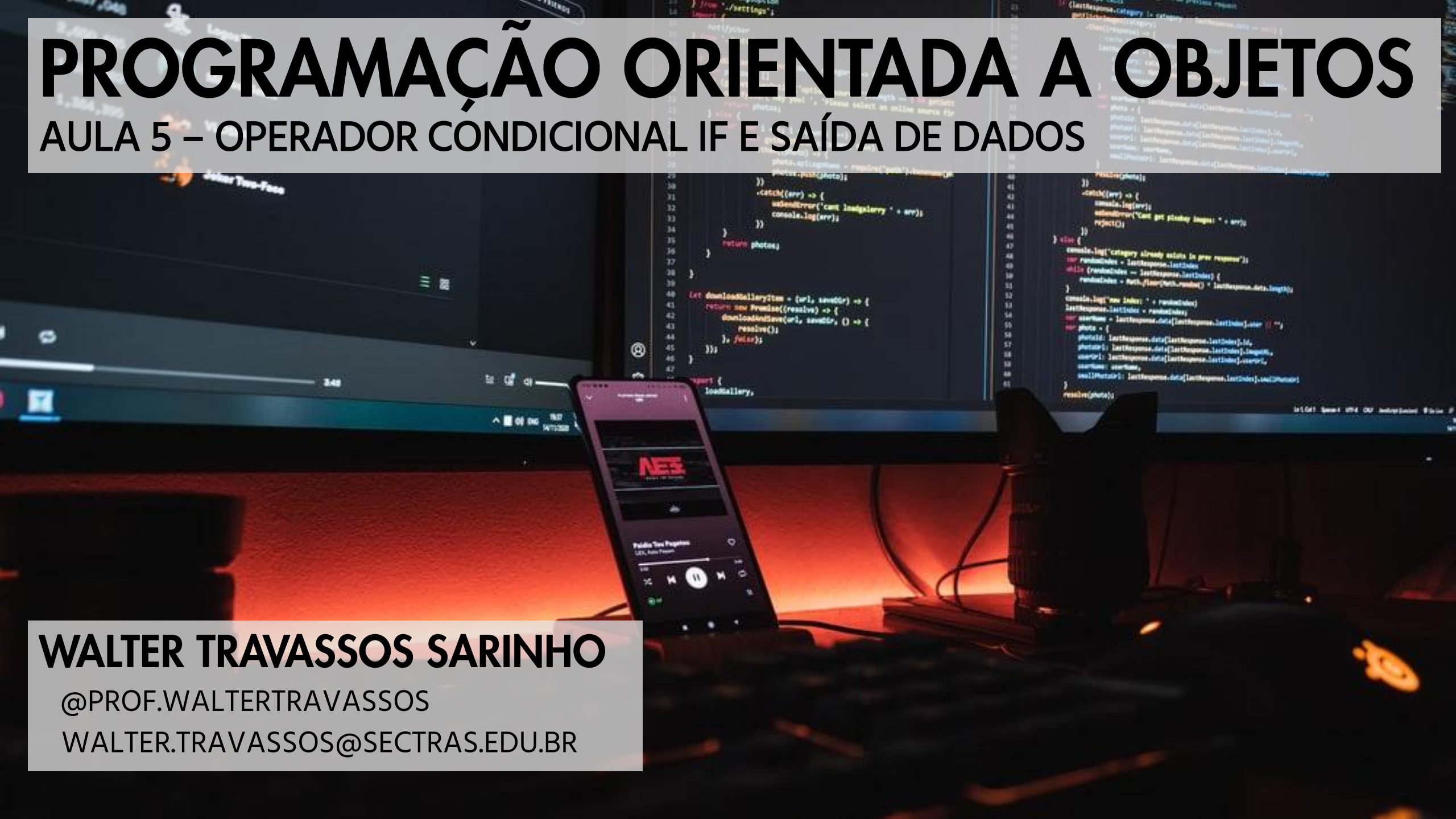
PROGRAMAÇÃO ORIENTADA A OBJETOS

AULA 5 – OPERADOR CONDICIONAL IF E SAÍDA DE DADOS



WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR

Dúvidas sobre a aula passada

Aula de Hoje

- Formatar a saída de dados numérica.
- Apresentar a estrutura condicional if.
- Conhecer alguns métodos para manipulação e comparação de Strings.
- Praticar!

Mapa de Estudos

AULA 01
Paradigmas de Programação e história do Java.
IDEs e Hello Universe

AULA 03
Visibilidade de variáveis, classe String e casting.

AULA 02
Manipulação de datas.
Sintaxe da Linguagem Java e variáveis primitivas e de referência.

AULA 04
Classes Wrapper.
Operadores e métodos de conversão.

AULA 05
Formatar a saída de dados numérica.
Operador condicional if.
Métodos para comparar Strings

Avaliação 1

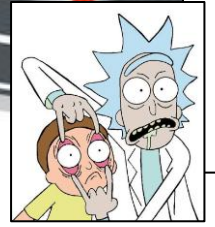
Java

Mapa de Estudos

AULA 06
Manipulação de Strings. Operador condicional switch.

AULA 08
Vetores (for each) e matrizes. Classe ArrayList

AULA 07
Operadores de atribuição compostos. Estruturas de repetição: while, do while e for.



