

PROGRAMAÇÃO ORIENTADA A OBJETOS

AULA 4 – OPERADORES E MÉTODOS DE CONVERSÃO

WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR

Dúvidas sobre a aula passada



i forgor

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 05

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



Avaliação 1

AULA 02

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

AULA 04

Classes Wrapper. Operadores e métodos de conversão.



Mapa de Estudos

AULA 06

Manipulação de Strings.
Operador condicional
switch.



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



AULA 10
Construindo
métodos e
Métodos
construtores



AULA 09
O paradigma
Orientado a
Objetos



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



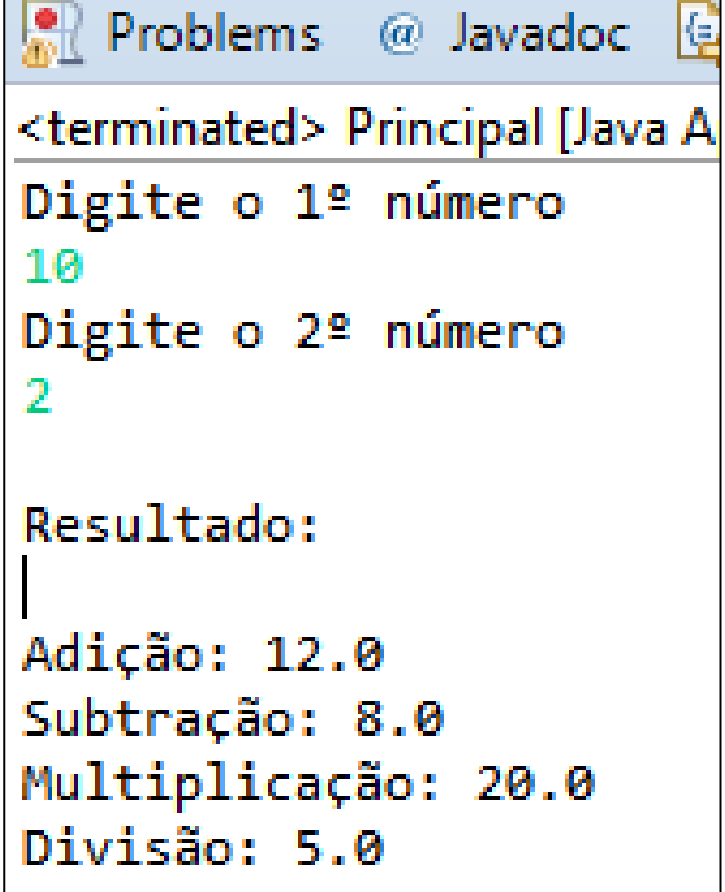
Avaliação 2

Exercício 1 - Revisão

Faça um programa que recebe dois números digitados pelo usuário e calcule a soma, subtração, multiplicação e divisão entre os números.

Exiba as respostas no final do processamento.

Tempo: 5 minutos

A screenshot of a Java IDE console window. The title bar reads "Problems @ Javadoc". The main content area shows the following text:

```
<terminated> Principal [Java A
Digite o 1º número
10
Digite o 2º número
2

Resultado:
|
Adição: 12.0
Subtração: 8.0
Multiplicação: 20.0
Divisão: 5.0
```

Resposta

```
Principal.java ✕
1 import java.util.Scanner;
2
3 public class Principal {
4     public static void main (String[] args) {
5         double a, b;
6         Scanner teclado = new Scanner (System.in);
7         System.out.println("Digite o 1º número");
8         a = teclado.nextDouble();
9         System.out.println("Digite o 2º número");
10        b = teclado.nextDouble();
11        System.out.println("\nResultado: \n");
12        System.out.println("Adição: "+(a+b));
13        System.out.println("Subtração: "+(a-b));
14        System.out.println("Multiplicação: "+(a*b));
15        System.out.println("Divisão: "+(a/b));
16    }
17 }
18
```

```
Problems @ Javadoc
<terminated> Principal [Java A
Digite o 1º número
10
Digite o 2º número
2

Resultado:
|
Adição: 12.0
Subtração: 8.0
Multiplicação: 20.0
Divisão: 5.0
```

Aula de Hoje

- (Re) Apresentar a Classe Scanner.
- Métodos de conversão.
- Compilar e executar um programa Java pelo Prompt do DOS.
- Apresentar os operadores em Java.

Entrada de Dados em Java (Utilizando o Console)

Entrada em Java (Console)

- Para a entrada de valores a partir do teclado, utilizamos métodos da classe **Scanner**.
- Sintaxe da instância:

```
Scanner nome_objeto = new Scanner(System.in);
```

- Para a classe Scanner, existem métodos de leitura específicos para cada tipo primitivo de dados.

Entrada em Java (Console)

- Para a leitura de dados inteiros:

```
Scanner entrada = new Scanner(System.in);  
int numero;  
numero = entrada.nextInt();
```

- Para a leitura de dados fracionários:

```
Scanner entrada = new Scanner(System.in);  
float numero1;  
double numero2;  
numero1 = entrada.nextFloat();  
numero2 = entrada.nextDouble();
```

Entrada em Java (Console)

- Para a leitura de dados literais:

```
Scanner entrada = new Scanner(System.in);  
String nome;  
nome = entrada.nextLine();  
nome = entrada.next();
```

- Para a leitura de dados lógicos (booleanos):

```
Scanner entrada = new Scanner(System.in);  
boolean ligou;  
ligou = entrada.nextBoolean();
```

Lembre-se...

- A classe Scanner faz parte da biblioteca de classes `java.util`.
- Devemos importar esta biblioteca em nossas classes para que elas possam instanciar um objeto da classe Scanner e, a partir deste objeto, acessar os métodos de captura de entrada de dados:

```
import java.util.Scanner;


public class nome_programa {

    Scanner entrada = new Scanner(System.in);


    ...
}
```

QUAL O ATALHO
PARA IMPORTAR
BIBLIOTECAS NO
ECLIPSE?



A digital illustration of Yoda, the green alien character from Star Wars, wearing his characteristic brown robes. He is positioned on the left side of the image, looking towards the right.

QUAL O ATALHO
PARA IMPORTAR
BIBLIOTECAS NO
ECLIPSE?

A digital illustration of Darth Vader, the iconic villain from Star Wars, wearing his black helmet and cape. He is positioned on the right side of the image, pointing his right hand towards the left.

CONTROL
+ SHIFT +
0

Qual o formato ideal para capturar a informação digitada pelo usuário?

Neste momento, usaremos String!

Em seguida faremos as conversões!

Já aprendemos a utilizar o *casting* para converter valores primitivos entre si. Agora, vamos utilizar métodos *parser* para converter String em números!

Métodos para Conversões entre Tipos de Dados (String para Número)

- É recomendável sempre receber os dados como String e, em seguida, realizar a conversão para o tipo específico.
- Use a classe referente ao tipo primitivo (ex. Byte) e o **método parse** específico para conversão:

```
String i = "1";
```

```
byte b = Byte.parseByte(i);
```

```
short s = Short.parseShort(i);
```

```
int in = Integer.parseInt(i);
```

```
long l = Long.parseLong(i);
```

```
String j = "1.8";
```

```
float f = Float.parseFloat(j);
```

```
double d = Double.parseDouble(j);
```

Exercício 2

Faça um programa que receba 3 notas de um aluno (em variáveis do tipo String, utilizando o método `nextLine()` ou `next()`) e calcule sua média.

Tempo: 5 minutos

```
1 import java.util.Scanner;
2 public class Principal {
3
4     public static void main(String[] args) {
5
6         String n1, n2, n3;
7         double nota1, nota2, nota3, media;
8         Scanner entrada = new Scanner(System.in);
9
10        System.out.println("1ª nota: ");
11        n1 = entrada.nextLine();
12        nota1 = Double.parseDouble(n1);
13
14        System.out.println("2ª nota: ");
15        n2 = entrada.nextLine();
16        nota2 = Double.parseDouble(n2);
17
18        System.out.println("3ª nota: ");
19        n3 = entrada.nextLine();
20        nota3 = Double.parseDouble(n3);
21
22        media = (nota1 + nota2 + nota3) / 3;
23        System.out.println("A média é: "+media );
24        entrada.close();
25    }
26 }
27
```

```
*Principal.java x
1 import java.util.Scanner;
2 public class Principal {
3
4     public static void main(String[] args) {
5
6         double nota1, nota2, nota3, media;
7         Scanner entrada = new Scanner(System.in);
8
9         System.out.println("1ª nota: ");
10        nota1 = Double.parseDouble(entrada.nextLine() );
11
12        System.out.println("2ª nota: ");
13        nota2 = Double.parseDouble(entrada.nextLine() );
14
15        System.out.println("3ª nota: ");
16        nota3 = Double.parseDouble(entrada.nextLine() );
17
18        media = (nota1 + nota2 + nota3) / 3;
19        System.out.println("A média é: "+media );
20        entrada.close();
21    }
22 }
23
```



Classes Wrapper

<https://www.devmedia.com.br/autoboxing-e-unboxing-em-java/28620>

Autoboxing e Unboxing

- Antes do JDK 1.5, não era fácil para converter tipos de dados primitivos (int, char, float, double) em seu objeto equivalente (classes Wrapper) – Integer, Character, Float, Double.
- O JDK 5 veio com recursos de **conversão automática de tipos de dados primitivos em seu objeto equivalente**. Esse recurso é conhecido como **autoboxing**.
- O contrário disso, é conhecido como unboxing, ou seja, o processo de **conversão de objetos em tipos primitivos de dados correspondentes** é chamado de **unboxing**.

Autoboxing



Tipo Primitivo	Classe Wrapper
boolean	Boolean
byte	Byte
char	Character
float	Float
double	Double
int	Integer
long	Long
short	Short



unboxing

Exemplos

Autoboxing – conversão de tipo primitivo para tipo objeto (classe Wrapper). Por exemplo, armazenar um número inteiro num tipo objeto.

```
Integer num1 = 9;
```

Unboxing – conversão de tipo objeto (classe Wrapper) para o tipo primitivo.

```
int num2 = 0;
```

```
num2 = new Integer(9);
```

Existem 2 desvantagens no uso de autoboxing

Desempenho e Comparação de Valores

1. Desempenho

- O uso **excessivo** do autoboxing ou unboxing **diminui o desempenho** de um aplicativo pois cria um objeto indesejado que leva o Garbage Colector a ser executado com mais frequência.
- Caso o autoboxing aconteça dentro de um loop de objetos desnecessários, pode diminuir o desempenho da aplicação.

2. Comparação de valores

- Veremos nas próximas aulas que o operador de igualdade `==` conduz a certa confusão, pois ele pode ser aplicado a ambos os tipos de dados (primitivos e de referência).
- Quando o operador é aplicado nos objetos, ele realmente **compara a referência** de um dos objetos, e não os valores.
- Assim, a comparação sempre será falsa.

Intervalo: 15 minutos

Tem alguma coisa estranha nesse código?

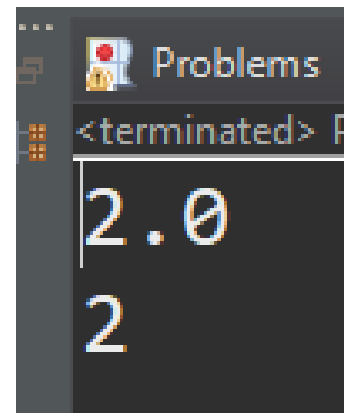
```
public static void main (String[] args) {  
    double num = 5/2;  
    System.out.println(num);  
    System.out.println(5/2);  
}
```

O que vai aparecer no console?

Tem alguma coisa estranha nesse código?

```
public static void main (String[] args) {  
    double num = 5/2;  
    System.out.println(num);  
    System.out.println(5/2);  
}
```

O que vai aparecer no console?



Podemos resolver de duas formas

1ª Resolução

```
double num = 5/2d;  
System.out.println(num);  
System.out.println(5/2);
```

2ª Resolução

```
double num = 5./2;  
System.out.println(num);  
System.out.println(5/2);
```

Em ambas soluções o resultado de num será 2.5.

O resultado inesperado acontece pois há a divisão de dois inteiros. A resposta é um int que depois é convertida para double, mas neste caso, a precisão já foi perdida logo na divisão. Então basta dizer na hora da divisão que você está trabalhando com pelo menos um valor double (com o **d** ou com o **ponto**).

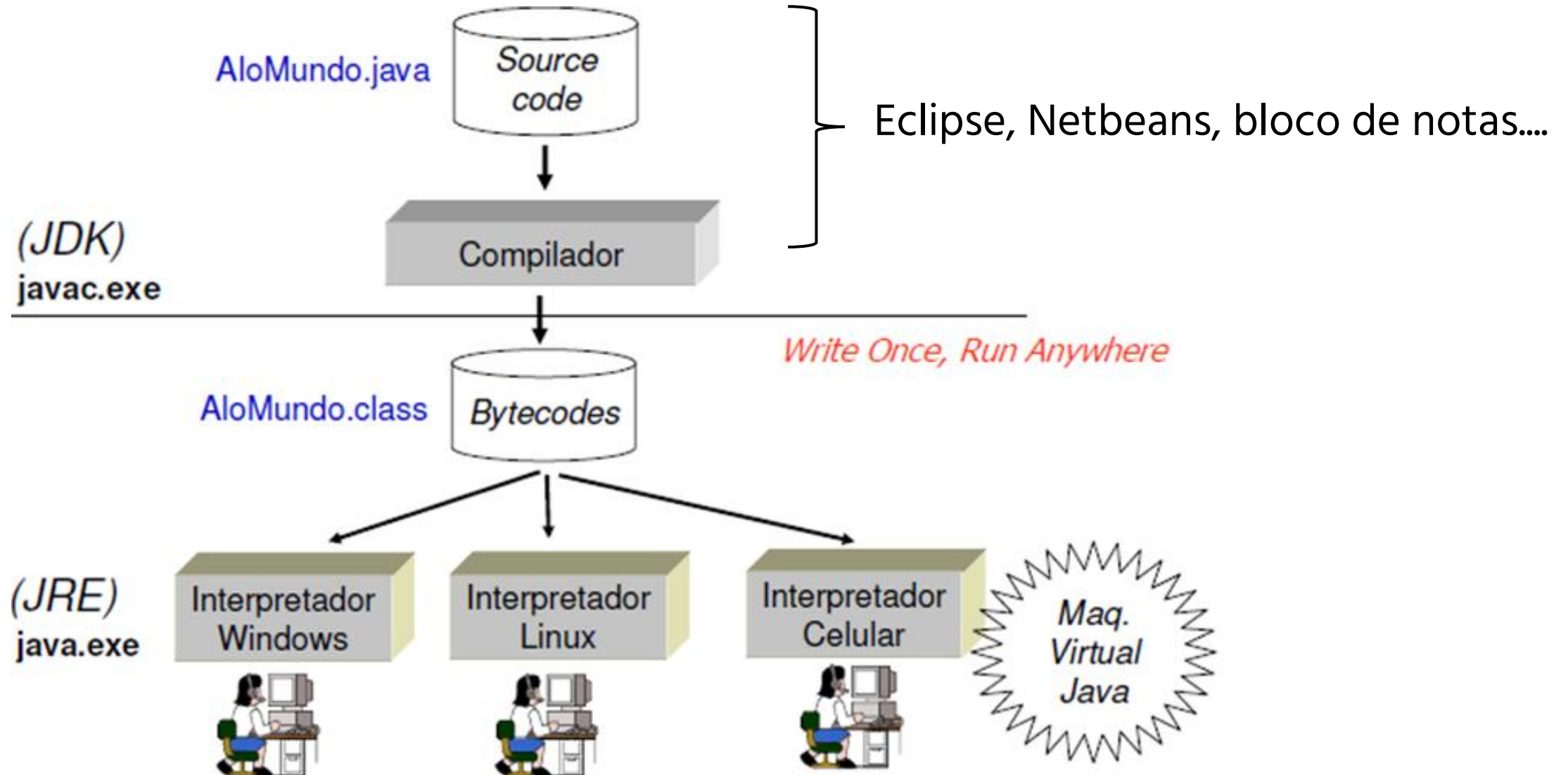
**Você sabia que não
precisa de uma IDE para
executar um programa Java?**

Escrevendo, compilando e executando aplicações Java

Arquitetura da Tecnologia Java

1. **Código-fonte:** crie um documento contendo o código fonte.
2. **Compilador:** execute o documento em um compilador. Enquanto houver erros, o programa não será compilado.
3. **Saída (bytecode):** o compilador cria um novo documento codificado em bytecode java (extensão .class). Qualquer dispositivo capaz de executar Java conseguirá interpretar o arquivo.
4. **Máquinas Virtuais:** você pode não ter uma máquina física, mas após instalar o JRE (ou JDK) seu computador instala uma máquina virtual java (implementada em software) que pode executar o bytecode criado.

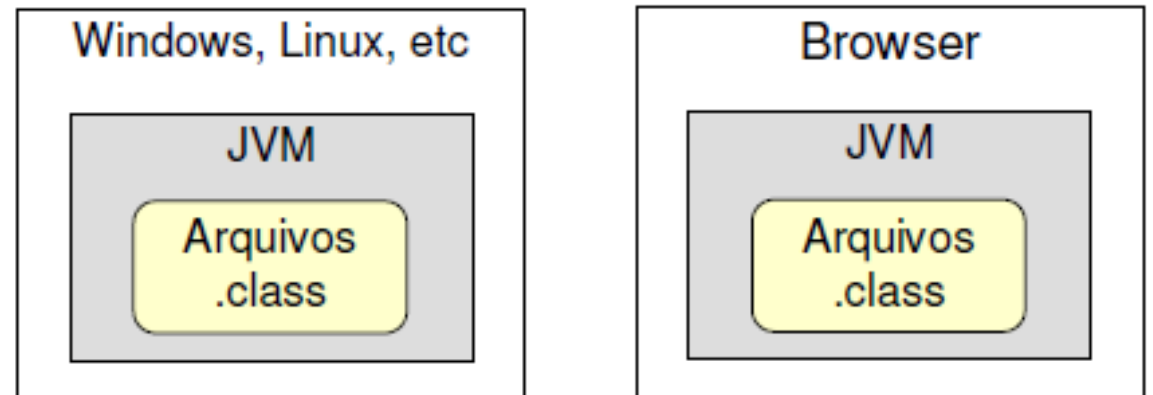
Portabilidade de um programa Java



Máquina Virtual Java

Para executar um arquivo de bytecodes (.class) o interpretador java.exe instancia na memória uma Máquina Virtual Java (JVM), que vai:

1. Carregar todas as classes necessárias (.class)
2. Verifica os bytecodes ilegais (segurança)
3. Iniciar a interpretação dos bytecodes. No S.O. inicia-se a partir do método main.



Executando o 1º programa Java sem IDE

Meu primeiro programa em Java (sem uma IDE)

Abra o bloco de notas e escreva a classe abaixo.

O nome do arquivo deve ser igual ao nome de sua classe.

```
class AloMundo {  
    public static void main (String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

SALVE EM
AloMundo.java

Compilando e Executando

```
C:\exemploJava>javac AloMundo.java  
  
C:\exemploJava>java AloMundo  
Hello World!  
  
C:\exemploJava>_
```

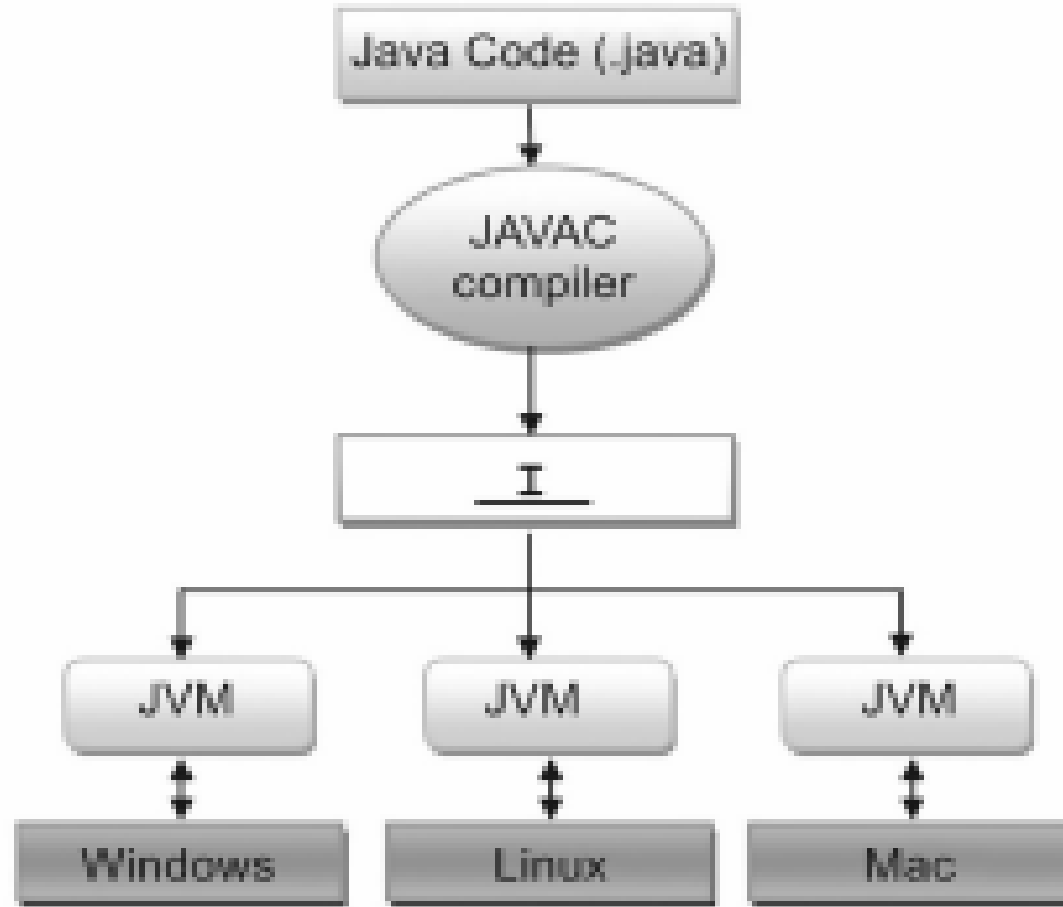
1. Abra o prompt de comando (menu iniciar + R; em seguida digite **cmd** e pressione **Ok**):
2. Vá até a pasta do arquivo **AloMundo.java**.
3. Para compilar, insira o comando: **javac AloMundo.java**
4. Caso seja compilado com sucesso, um novo arquivo vai aparecer na pasta. É o bytecode. Vai se chamar **AloMundo.class**
5. Para executar, é só inserir o comando: **java AloMundo**
6. Deve aparecer "Hello World" logo abaixo.

Caso não funcione...

- É preciso configurar a variável de ambiente JAVA_HOME!
- Assista ao vídeo do professor <https://youtu.be/8gVjUZ1BWlg>

Pergunta 1

Considere a imagem que mostra o esquema de funcionamento da Java Virtual Machine – JVM.

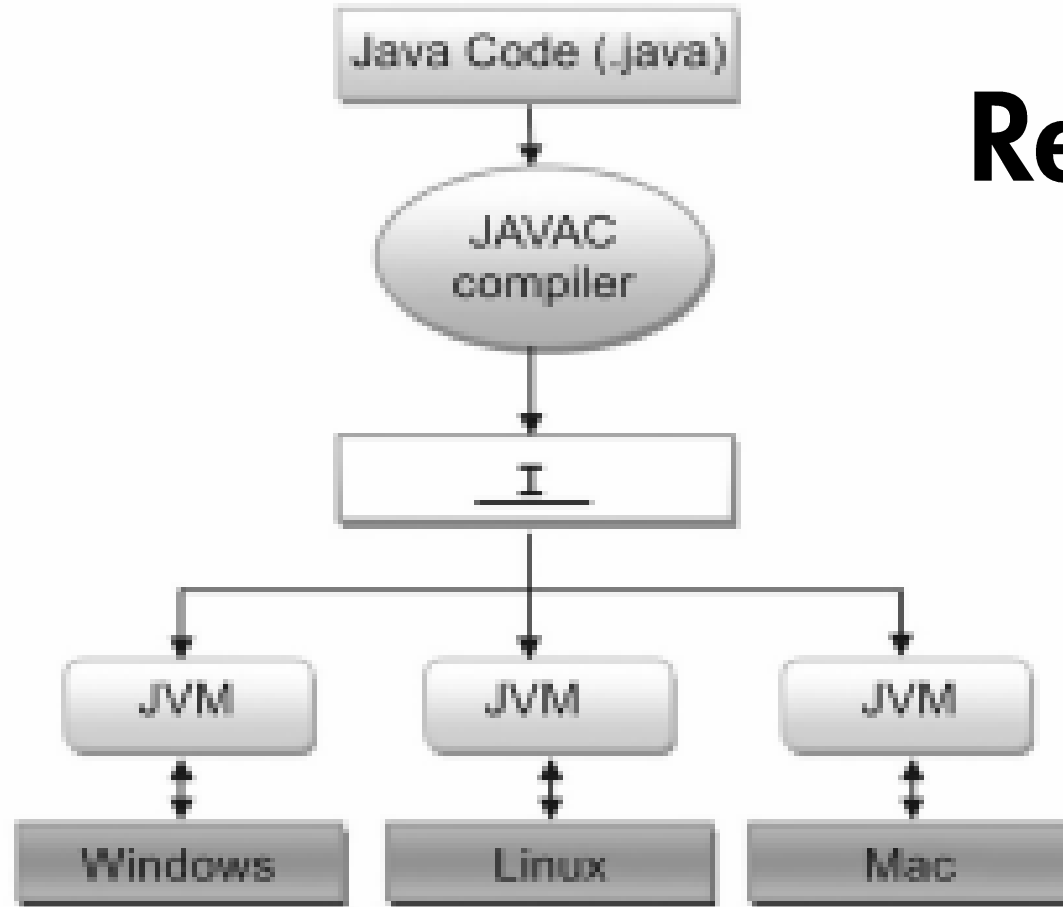


O produto resultante do processo de compilação de classes Java, que deve preencher a lacuna I, é conhecido como

- a) hashcode.
- b) bytecode.
- c) CIL (Common Intermediate).
- d) assembly.
- e) classcode.

Pergunta 1

Considere a imagem que mostra o esquema de funcionamento da Java Virtual Machine – JVM.



Resposta: B

O produto resultante do processo de compilação de classes Java, que deve preencher a lacuna I, é conhecido como

- a) hashcode.
- b) bytecode.
- c) CIL (Common Intermediate).
- d) assembly.
- e) classcode.

Pergunta 2

Para executar um programa Java deve ocorrer um processo que envolve compilação e interpretação. Quando se compila uma classe com extensão .java é gerado um arquivo com extensão...

- a) .class, conhecido como bytecode, que pode ser compilado pela JVM.
- b) .jar, conhecido como bytecode, que pode ser lido pela JVM.
- c) .class, que instala a classe na memória virtual para ser executada.
- d) .jar, que quando executado, cria um arquivo .class, que é interpretado pela JVM.
- e) .class, conhecido como bytecode, que pode ser interpretado pela JVM.

Pergunta 2

Resposta: E

Para executar um programa Java deve ocorrer um processo que envolve compilação e interpretação. Quando se compila uma classe com extensão .java é gerado um arquivo com extensão...

- a) .class, conhecido como bytecode, que pode ser compilado pela JVM.
- b) .jar, conhecido como bytecode, que pode ser lido pela JVM.
- c) .class, que instala a classe na memória virtual para ser executada.
- d) .jar, que quando executado, cria um arquivo .class, que é interpretado pela JVM.
- e) .class, conhecido como bytecode, que pode ser interpretado pela JVM.

Pergunta 3

Os comandos que compilam um arquivo java e executam o bytecode gerado no prompt de comando, respectivamente, são:

- a) java e javac
- b) java e java
- c) javac e java
- d) javac e javac
- e) java e bytecode

Pergunta 3

Resposta: C

Os comandos que compilam um arquivo java e executam o bytecode gerado no prompt de comando, respectivamente, são:

- a) java e javac
- b) java e java
- c) javac e java
- d) javac e javac
- e) java e bytecode

Operadores

Operadores

- **Aritméticos:** soma (+), subtração (-), multiplicação (*), divisão (/) e resto da divisão (%).
- **Atribuição:** = (Equivalente ao <-).
- **Concatenação:** + (Unir textos).
- **Incremento** (++) e **decremento** (--).
- **Operadores relacionais:** maior que (>), menor que (<), igual (==), diferente (!=), maior ou igual (>=), menor ou igual (<=).
- **Operadores lógicos:** E (&&), OU (||), NÃO (!).

Operadores de bit

Esses operadores são usados para realizar operações a nível de bit em variáveis inteiras. Os operadores de bits em Java são `&` (AND bit a bit), `|` (OR bit a bit), `^` (XOR bit a bit), `~` (NOT bit a bit) e `<<`, `>>` e `>>>` (deslocamento de bits para a esquerda e para a direita).

Por exemplo:

```
int a = 5, b = 3;
```

```
int resultado = a & b;
```

```
// Retorna 1 (binário: 0101 & 0011 = 0001)
```

Operador **Condiciona**l Ternário

Sintaxe: (condição) ? <Verdadeiro> : <Falso>

Calculando o maior valor entre dois números:

```
int a = 120, b = 0;
```

```
int maior = (a>b) ? a : b;
```

- **Qual valor será mostrado?**
- **R: 120**

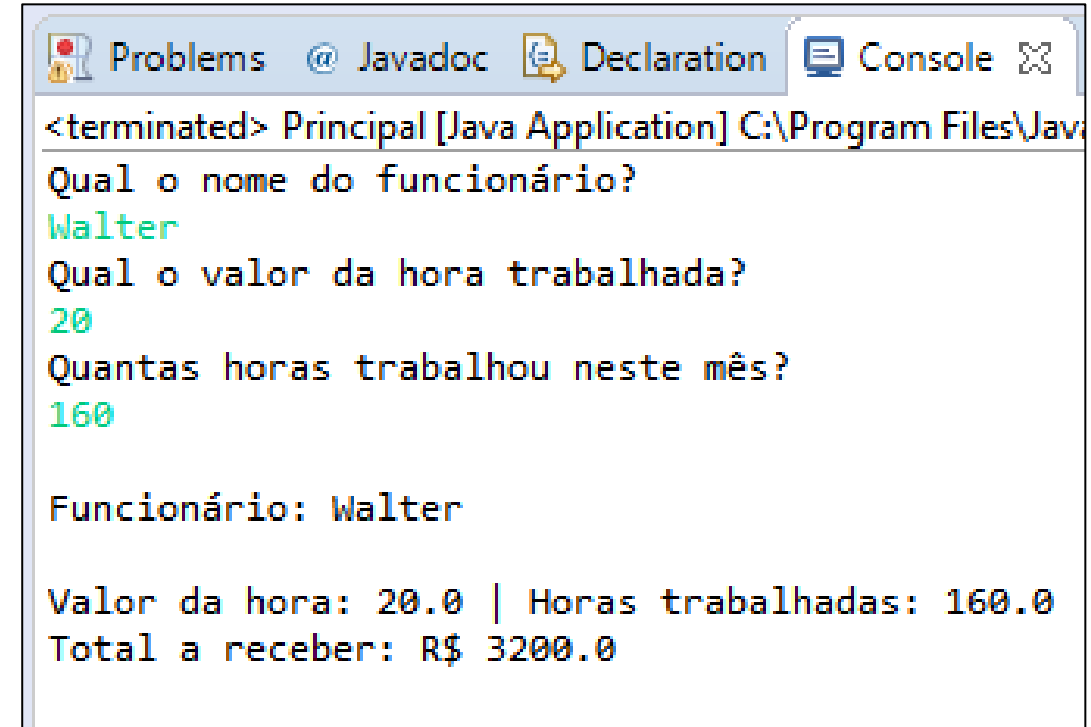
**Você leu os capítulos 1 e 2
do livro **Java! Use a cabeça?****

Mais Exercícios...

Exercício 3

Faça um programa que recebe o nome de um funcionário, o número de horas trabalhadas e o valor que recebe por hora. Com estas informações, calcular o salário deste funcionário e exibir seu nome e salário final.

OBS: Receba os valores em String e converta para double!



```
Problems @ Javadoc Declaration Console X
<terminated> Principal [Java Application] C:\Program Files\Java
Qual o nome do funcionário?
Walter
Qual o valor da hora trabalhada?
20
Quantas horas trabalhou neste mês?
160

Funcionário: Walter

Valor da hora: 20.0 | Horas trabalhadas: 160.0
Total a receber: R$ 3200.0
```

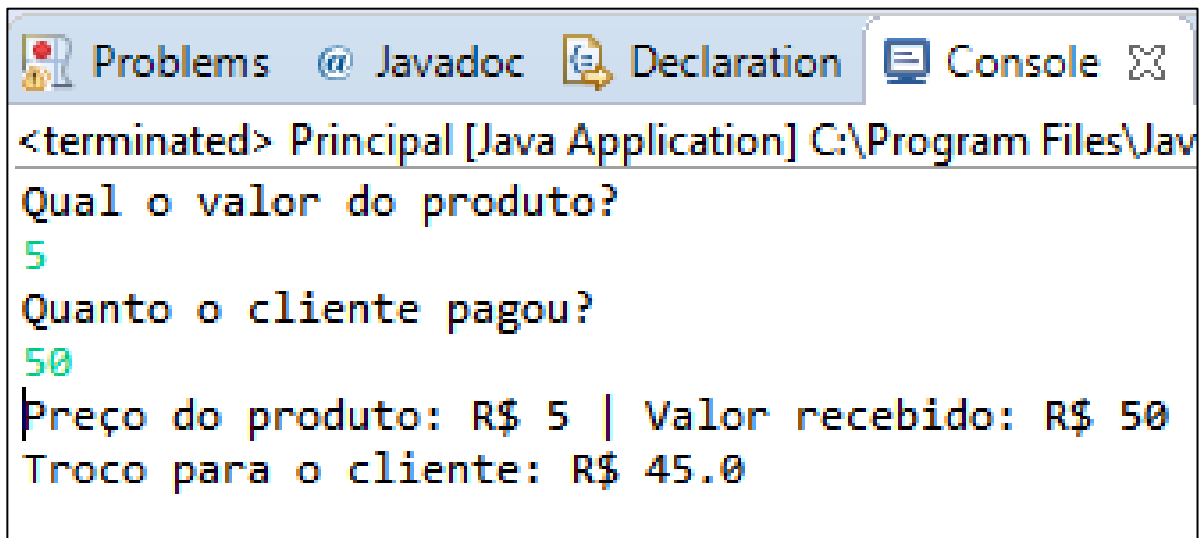
Resposta Exercício 3

```
Principal.java ✖
1 import java.util.Scanner;
2
3 public class Principal {
4     public static void main (String[] args) {
5         String nome;
6         Double qtdhoras, valorhora;
7         Scanner teclado = new Scanner (System.in);
8         System.out.println("Qual o nome do funcionário?");
9         nome = teclado.next();
10        System.out.println("Qual o valor da hora trabalhada?");
11        valorhora = Double.parseDouble(teclado.next());
12        System.out.println("Quantas horas trabalhou neste mês?");
13        qtdhoras = Double.parseDouble(teclado.next());
14        System.out.println("\nFuncionário: "+nome+"\n");
15        System.out.print("Valor da hora: "+valorhora+" | Horas trabalhadas: "+ qtdhoras);
16        System.out.println("\nTotal a receber: R$ "+ (valorhora*qtdhoras));
17    }
18 }
19
```

Exercício 4

Faça um programa que receba o **preço de um produto**, o **valor pago** pelo cliente e **retorne o troco** a ser dado.

OBS: Receba o valor em String e converta para os tipos numéricos



```
Problems @ Javadoc Declaration Console X
<terminated> Principal [Java Application] C:\Program Files\Jav
Qual o valor do produto?
5
Quanto o cliente pagou?
50
Preço do produto: R$ 5 | Valor recebido: R$ 50
Troco para o cliente: R$ 45.0
```

Resposta Exercício 4

Principal.java

```
1 import java.util.Scanner;
2
3 public class Principal {
4     public static void main (String[] args) {
5         String preçoproduto, valorpago;
6         Scanner teclado = new Scanner (System.in);
7         System.out.println("Qual o valor do produto?");
8         preçoproduto = teclado.next();
9         System.out.println("Quanto o cliente pagou?");
10        valorpago = teclado.next();
11        System.out.print("Preço do produto: R$ "+preçoproduto+" | Valor recebido: R$ "+ valorpago);
12        System.out.println("\nTroco para o cliente: R$ "+ (Double.parseDouble(valorpago)-Double.parseDouble(preçoproduto)));
13    }
14 }
15
```

Problems @ Javadoc Declaration Console

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

Qual o valor do produto?

5

Quanto o cliente pagou?

50

Preço do produto: R\$ 5 | Valor recebido: R\$ 50

Troco para o cliente: R\$ 45.0

Exercício 5

$$\text{área do trapézio} = \frac{(\text{base maior} + \text{base menor})}{2} * \text{altura}$$

Faça um programa que recebe a **base maior**, a **base menor** e a **altura de um trapézio**. A partir destes dados, o programa deverá **calcular a área do trapézio**.

OBS: Receba os dados em String e converta para o tipo numérico adequado.

```
Problems @ Javadoc Decla
<terminated> Principal [Java Applicat
Base maior?
10
Base Menor?
8
Altura?
5
Área do Trapézio = 45.0
```

Resposta Exercício 5

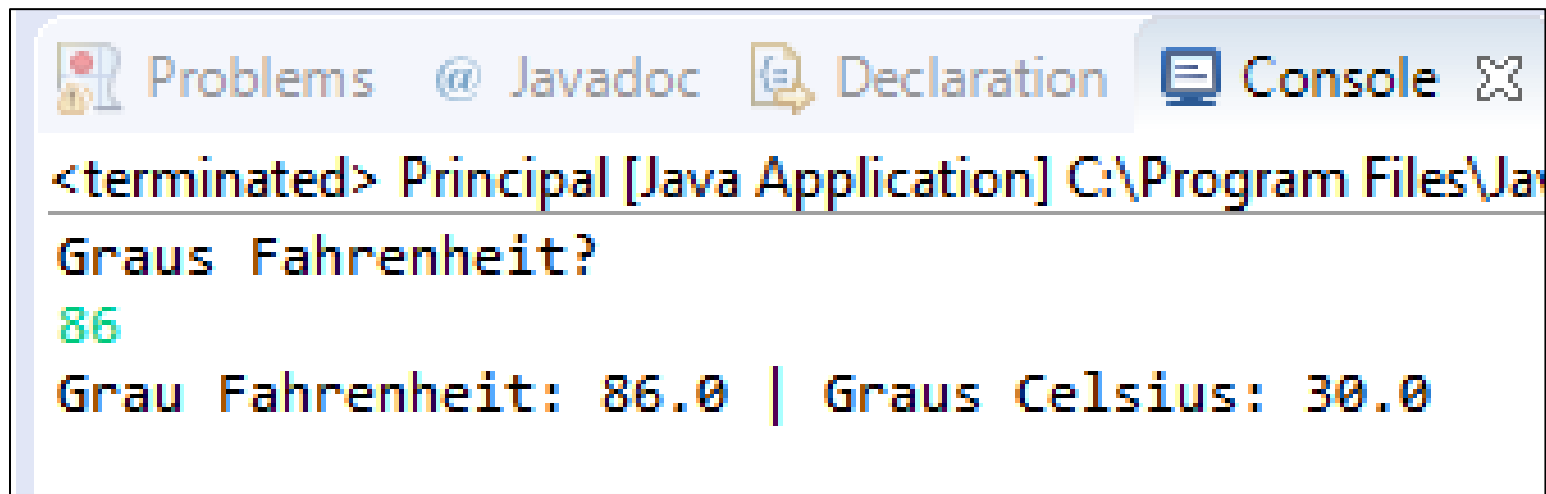
```
Principal.java ✕
1 import java.util.Scanner;
2
3 public class Principal {
4     public static void main (String[] args) {
5         String bmaior, bmenor, altura;
6         Scanner teclado = new Scanner (System.in);
7         System.out.println("Base maior?");
8         bmaior = teclado.next();
9         System.out.println("Base Menor?");
10        bmenor = teclado.next();
11        System.out.println("Altura?");
12        altura = teclado.next();
13        double bme, bma, alt;
14        bma = Double.parseDouble(bmaior);
15        bme = Double.parseDouble(bmenor);
16        alt = Double.parseDouble(altura);
17        System.out.println("Área do Trapézio = "+ ( ( (bma+bme)/2)*alt ));
18    }
19 }
```

```
Problems @ Javadoc Decla
<terminated> Principal [Java Applicat
Base maior?
10
Base Menor?
8
Altura?
5
Área do Trapézio = 45.0
```

Exercício 6

Escreva um programa que **recebe uma temperatura em graus Fahrenheit, calcula e escreve** o valor correspondente em graus Celsius. Fórmula: $C = ((F-32) / 9) * 5$

OBS: Receba os dados em String e converta para o tipo numérico adequado.



```
<terminated> Principal [Java Application] C:\Program Files\Ja
Graus Fahrenheit?
86
Grau Fahrenheit: 86.0 | Graus Celsius: 30.0
```

Resposta Exercício 6

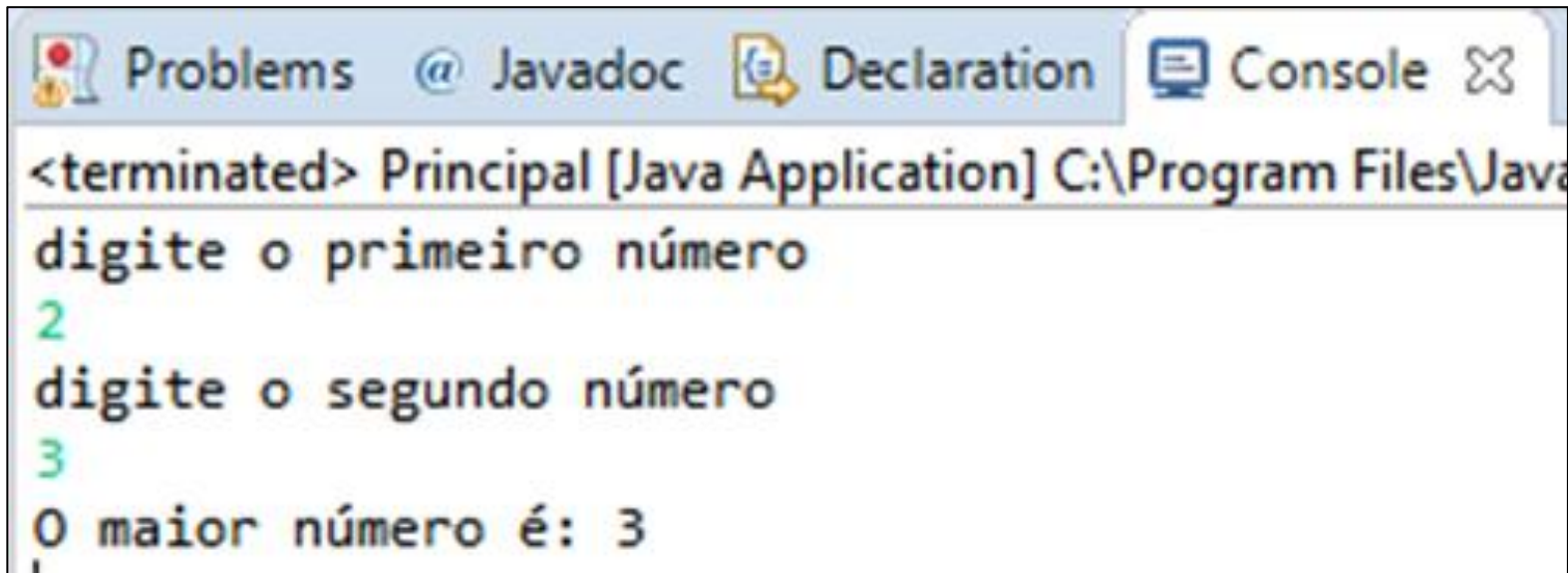
```
*Principal.java ✖
1 import java.util.Scanner;
2
3 public class Principal {
4     public static void main (String[] args) {
5         Scanner teclado = new Scanner (System.in);
6         System.out.println("Graus Fahrenheit?");
7         double grausfahrenheit = Double.parseDouble(teclado.next());
8         double grauscelsius = ((grausfahrenheit-32)/9 ) *5;
9         System.out.println("Grau Fahrenheit: "+ grausfahrenheit +
10             " | Graus Celsius: "+grauscelsius);
11     }
12 }
13
```

Problems @ Javadoc Declaration Console ✖

```
<terminated> Principal [Java Application] C:\Program Files\Ja
Graus Fahrenheit?
86
Grau Fahrenheit: 86.0 | Graus Celsius: 30.0
```

Exercício 7 – Tempo: 5 minutos

Utilizando o operador ternário, faça um programa que receba 2 números e apresente o maior.



```
<terminated> Principal [Java Application] C:\Program Files\Java
digite o primeiro número
2
digite o segundo número
3
O maior número é: 3
```

Exercício 7 - Resposta

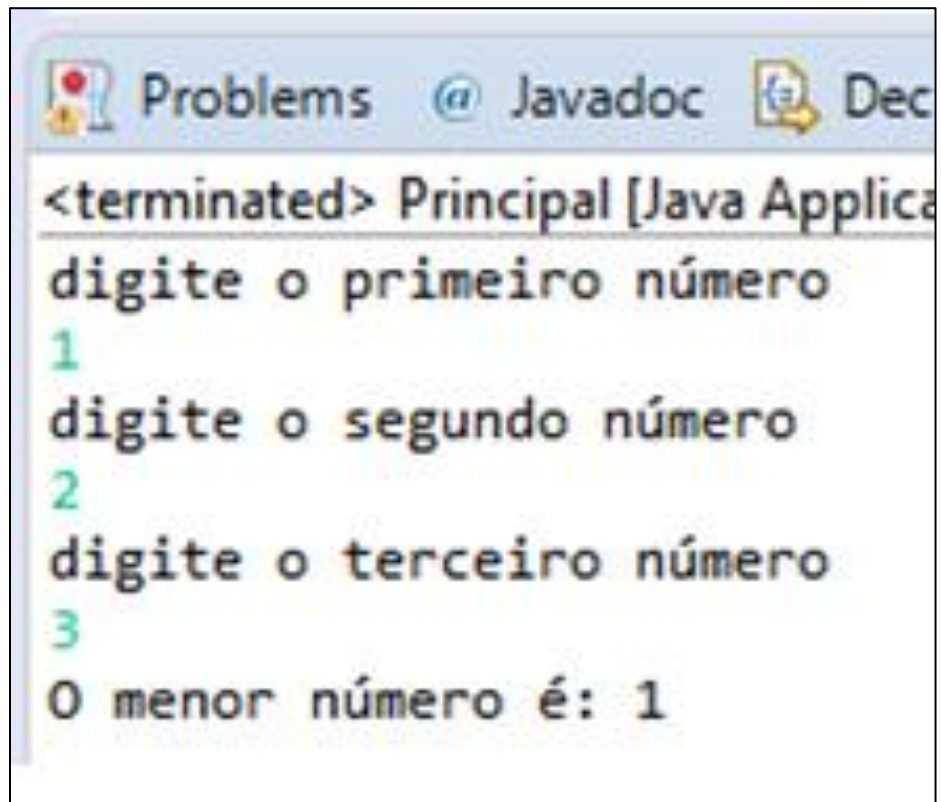
```
*Principal.java ✖
2
3 public class Principal {
4     public static void main (String[] args) {
5         Scanner teclado = new Scanner (System.in);
6         System.out.println("digite o primeiro número");
7         int num1 = teclado.nextInt();
8         System.out.println("digite o segundo número");
9         int num2 = teclado.nextInt();
10        System.out.println("O maior número é: "+(num1>num2?num1:num2));
11    }
12 }
13
```

Problems @ Javadoc Declaration Console ✖

```
<terminated> Principal [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (8 de fev de
digite o primeiro número
2
digite o segundo número
3
O maior número é: 3
```

Exercício 8

Modifique o exercício anterior para que receba três números e mostre o menor.



```
Problems @ Javadoc Dec  
<terminated> Principal [Java Applica  
digite o primeiro número  
1  
digite o segundo número  
2  
digite o terceiro número  
3  
O menor número é: 1
```

Exercício 8 - Resposta

```
3 public class Principal {
4     public static void main (String[] args) {
5         Scanner teclado = new Scanner (System.in);
6         System.out.println("digite o primeiro número");
7         int num1 = teclado.nextInt();
8         System.out.println("digite o segundo número");
9         int num2 = teclado.nextInt();
10        System.out.println("digite o terceiro número");
11        int num3 = teclado.nextInt();
12        int auxiliar = (num1<num2?num1:num2);
13        int menor = (auxiliar<num3)?auxiliar:num3;
14        System.out.println("O menor número é: "+menor);
15    }
16 }
```

```
Problems @ Javadoc Dec
<terminated> Principal [Java Applica
digite o primeiro número
1
digite o segundo número
2
digite o terceiro número
3
O menor número é: 1
```

Lista de Exercícios 4

PROGRAMAÇÃO ORIENTADA A OBJETOS

AULA 4 – OPERADORES E MÉTODOS DE CONVERSÃO

WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR