JRE System Library, carregada no seu projeto (dentro do Package Explorer), como você pode ver na figura abaixo:



Bibliotecas Java

Se você expandir esta pasta (em destaque com círculo vermelho), você vai perceber que existe uma infinidade de classes que podem ser importadas e utilizadas. Cada uma tem uma função específica!

Faça o seguinte teste:

- Clique e expanda a pasta JRE System Library. Você vai encontrar, logo em primeiro lugar, uma outra pasta com o nome de “java.base”. Mas repare que existem outras, tais como “java.sql”, “java.xml” e muitas outras pastas.
- Clique e expanda esta pasta “java.base” e você vai encontrar muitos pacotes, dentre os quais você vai conseguir identificar: “java.math”, “Java.io”, “java.time” e “java.util”. Este último é o pacote de utilidades Java. Dentro deste pacote, você vai encontrar a classe “Scanner.class”. É esta classe que estamos importando agora.

De acordo com o site DevMedia.com.br: Em linguagens de programação estruturada, como C e Pascal, a entrada de dados usando o teclado é realizada utilizando subprogramas desenvolvidos para esta finalidade. Na linguagem Java, a partir do Java 1.5 ou Java 5.0, o pacote de classes “java.util” disponibilizou a classe “Scanner”, que implementa operações de entrada de dados pelo teclado.

DICA: Para saber se é necessário importar uma classe antes de usar um comando faça sempre uma pesquisa em sites especializados em programação.

- A linha 5: **public class** Missao3_Input //início da classe

Linha de código que você já conhece e que declara o início da classe Missao3_Input.

- A Linha 6: Abre chaves para realmente iniciar a classe.
- A linhas 8: **public static void** main(String[] args) //início do método principal

Linha de código que você já conhece e que declara o início do método principal.

- A linha 9: Abre chaves para realmente iniciar o método principal.
- A linha 10: Scanner **entrada** = **new** Scanner(System.**in**);

PRESTE ATENÇÃO -- ESTA LINHA VOCÊ NÃO CONHECE!

Esta linha de código declara um objeto chamado “entrada” (poderia ser qualquer outro nome). Este objeto, a partir desta linha recebe toda as características (atributos) e comportamentos (métodos) que a classe Scanner disponibiliza. Algumas destes serão vistos logo abaixo.

- A linha 12: System.**out**.print(“**Digite uma frase:**”); //Mostrando a mensagem 1

Esta linha você já conhece. Ela mostra no console a mensagem que está entre as aspas duplas.

- A linha 13: String **Texto** = **entrada**.nextLine();

PRESTE ATENÇÃO -- ESTA LINHA VOCÊ NÃO CONHECE!

Esta linha declara uma variável do tipo String (Em Java se utiliza o tipo String para variáveis que vão armazenar textos);

Depois do sinal de “=” o objeto entrada utiliza um método chamado “nextLine()” cuja função é esperar que você digite uma frase ou um conjunto de palavras separadas por espaços;

Para informar ao computador que você acabou de escrever a frase, você precisa pressionar a tecla ENTER.

- A linha 15: `System.out.println("Digite um número inteiro:");`

Esta linha você já conhece. Apenas mostra no console a mensagem que está entre aspas. Esta mensagem só vai aparecer depois que você pressionar ENTER (no item anterior).

- A linha 16: `int N_int = entrada.nextInt();`

PRESTE ATENÇÃO -- ESTA LINHA VOCÊ NÃO CONHECE!

O método “nextInt()” utilizado pelo objeto entrada faz com que o computador entenda que, desta vez, ele deve esperar que o usuário digite um número inteiro e não um texto e nem um caractere ou número real.

Depois que você digitar o número inteiro, precisa pressionar a tecla ENTER para o computador entender que ele pode continuar a ler e executar o código.

- A linha 18: `System.out.println("Digite um número real:");`

Esta linha você já conhece. Tem as mesmas características da linha 15.

- A linha 19: `double N_double = entrada.nextDouble();`

PRESTE ATENÇÃO -- ESTA LINHA VOCÊ NÃO CONHECE!

O método “nextDouble()” utilizado pelo objeto entrada faz com que o computador entenda que, desta vez, ele deve esperar que o usuário digite um número real, isto é, com vírgula.

Depois que você digitar o número inteiro, precisa pressionar a tecla ENTER para o computador entender que ele pode continuar a ler e executar o código.

- A linha 21: `System.out.println("Digite uma Letra ou palavra:");`

Esta linha você já conhece. Tem as mesmas características da linha 15 e 18.

- A linha 22: `String Palavra = entrada.next();`

PRESTE ATENÇÃO -- ESTA LINHA VOCÊ NÃO CONHECE!

O método “next()” utilizado pelo objeto entrada faz com que o computador entenda que, desta vez, ele deve esperar que o usuário digite uma palavra (sem espaços) ou uma simples letra.

Depois que você digitar o número inteiro, precisa pressionar a tecla ENTER para o computador entender que ele pode continuar a ler e executar o código.

- Linhas 24 a 28: `//mostrando os resultados`

`System.out.println("A frase digitada foi: " + Texto);`

`System.out.println("O número inteiro digitado foi: " + N_int);`

`System.out.println("O número real foi: " + N_double);`

`System.out.println("A Letra ou Palavra digitada foi: " + Palavra);`

Estas linhas você já conhece e mostram o que foi digitado e armazenado nas variáveis.

- Linha 31: `entrada.close();`

Como a entrada de dados acabou, é boa prática fechar o objeto para que nada de ruído ou sujeira seja captado.

- Linha 33: fecha chaves, indicando que o método principal acabou.
- Linha 35: fecha chaves, indicando que a classe acabou.

FIM DA ANÁLISE!!!