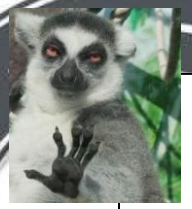
AULA 09
O paradigma Orientado a Objetos

AULA 10
Construindo métodos e Métodos construtores

Avaliação 2

Java

Mapa de Estudos



AULA 12
Classes Abstratas e Interfaces

AULA 11
Pilares da POO: Abstração, Encapsulamento, Herança e Polimorfismo

Avaliação 3

Java

Formatando a saída de dados numérica

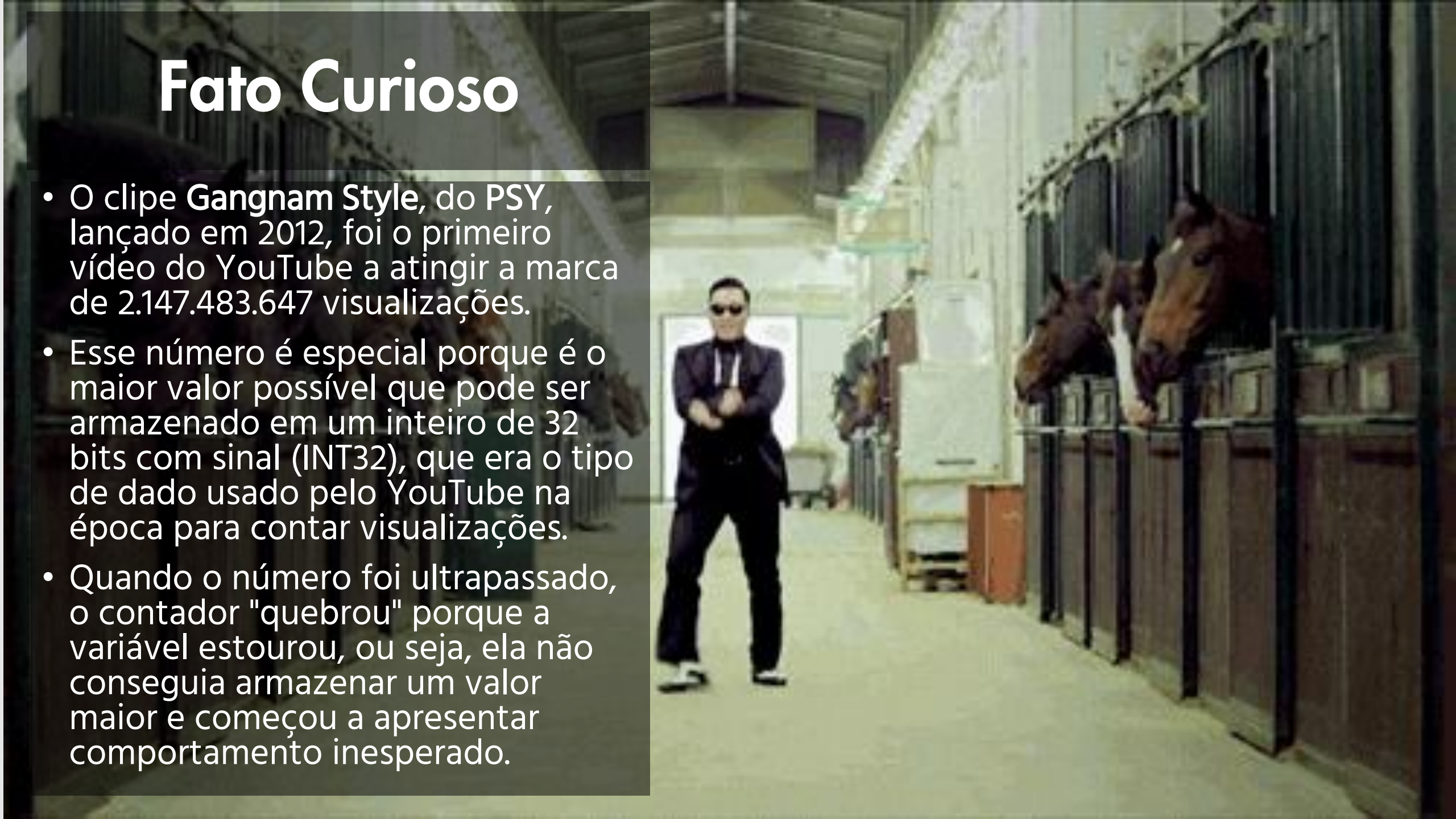
Formatando a saída de dados numérica

Muitos alunos perguntam como formatar a saída de dados de uma variável do tipo **double** com **duas casas decimais.**

- Existem diversos objetos que lidam com essa saída, como: **DecimalFormat**, **BigDecimal**...
- Hoje apresentaremos a formatação de saída de dados com os comandos **System.out.printf()** e com **String.format()**.

Fato Curioso

- O clipe **Gangnam Style**, do **PSY**, lançado em 2012, foi o primeiro vídeo do YouTube a atingir a marca de 2.147.483.647 visualizações.
- Esse número é especial porque é o maior valor possível que pode ser armazenado em um inteiro de 32 bits com sinal (INT32), que era o tipo de dado usado pelo YouTube na época para contar visualizações.
- Quando o número foi ultrapassado, o contador "quebrou" porque a variável estourou, ou seja, ela não conseguia armazenar um valor maior e começou a apresentar comportamento inesperado.



Fato Curioso

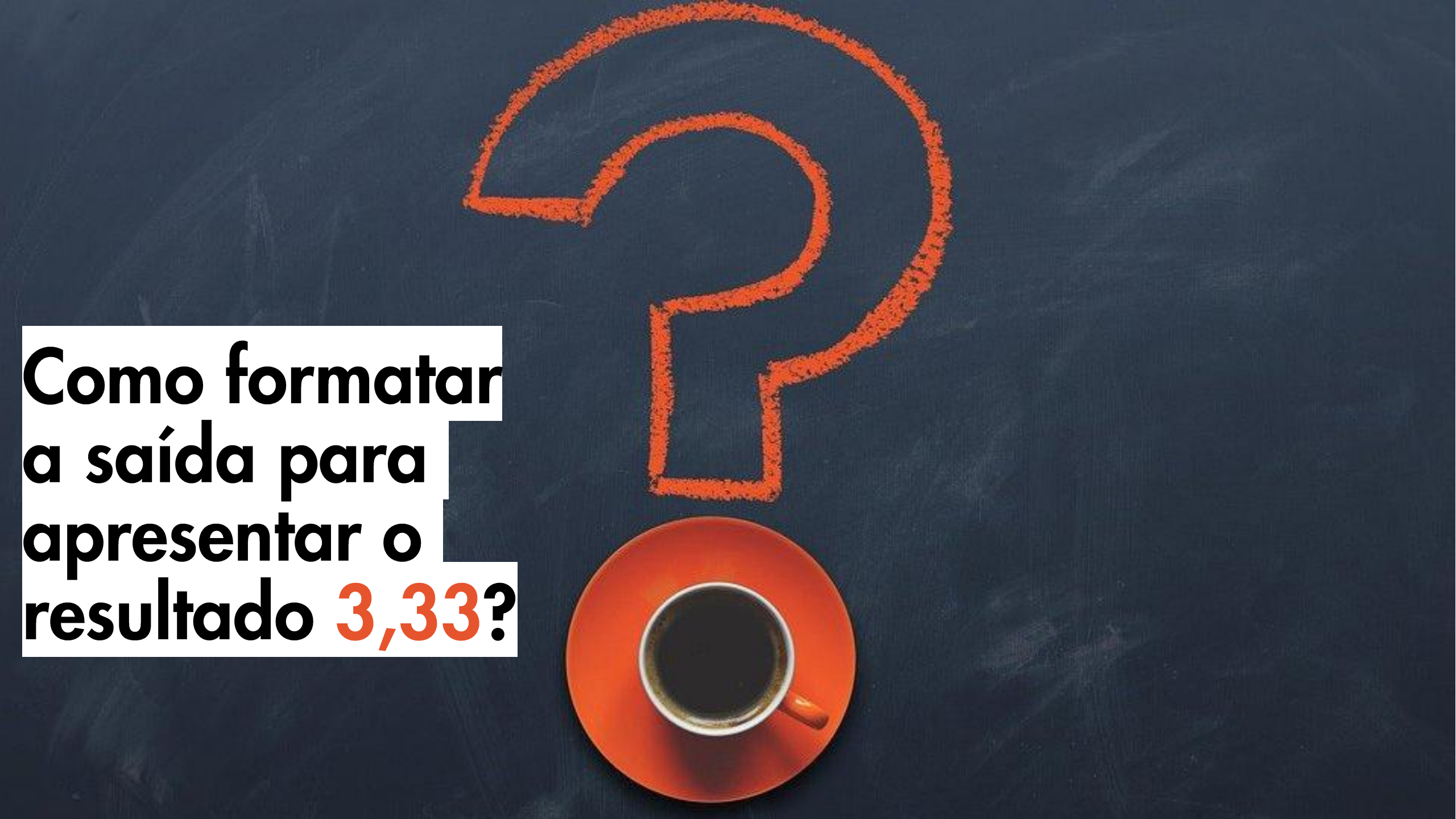
- O YouTube teve que fazer uma atualização de emergência, mudando o contador para um inteiro de 64 bits (INT64), permitindo armazenar números absurdamente maiores (até 9 quintilhões de visualizações).
- Eles até brincaram com a situação, postando uma mensagem dizendo: "**Nunca pensamos que um vídeo seria assistido tantas vezes!"**
- Depois desse evento, o YouTube garantiu que o problema nunca mais aconteceria, e hoje os vídeos podem acumular um número praticamente infinito de visualizações sem quebrar. 😆

Problematização

- Considere o seguinte trecho de código:

```
double n1 = 10;
double n2 = 3;
double resultado = n1/n2;
System.out.println(resultado);
```

- A saída do console com o método println() será:
  3.3333333333333335


Como formatar a saída para apresentar o resultado 3,33?

System.out.printf()

- Ao invés de utilizar **println**, utilize **printf**!
- É preciso passar dois argumentos ao método:

  (i) o formato de saída e (ii) a(s) variável(eis).

- Sintaxe: System.out.printf(arg1, arg2);

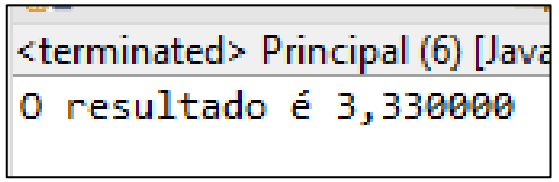
System.out.printf(arg1, arg2);

O arg1 é o formato de saída. Utilizamos **%d** para números inteiros e **%f** tanto para números em ponto flutuante quanto para inteiros. Este argumento deve ser passado entre as aspas.

Por exemplo:

```
double resultado = 3.33;
System.out.printf("O resultado é: %f", resultado);
```

SÓ QUE O RESULTADO AINDA ESTÁ COM MUITAS CASAS DECIMAIS.

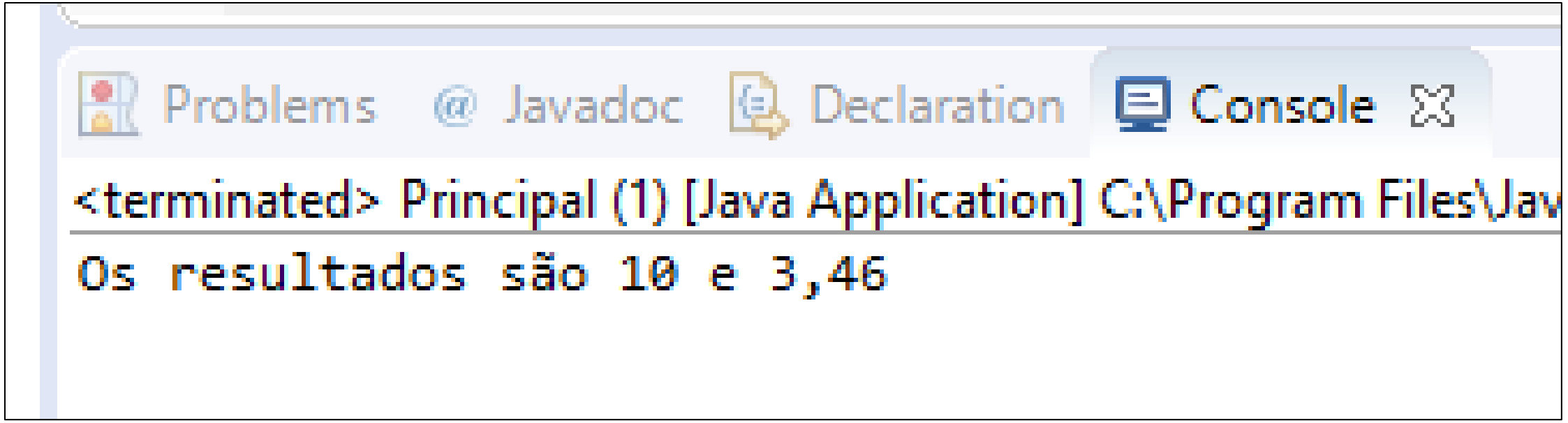


System.out.printf("%.2f", resultado);

- Utilizar apenas o **%f** não irá alterar o resultado. É preciso passar a quantidade de casas que se deseja exibir.
- Entre o **%** e o **f**, colocamos um ponto seguido da quantidade de casas decimais, por exemplo: **%.2f**.
- Utilizar o **.2** significa que serão exibidas apenas 2 casas decimais após o ponto.
- Por exemplo: **System.out.printf("%.2f", resultado);**
- O resultado exibido será 3,33.

Podemos exibir mais de uma variável

```
int i = 10;
double d = 3.456;
System.out.printf("Os resultados são %d e %.2f ", i, d);
```



String.format()

- Ainda é possível formatar a saída dentro do System.out.println com o método String.format.
- Utilizamos esse método para formatar a saída do println.
- Por exemplo:

```
System.out.println(resultado);
System.out.println(String.format(" %.2f ", resultado) );
```

- Atente que utilizamos o mesmo formato do printf: **"%.2f";**

Conversão para String

- Aprendemos diversos tipos de conversão do formato String para o tipo numérico, no entanto, se quisermos fazer o contrário: converter do formato numérico para String?
- Que método utilizamos?
- Resposta: **String.valueOf()**

String.valueOf()

- Utilizaremos este método no **netbeans** para exibir algum valor numérico num componente visual da biblioteca java.Swing.
- Deve-se passar a variável numérica como argumento do método **valueOf(arg)** e salvar sua saída numa variável String.
- **Fazer demonstração no Netbeans.**

Por exemplo:

double **i** = 3.5;

String str = String.valueOf(**i**);

Operadores Condicionais

Estruturas de Decisão

Estruturas Condicionais

- São estruturas que seguem a sintaxe e regras definidas na estrutura “se” e “escolha...caso” estudadas em algoritmos.
- Em Java, existem dois tipos de estruturas de decisão:
  - if...else
  - switch

Sintaxe do if

```
if (condição) {
        comando_1;
        comando_2;
        ...
        comando_n;
        //Comandos executados caso a condição seja verdadeira
  }
```

Sobre a "condição"

- if (condição) { comandos }

- Essa condição é uma expressão **booleana** ou seja, deve retornar *true* (verdadeiro) ou *false* (falso).
- Você pode colocar uma expressão condicional (**a>b**, por exemplo) ou uma variável booleana dentro da condição.
- Você também pode colocar algum método que retorna um valor booleano, como por exemplo, o método **equals** (daqui a pouco falaremos sobre ele).

Estrutura if .. else

```
if (condição) {
        //Comandos executados caso a condição seja verdadeira
} else {
        // Comandos executados caso a condição seja falsa
}
```

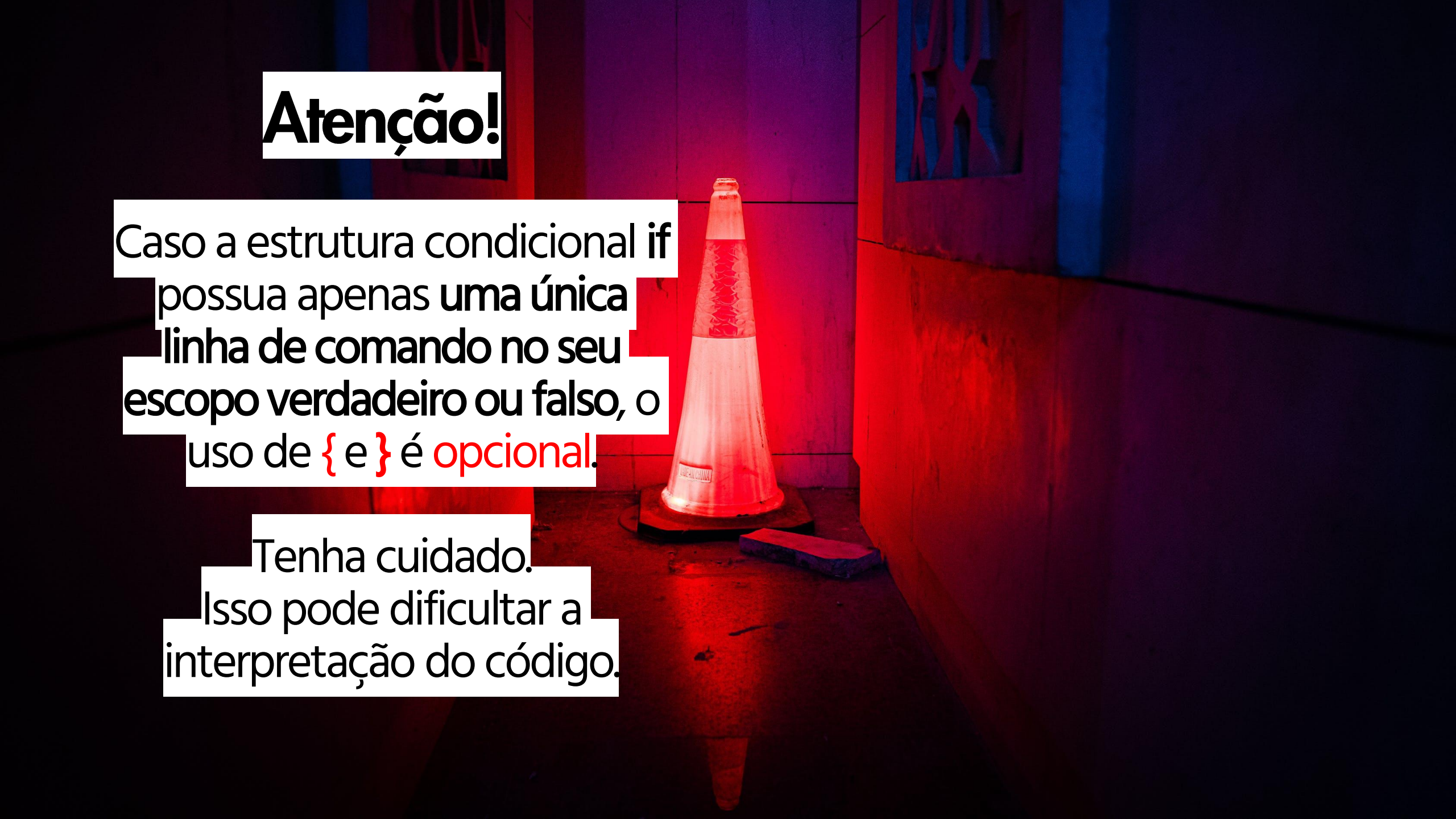
Estruturas Condicionais

- if...else if...else

- Trata-se de uma estrutura condicional if não limitada a duas condições.
- Podemos combinar um else com um if para testar possibilidades mutuamente exclusivas.

Sintaxe

```
if (condição)  {
        // Lista de instruções
}
else if (condição 2)  {
        // Lista de instruções
}
else if (condição 3)  {
        // Lista de instruções
}
else  {
        // Lista de instruções
}
```



Atenção!

Caso a estrutura condicional **if** possua apenas **uma única linha de comando no seu escopo verdadeiro ou falso**, o uso de { e } é opcional.

Tenha cuidado.
Isso pode dificultar a interpretação do código.

Exercício 1

Resposta: 13

Qual o valor irá aparecer no console de acordo com o trecho de código abaixo?

```
int i = 11;
if (i > 10)
i++;
i++;
System.out.println(i);
```

Exercício 2

Resposta: 11

Qual o valor irá aparecer no console de acordo com o trecho de código abaixo?

```
int i = 10;
if (i > 10)
i++;
i++;
System.out.println(i);
```

Fique Atento!

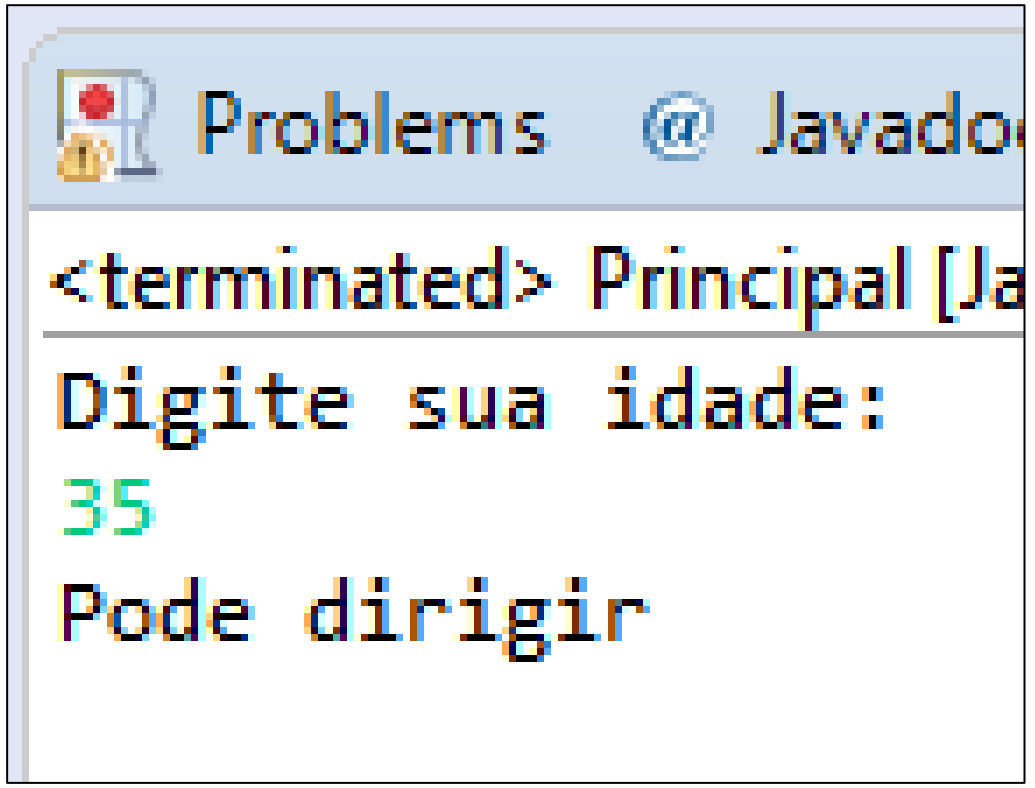
- Quando não delimitamos o escopo dos comandos que irão ser executados com as chaves, só fará parte do escopo do bloco Verdadeiro (ou Falso) do if apenas a linha de código seguinte. Apenas ela.
- Dessa forma, as duas linhas de código a seguir terão o mesmo resultado:

```
if (i > 10) i++;
if (i > 10) { i++; }
```

Exercício 3

Sabendo que a idade necessária para obter habilitação é 18 anos, faça um programa em Java para receber do usuário a idade de uma pessoa e informar se ela **pode** ou **não pode** dirigir.

Vamos resolver esse juntos!



Resposta Exercício 3

Principal.java

```java
import java.util.Scanner;

public class Principal {
    public static void main (String[] args) {
        Scanner entrada = new Scanner(System.in);
        int idade;
        System.out.println("Digite sua idade:");
        idade = entrada.nextInt();
        if(idade >= 18){
            System.out.println("Pode dirigir");
        } else {
            System.out.println("Não pode dirigir");
        }
    }
}
```

Problems @ Javado

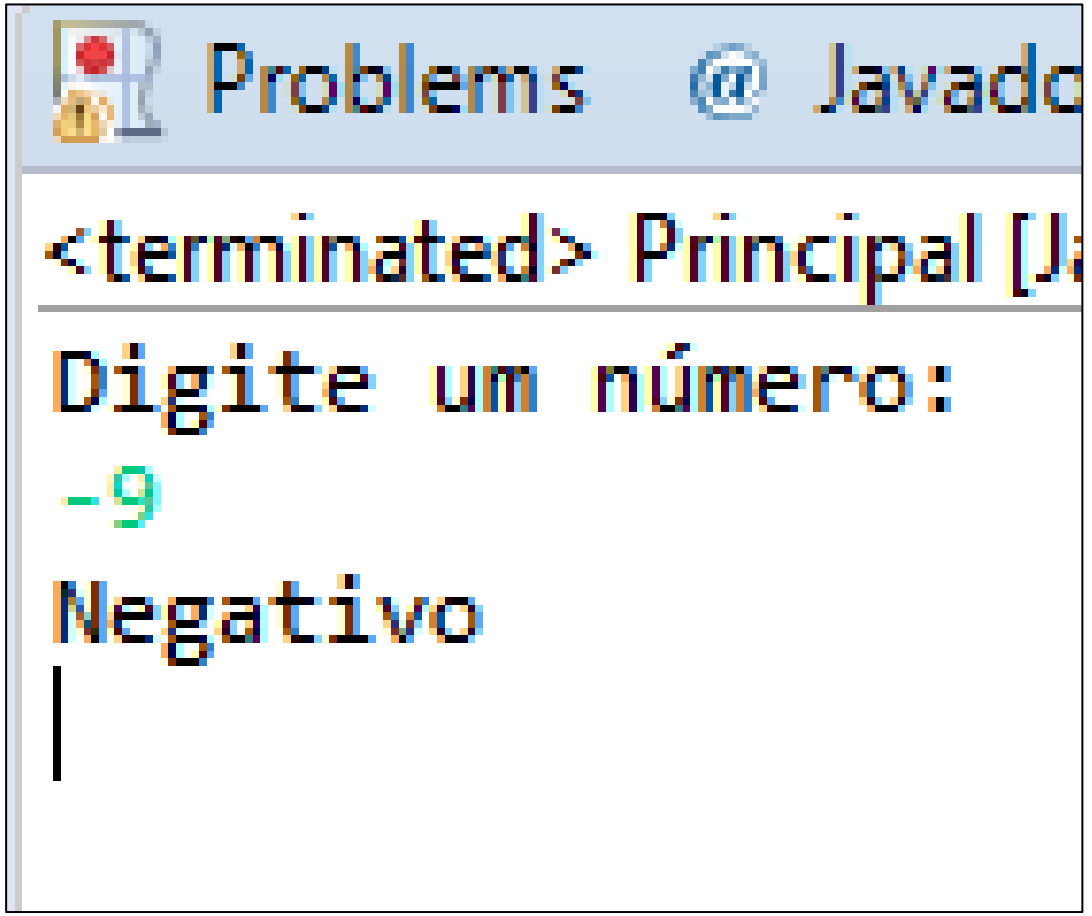
<terminated> Principal [Ja

```
Digite sua idade:
35
Pode dirigir
```

Exercício 4

Faça um programa em Java que receba do usuário um número e determine se o mesmo é **positivo, negativo ou zero.**

Tempo: 5 minutos



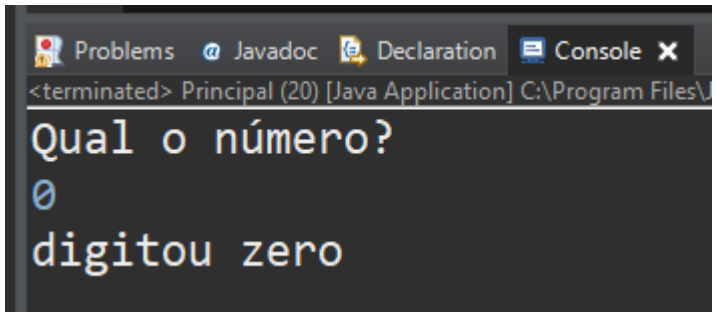
Resposta Exercício 4

```java
import java.util.Scanner;

public class Principal {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.println("Qual o número?");
        int n = teclado.nextInt();

        if (n > 0)
            System.out.println("positivo");
        else if ( n < 0)
            System.out.println("negativo");
        else
            System.out.println("digitou zero");

    }
}
```



Exercício 5

Faça um programa que receba a média do aluno e diga se ele passou por média, foi reprovado ou foi pra prova final. Use um if aninhado para resolver esse problema.

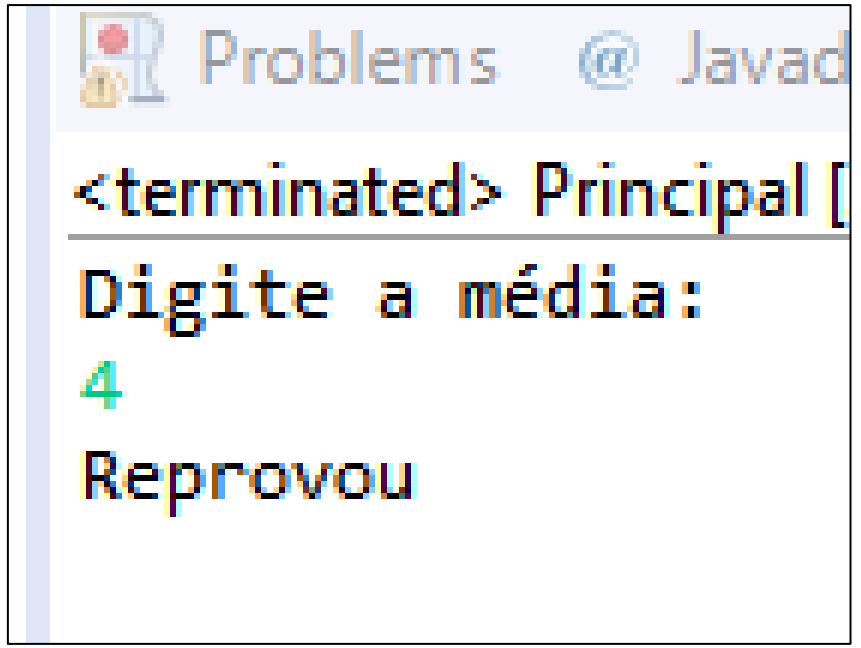
Regras

Média >= 7 **passou por média**

Média >=5 e Média <7 **vai pra prova final**

Média < 5 **reprovou**

Tempo: 5 minutos



Resposta Exercício 5

Principal.java

```
1  import java.util.Scanner;
2
3  public class Principal {
4      public static void main (String[] args) {
5          Scanner entrada = new Scanner(System.in);
6          int media;
7
8          System.out.println("Digite a média:");
9          media = entrada.nextInt();
10
11         if(media >= 7){
12             System.out.println("Passou por média");
13         } else  if (media >=5 ){
14             System.out.println("Vai fazer final");
15         }
16         else {System.out.println("Reprovou");
17         }
18     }
19 }
20
```

Problems @ Javad

<terminated> Principal [

```
Digite a média:
4
Reprovou
```

Intervalo: 15 minutos

Comparação de Strings

Método equals()

Método equals()

- Compara o conteúdo de dois objetos do tipo String. Caso sejam iguais, retorna true. Caso não sejam, retorna false.
- O método é *case sensitive*. Logo: **João != joão**;

Sintaxe:

String objeto1, objeto2;

objeto1.equals(objeto2)

OBS: Este método pode ser utilizado dentro da condição do **if** pois ele retorna V ou F!

Método equalsIgnoreCase()

Funciona do mesmo jeito que o método equals, no entanto ignora o case, ou seja, uma comparação entre as Strings **João** e **joão** resultaria num valor verdadeiro.

Sintaxe:

String objeto1, objeto2;

objeto1.equalsIgnoreCase(objeto2)

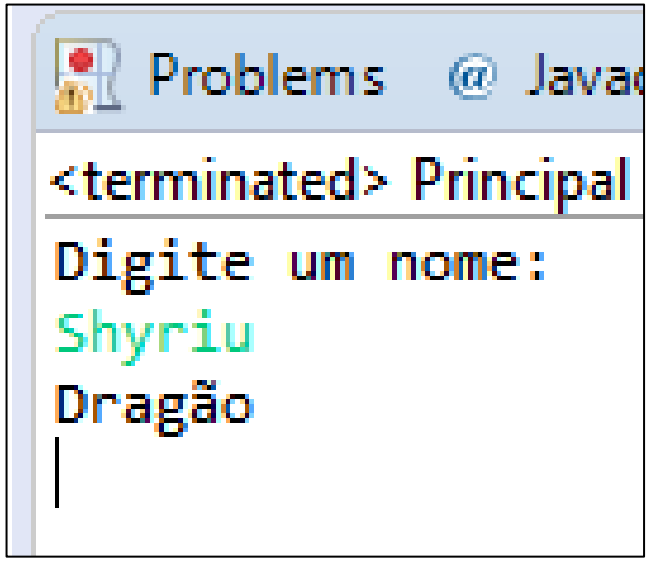
Exercício 6

Usando o método **equals**, faça um programa que recebe do usuário um dos nomes dos cinco cavaleiros de bronze e em seguida apresenta sua constelação guardiã.

Ex.: Se o usuário digitar o nome “Seiya” seu programa deverá escrever “Pégasus”.

Lista: Seiya (Pégasus), Hyoga (Cisne), Shiryu (Dragão), Shun (Andromeda) e Ikki (Fenix).

Tempo: 5 minutos



Resposta Exercício 6

Principal.java

```java
import java.util.Scanner;

public class Principal {
    public static void main (String[] args) {
        Scanner entrada = new Scanner(System.in);
        String nome;
        System.out.println("Digite um nome:");
        nome = entrada.nextLine();

        if (nome.equals("Seiya")) System.out.println("Pegasus");
        else if (nome.equals("Ikki")) System.out.println("Fenix");
        else if (nome.equals("Hyoga")) System.out.println("Cisne");
        else if(nome.equals("Shyriu")) System.out.println("Dragão");
        else if(nome.equals("Shun")) System.out.println("Andromeda");
        else System.out.println("Cavaleiro Desconhecido");
    }
}
```

Problems @ Javac

<terminated> Principal

```
Digite um nome:
Shyriu
Dragão
```

Método compareTo

Exemplo:

s1.compareTo(s2)

- Vai comparar duas Strings!
- Diferente do método **equals**, é possível obter 3 resultados diferentes:

1. As Strings são iguais. A resposta do método será **0**.
2. A **primeira é maior que a segunda**, a resposta será **maior que zero**.
3. A **primeira é menor que a segunda**, a resposta será **menor que zero**.

Método compareTo

Como os resultados do método são números, é preciso criar uma expressão booleana para utilizar este método dentro de estruturas condicionais.

Por exemplo:

if (**s1.compareTo(s2) > 0**) System.out.println(“s1 é maior”);

if (**s1.compareTo(s2) < 0**) System.out.prinltn(“s2 é maior”);

Maior... Significa mais próximo da letra Z

Menor... Significa mais próximo da letra A

A comparação das Strings é realizada pela tabela ASCII - Caractere a caractere

Tabela ASCII (códigos de caracteres 0 - 127)

000	016 ►	032	048 0	064 @	080 P	096 `	112 p
001 ☺	017 ◄	033 !	049 1	065 A	081 Q	097 a	113 q
002 ☻	018 ↕	034 "	050 2	066 B	082 R	098 b	114 r
003 ♥	019 ‼	035 #	051 3	067 C	083 S	099 c	115 s
004 ♦	020 ¶	036 $	052 4	068 D	084 T	100 d	116 t
005 ♣	021 §	037 %	053 5	069 E	085 U	101 e	117 u
006 ♠	022 ■	038 &	054 6	070 F	086 V	102 f	118 v
007	023 ↨	039 '	055 7	071 G	087 W	103 g	119 w
008	024 ↑	040 (	056 8	072 H	088 X	104 h	120 x
009	025 ↓	041)	057 9	073 I	089 Y	105 i	121 y
010	026 →	042 *	058 :	074 J	090 Z	106 j	122 z
011 ♂	027 ←	043 +	059 ;	075 K	091 [	107 k	123 {
012 ♀	028 ∟	044 ,	060 <	076 L	092 \	108 l	124 |
013	029 ↔	045 -	061 =	077 M	093]	109 m	125 }
014 ♫	030 ▲	046 .	062 >	078 N	094 ^	110 n	126 ~
015 ☼	031 ▼	047 /	063 ?	079 O	095 _	111 o	127 ⌂

Qual o resultado das expressões condicionais das linhas 2, 3 e 4?

1. String nome=“jose”;
2. if (nome.equals(“joao”))...
3. if (**!**nome.equals(“maria”))...
4. if (“ze”.compareTo(“antonio”) **< 0**)...

Qual o resultado das expressões condicionais das linhas 2, 3 e 4?

1. String nome=“jose”;
2. if (nome.equals(“joao”))...
3. if (**!**nome.equals(“maria”))...
4. if (“ze”.compareTo(“antonio”) **< 0**)...

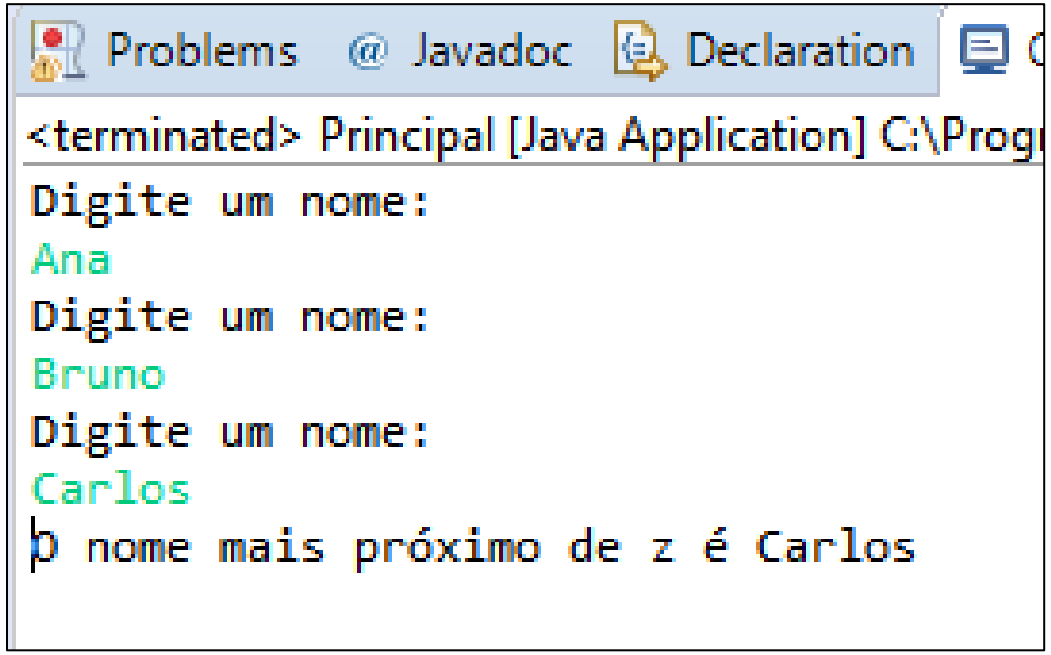
Respostas
2 – false
3 – true
4 – false

Método compareTo()

- No fim, o método compareTo faz uma **subtração** entre o código ASCII das primeiras letras de cada String comparada.
- “zé”.compareTo(“antônio“) vai comparar a letra **z** com a letra **a.**
- **z** vale **122** na tabela ASCII.
- **a** vale **97.**
- Logo, “zé”.compareTo(“antônio“) monta a operação **122 – 97.**
- O resultado é 25. Que é maior que zero. Significa que **zé** é maior que **antônio**. Por isso, está mais próximo da letra z (de acordo com a tabela ASCII).

Exercício 7

Faça um programa que receba três nomes e apresente o maior deles. O mais próximo da letra **z**.



Resposta Exercício 7

Principal.java

```java
import java.util.Scanner;

public class Principal {
    public static void main (String[] args) {
        Scanner entrada = new Scanner(System.in);
        String nome1, nome2, nome3, maior;
        System.out.println("Digite um nome:");
        nome1 = entrada.nextLine();
        System.out.println("Digite um nome:");
        nome2 = entrada.nextLine();
        System.out.println("Digite um nome:");
        nome3 = entrada.nextLine();

        if (nome1.compareTo(nome2)>0) {maior = nome1;}
        else { maior = nome2;}

        if (maior.compareTo(nome3)>0) {}
        else {maior = nome3;}

        System.out.println("O nome mais próximo de z é "+maior);
    }
}
```

Problems | Javadoc | Declaration

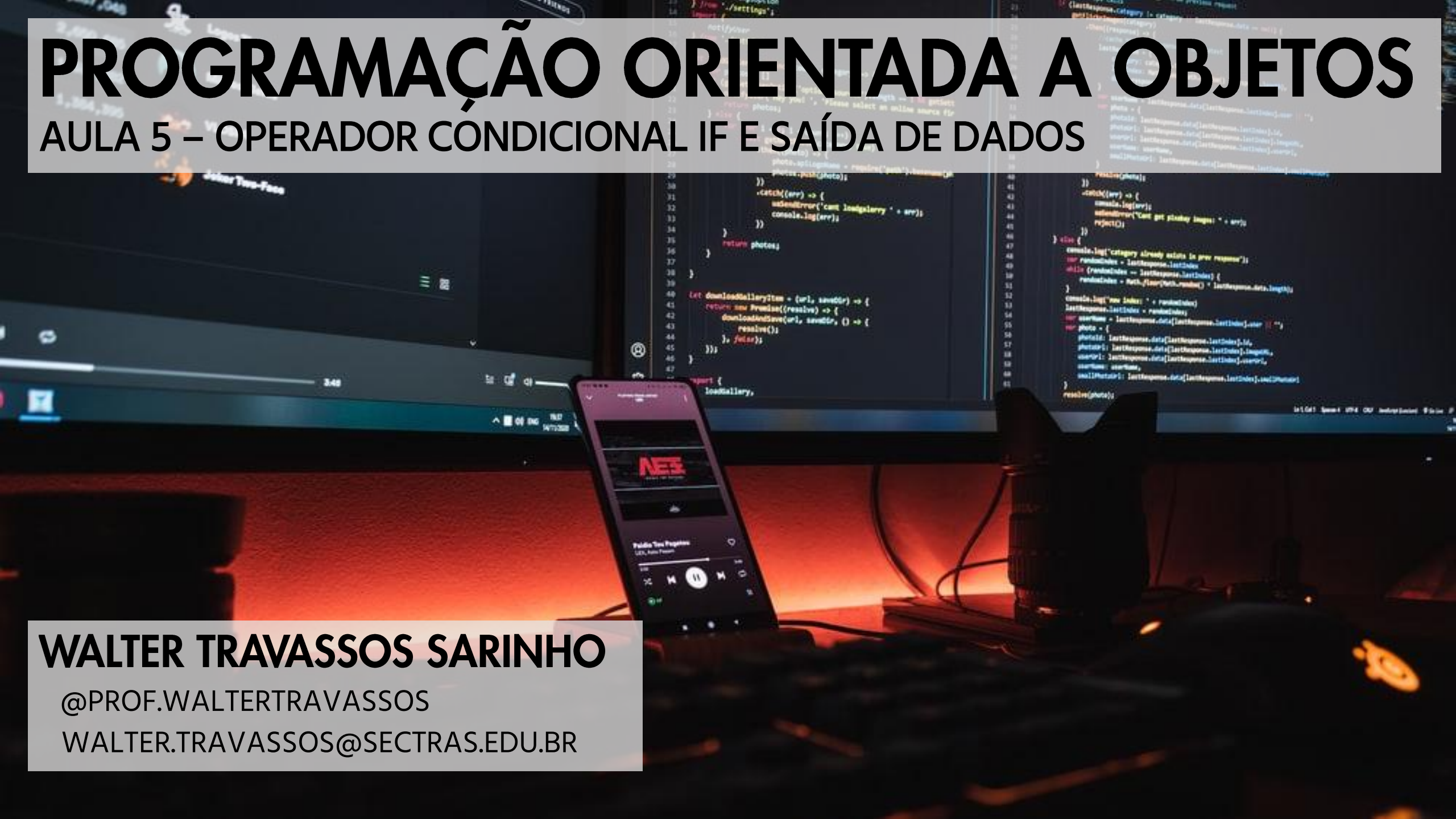
<terminated> Principal [Java Application] C:\Progr

```
Digite um nome:
Ana
Digite um nome:
Bruno
Digite um nome:
Carlos
O nome mais próximo de z é Carlos
```

Lista de Exercícios 5

PROGRAMAÇÃO ORIENTADA A OBJETOS

AULA 5 – OPERADOR CONDICIONAL IF E SAÍDA DE DADOS



WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR