

PROGRAMAÇÃO ORIENTADA A OBJETOS

AULA 2 – SINTAXE DA LINGUAGEM E VARIÁVEIS

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

Mapa de Estudos



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



AULA 12
Classes Abstratas e
Interfaces



Avaliação 3

Método de **saída** de dados `System.out.println();`

Você pode exibir o conteúdo de uma variável String (atente para o S maiúsculo de String):

```
System.out.println("Hello World");  
String teste = "Hello World";  
System.out.println(teste);
```

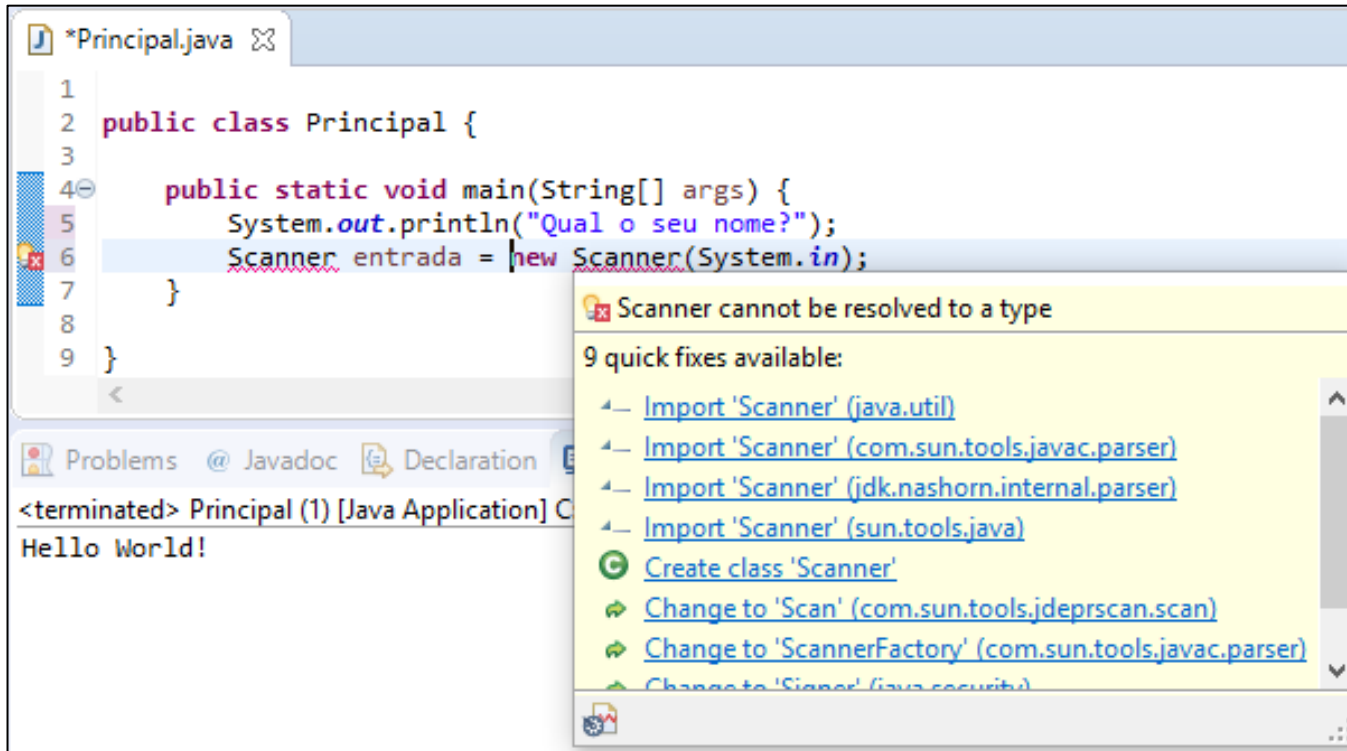
Recebendo dados do usuário

É preciso instanciar (criar) um objeto da classe Scanner:

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

Será preciso importar a biblioteca Scanner do pacote `java.util` para seu programa para utilizar esta classe.

Importando classes



```
1
2 public class Principal {
3
4     public static void main(String[] args) {
5         System.out.println("Qual o seu nome?");
6         Scanner entrada = new Scanner(System.in);
7     }
8
9 }
```

Scanner cannot be resolved to a type

9 quick fixes available:

- Import 'Scanner' (java.util)
- Import 'Scanner' (com.sun.tools.javac.parser)
- Import 'Scanner' (jdk.nashorn.internal.parser)
- Import 'Scanner' (sun.tools.java)
- Create class 'Scanner'
- Change to 'Scan' (com.sun.tools.jdeprscan.scan)
- Change to 'ScannerFactory' (com.sun.tools.javac.parser)
- Change to 'Singer' (java.recutils)

Problems @ Javadoc Declaration

<terminated> Principal (1) [Java Application] C

Hello World!

- Você pode importar classes necessárias ao programa utilizando o atalho **control + shift + o**.
- Ou você pode posicionar o mouse em cima do nome sublinhado em vermelho e aguardar a lista de opções para solucionar o problema.
- Escolha a sugestão de solução **import 'Scanner' (java.util)**.

Resultado da ação de importar

```
*Principal.java ✕  
1  import java.util.Scanner; ←  
2  
3  public class Principal {  
4  
5  public static void main(String[] args) {  
6      System.out.println("Qual o seu nome?");  
7      Scanner entrada = new Scanner(System.in);  
8  }  
9  
10 }  
11
```

Entrada de Dados Utilizando um objeto da classe Scanner

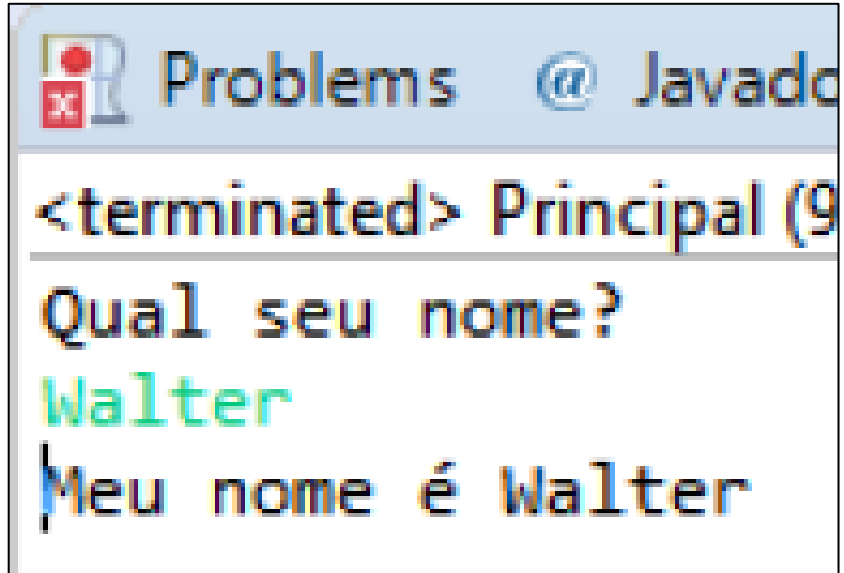
- Para ler dados do tipo String, usamos o método `nextLine()` de um objeto da classe Scanner.
- É possível também utilizar apenas o método `next()` para realizar ação de leitura de uma String (**mas ele captura até a existência de um espaço na String**).
- Para ler dados de outros tipos, usaremos outros métodos da classe. Para ler um valor inteiro, usamos o método `nextInt()`.

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

```
String nome = entrada.nextLine();
```

Exercício 1

Faça um programa que pergunte o nome e em seguida mostre as informações para o usuário.



```
Problems @ Javado
<terminated> Principal (9
Qual seu nome?
Walter
Meu nome é Walter
```

Resposta Exercício 1

```
Principal.java ✕
1 import java.util.Scanner;
2
3 public class Principal {
4
5     public static void main(String[] args) {
6         System.out.println("Qual seu nome?");
7         Scanner entrada = new Scanner(System.in);
8         String nome = entrada.next();
9         System.out.println("Meu nome é " + nome);
10    }
11 }
12
```

Problems @ Javadoc Declaration Console ✕ Git Repositories Error Log

<terminated> Principal (9) [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (31 de jan de 2017 12:10:50)

```
Qual seu nome?
Walter
Meu nome é Walter
```

Exercício 2

Faça um programa que recebe o ano de nascimento e retorne a idade do usuário.

```
<terminated> Principal (9) [Java Appli  
Qual seu ano de nascimento?  
1981  
Sua idade é 36
```

Resposta Exercício 2

```
Principal.java ✕
1 import java.util.Scanner;
2
3 public class Principal {
4
5     public static void main(String[] args) {
6         System.out.println("Qual seu ano de nascimento?");
7         Scanner entrada = new Scanner(System.in);
8         int ano = entrada.nextInt();
9         System.out.println("Sua idade é " + (2017-ano));
10    }
11 }
12
```

Problems @ Javadoc Declaration Console ✕ Git Repositories Error Log

<terminated> Principal (9) [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (31 de jan de 2017 12:17:04)

Qual seu ano de nascimento?
1981
Sua idade é 36

**Observe que no exercício 2
o programa retorna a idade
correta se estivermos em 2017.
Como resolver esse problema?**

Bibliotecas para Manipulação de Dados

<https://www.devmedia.com.br/java-8-e-sua-nova-api-para-datas/31462>

1. Classe Date

- Por muitos anos, na linguagem Java tivemos dificuldades com a manipulação de datas e horas.
- Como primeiro recurso disponível, tivemos a classe `java.util.Date`, mas rapidamente passou a ser considerada complexa e bastante limitada.

2. Classe Calendar

- Em 1998, a IBM desenvolveu e disponibilizou a classe `java.util.Calendar`, com recursos como internacionalização, métodos para alterar individualmente o dia, o mês ou o ano de uma data, etc.
- A classe `Calendar`, embora tenha trazido maior flexibilidade à manipulação de datas, ainda era considerada complexa por grande parte dos programadores

O problema dessas classes é que...

- Os anos iniciam a partir de 1900 e os meses a partir de zero.
- Por exemplo, o mês de Janeiro tem valor correspondente à zero (0), o mês de Fevereiro a (1), e assim segue até dezembro que tem valor 11.
- Os métodos da classe **Calendar** não são considerados intuitivos, já que os nomes dados a eles muitas vezes não deixa clara as suas funções.
- A classe **Date** já possui grande parte de seus métodos marcados como *deprecated* (descontinuado) e a inserção de novos métodos poderia deixar a classe mais confusa.

3. (Foi ai que surgiu o) Projeto Joda

- Como solução para Date e Calendar, em 2005 o projeto Joda Time foi lançado.
- Ele consiste de uma biblioteca externa ao JDK/JRE, criada por Stephen Colebourne, que rapidamente foi adotada por muitos desenvolvedores como a principal API para datas no Java.

LocalDate
LocalDateTime
LocalTime

```
Principal.java ✕
1 import java.time.LocalDate;
2 import java.time.LocalDateTime;
3 import java.time.LocalTime;
4
5 public class Principal {
6
7     public static void main(String[] args) {
8
9         LocalDate dataAtual = LocalDate.now();
10        System.out.println("Data: " + dataAtual);
11
12        LocalTime horaAtual = LocalTime.now();
13        System.out.println("Hora: " + horaAtual);
14
15        LocalDateTime dataHoraAtual = LocalDateTime.now();
16        System.out.println("Data e Hora: " + dataHoraAtual);
17    }
18
19 }
```

Problems @ Javadoc Declaration Console ✕

<terminated> Principal (4) [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\java
Data: 2020-08-24
Hora: 18:59:46.037
Data e Hora: 2020-08-24T18:59:46.037

Exercício 3

Modifique o Exercício 2 para que utilize a classe `Calendar` (ou `LocalDate`) para recuperar o ano atual.

```
Qual seu ano de nascimento?
```

```
1981
```

```
Sua idade é 36
```

Resposta Exercício 3 – classe Calendar

```
Principal.java
```

```
1 import java.util.Calendar;
2 import java.util.Scanner;
3
4 public class Principal {
5
6     public static void main(String[] args) {
7         System.out.println("Qual seu ano de nascimento?");
8         Scanner entrada = new Scanner(System.in);
9         int nascimento = entrada.nextInt();
10
11         Calendar c1 = Calendar.getInstance();
12         int ano = (c1.get(Calendar.YEAR));
13         System.out.println("Sua idade é " + (ano - nascimento));
14     }
15 }
16
```

Problems @ Javadoc Declaration Console Git Repositories Error Log

```
<terminated> Principal (9) [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (31 de jan de 2017 12:35:00)
Qual seu ano de nascimento?
1981
Sua idade é 36
```

Resposta Exercício 3 – classe LocalDate

```
2 import java.time.LocalDate;
3 import java.util.Scanner;
4 public class Principal {
5     public static void main(String[] args) {
6         System.out.println("Ano do nascimento?");
7         Scanner entrada = new Scanner(System.in);
8         int ano = entrada.nextInt();
9         LocalDate l1 = LocalDate.now();
10        int anoatual = l1.getYear();
11        int idade = (anoatual - ano);
12        System.out.println("a idade é: " + idade);
13    }
14 }
```

Problems Javadoc Declaration Console x

<terminated> Principal (4) [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (2 de mar. de 2023 09:02:28 – 09:02:31)

Ano do nascimento?

1981

a idade é: 42

Recapitulando...

PARA QUE
SERVE O
MÉTODO MAIN?

QUAL MÉTODO
FAZ A SAÍDA DE
DADOS NO
CONSOLE?

COMO FAZER A
LEITURA DE UM
DADO
DIGITADO PELO
USUÁRIO?

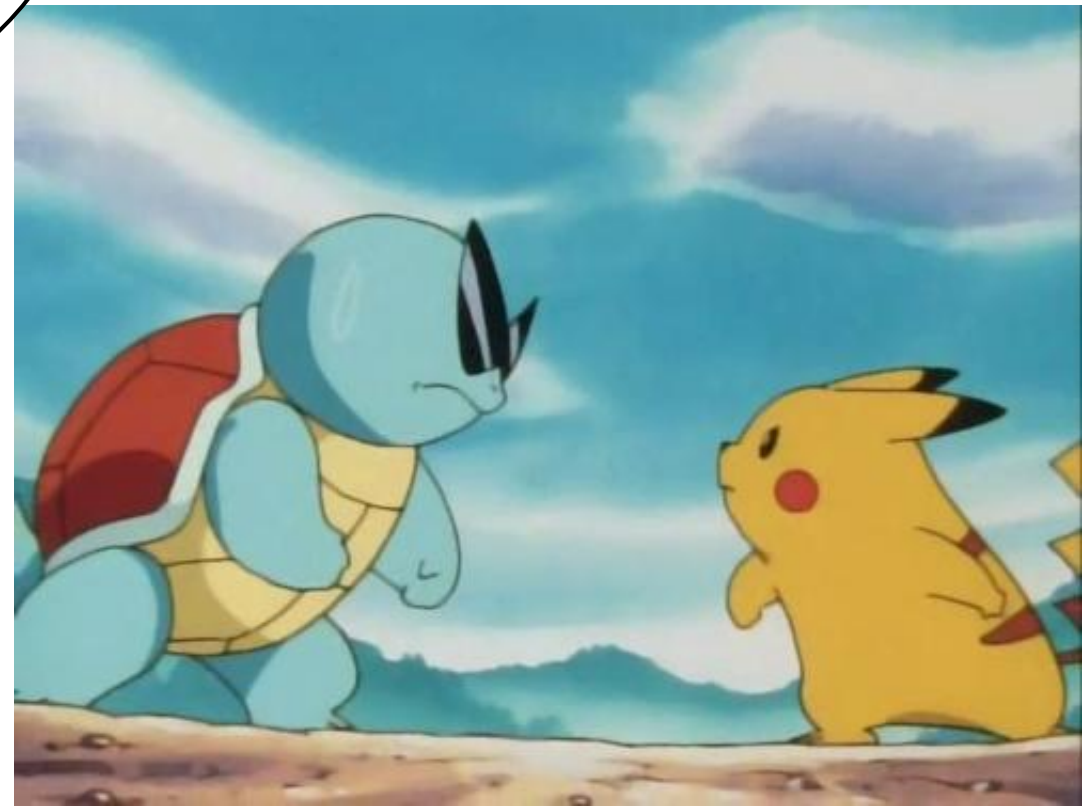
COMO
CRIAMOS UMA
CLASSE?

VOCÊ LEMBRA
COMO CRIAR UM
PROJETO NO
ECLIPSE?



NUMA LUTA ENTRE
POKÉMONS DO
TIPO ÁGUA E
ELÉTRICO, QUEM
TEM VANTAGEM?

O TIPO
ELÉTRICO
TEM
VANTAGEM!



Exercício de Revisão

Crie um novo Projeto Java e na classe Principal faça um programa que pergunte o seu nome e armazene o valor digitado pelo usuário na variável fulano. Em seguida apresente no console a seguinte mensagem:

Boa noite fulano.

Você já é um mestre pokemon?

```
Problems @ Javadoc Declaratio
<terminated> Principal (1) [Java Applicatio
Qual o seu nome?
Walter
Boa noite Walter.
Você já é um mestre pokémon?
```



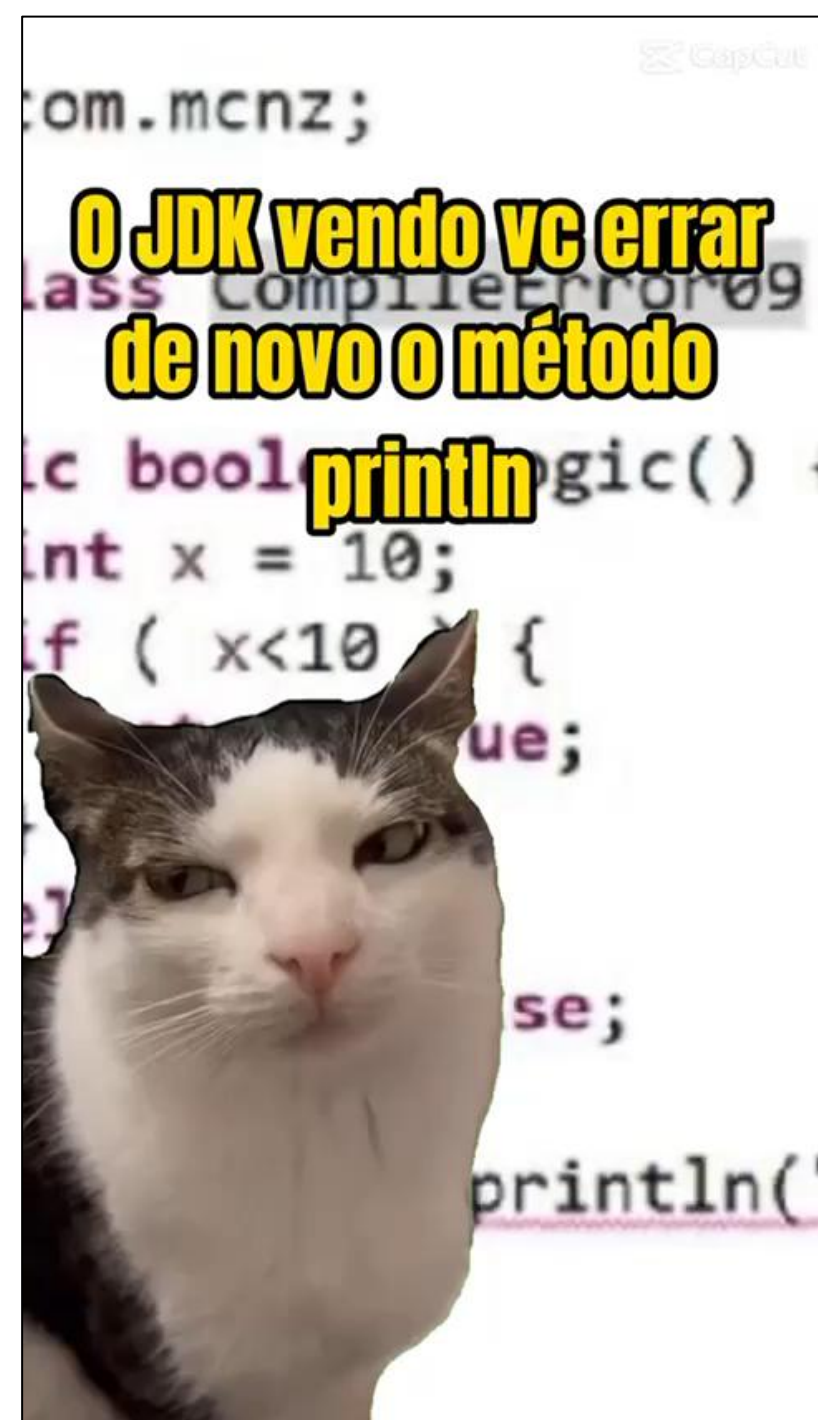
Exercício de Revisão

Resposta

```
Principal.java ✖
1 import java.util.Scanner;
2
3 public class Principal {
4
5     public static void main(String[] args) {
6         System.out.println("Qual o seu nome?");
7         Scanner teclado = new Scanner(System.in);
8         String fulano = teclado.nextLine();
9         System.out.println("Boa noite "+fulano+ ". \nVocê já é um mestre pokémon?");
10        teclado.close();
11    }
12 }
<
```

Problems @ Javadoc Declaration Console ✖

```
<terminated> Principal (1) [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (11 de fev de 2019 17:26:31)
Qual o seu nome?
Walter
Boa noite Walter.
Você já é um mestre pokémon?
```



Tente executar no eclipse esse programa

```
Principal.java ✖
4
5 public static void main(String[] args) {
6     Scanner teclado = new Scanner (System.in);
7     System.out.println("digite um numero:");
8     int n1 = teclado.nextInt();
9     System.out.println("digite agora um nome: ");
10    String nome = teclado.nextLine();
11    System.out.println("O número digitado foi "+n1+" e nome foi "+nome);
12    teclado.close();
13 }
14 }
15
```

```
Problems @ Javadoc Declaration Console ✖
<terminated> Principal (1) [Java Application] C:\Program Files\Java
digite um numero:
3
digite agora um nome:
O número digitado foi 3 e nome foi
```



ONDE ESTÁ O
ERRO DO
PROGRAMA?



AO TENTAR REALIZAR A
LEITURA DE DADOS USANDO
O OBJETO DA CLASSE
SCANNER JÁ EXISTIA UM
"BUFFER" ARMAZENADO
NESSE OBJETO.
ISSO NÃO ME DEIXOU
REALIZAR A INSERÇÃO DE
UM DADO NA VARIÁVEL.



PASSOU A IMPRESSÃO
QUE FOI DIGITADO A
TECLA ENTER SEM EU
TER FEITO NADA!

Classe Scanner

- Ao utilizar o mesmo objeto da classe Scanner para ler diversas entradas do usuário pode acontecer um erro quando os tipos de dados são diferentes (o 1º é int, o 2º é String, por exemplo).
- Para resolver este problema é preciso “limpar o buffer” do objeto da classe Scanner após a leitura de variáveis que não são do tipo String.
- Após a leitura de uma variável (inteira por exemplo, utilizando o método `teclado.nextInt();`), adicione em seguida o comando `teclado.nextLine()`.
- Isso vai funcionar como um “limpador do buffer do objeto”.

Resolvendo o problema

```
Principal.java ✖
4
5 public static void main(String[] args) {
6     Scanner teclado = new Scanner (System.in);
7     System.out.println("digite um numero:");
8     int n1 = teclado.nextInt();
9     System.out.println("digite agora um nome: ");
10    teclado.nextLine();
11    String nome = teclado.nextLine();
12    System.out.println("O número digitado foi "+n1+" e nome foi "+nome);
13    teclado.close();
14 }
15 }
```

Problems @ Javadoc Declaration Console ✖

<terminated> Principal (1) [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (11 de fev de 2019)

```
digite um numero:
3
digite agora um nome:
Walter
O número digitado foi 3 e nome foi Walter
```

Intervalo: 10 minutos

Vamos começar...

- Apresentar algumas observações de sintaxe da linguagem.
- Apresentar os tipos de variáveis em Java.
- Discutir sobre os tipos primitivos e de referência.

Observações da Sintaxe Java

1. Escrita e Escopo de Código

- Java É CASE SENSITIVE, ou seja, faz diferença entre letras maiúsculas e minúsculas.
- Cada instrução em Java deverá ser finalizada com ponto e vírgula (;).
- O corpo de código de classes, estruturas e métodos devem ser iniciados com abertura de chave ({) e encerrados com o fechamento de uma chave (}).

2. Onde começamos?

- Em Java tudo é inserido numa classe.
- O método `main()` é onde seu programa começará a ser executado.
- O que você pode inserir no método `main`? Tudo que você já fazia em outras linguagens de programação!



AGORA QUE JÁ SABEMOS
COMO COMEÇAR UM
PROJETO JAVA,
É HORA DE CONHECER OS
TIPOS DE DADOS (OU
VARIÁVEIS) SUPORTADOS
PELA LINGUAGEM!

Tipos de Dados

Variáveis

A teal ceramic mug filled with coffee sits on a burlap surface scattered with coffee beans. A white rectangular label with the word "Variáveis" is placed on the surface of the coffee in the mug.

Variáveis

Em Java, podemos declarar as variáveis em qualquer região dentro do escopo da classe. Podemos colocar também dentro dos métodos, mas a variável só existirá dentro deste método (funciona como uma variável local).

Sintaxe:

```
tipo_primitivo identificador;
```

```
int n1;
```

ou

```
tipo_primitivo identificador = valor_inicial;
```

```
int n1 = 3;
```

Variáveis x Tipos de Dados

- Além de um nome (o identificador), as variáveis precisam ter um tipo.
- Uma variável é um “recipiente” na memória que guarda uma dada quantidade de informação. Pense numa xícara, um copo e uma garrafa pet. Cada recipiente armazena uma quantidade (e um tipo) diferente de informação – é a mesma coisa com os tipos de dados.
- **Java é fortemente tipada**, ou seja, os tipos de dados das variáveis são definidos no momento de sua declaração e não podem mudar em tempo de execução.

Exercício

- Devido à linguagem de programação Java ser fortemente tipada, não é possível realizar conversão entre tipos de dados diferentes.
- Certo ou Errado?

Exercício

- Devido à linguagem de programação Java ser **fortemente tipada**, não é possível realizar conversão entre tipos de dados diferentes.
- Certo ou Errado?
- Resposta: **Errado!**
- Não podemos mudar o tipo de dado de uma variável já declarada, mas podemos criar uma nova e armazenar o resultado convertido.

Identificadores

- Identificam os nomes de: classes, variáveis, métodos ...
- Composição dos nomes dos identificadores de variáveis:
 - iniciam com **letra**, **_** ou **\$**
 - continua com letras, números, **_** ou **\$**
 - case sensitive
- Exemplos de identificadores **válidos**:
 - soma, n1, primeiro_nome, _aux, \$aux
- Exemplos de identificadores **inválidos**:
 - 1num, ?sujeito, #literal, %total

Exercício

Assinale a opção que contém apenas identificadores válidos:

a) A1, 1B, _2

b) _R, \$45, %4

c) U1, _4, \$T

d) #3, 88, AB

e) 2W, 33, -R

Exercício

Assinale a opção que contém apenas identificadores válidos:

- a) A1, 1B, _2
- b) _R, \$45, %4
- c) U1, _4, \$T
- d) #3, 88, AB
- e) 2W, 33, -R

Resposta: C

Palavras Reservadas

Mais um detalhe: não podemos utilizar **palavras reservadas** para nomes de identificadores!

Lista de Palavras Reservadas



AS PALAVRAS
RESERVADAS **CONST** E
GOTO NÃO SÃO
COMANDOS JAVA, NO
ENTANTO, SÃO PALAVRAS
RESERVADAS.

abstract
boolean
break
byte
case
catch
char
class
const
continue
default

do
double
else
extends
false
final
finally
float
for
goto
assert

if
implements
import
instanceof
int
interface
long
native
new
null

package
private
protected
public
return
short
static
strictfp
super
switch

synchronized
this
throw
throws
transient
true
try
void
volatile
while

Variáveis

As variáveis podem ser de dois tipos:

1. **primitivos** (ou de valor): int, double, float...
2. **referência** (para um objeto): String, Scanner

Observe que tipos **primitivos**, começam com letra **minúscula** e de **referência** (tipos de objetos) começam com letras **maiúsculas**.

Quais são os tipos primitivos?

- **byte** – número inteiro com sinal (8 bits)
- **short** – número inteiro com sinal (16 bits)
- **int** – número inteiro com sinal (32 bits)
- **long** – número inteiro com sinal (64 bits)
- **float** – número em ponto-flutuante (32 bits)
- **double** – número em ponto-flutuante (64 bits)
- **boolean** – true ou false (8 bits)
- **char** – character UNICODE (16 bits)

Tipos de Dados Primitivos x Qtd de Bits

8bits	byte	boolean
16bits	short	char
32bits	int	float
64bits	long	double

Exemplos de Tipos Primitivos

- byte: 4b ou 4B
- short: 5s ou 5S
- int: -2
- long: 1200l ou 1200L
- float: 32.3f ou 32.3F
- double: 5.25, -0.333001
- boolean: true ou false
- char (não é String): 'a', 'A', '*'

**Ano: 2022 Banca: VUNESP Órgão: Prefeitura de Sorocaba - SP Prova:
VUNESP - 2022 - Prefeitura de Sorocaba - SP - Analista de Sistemas I**

A linguagem Java, assim como outras linguagens de programação, permite a representação de diferentes tipos de dados. O número de bits necessários para representar o tipo short na linguagem Java é igual a

- a) 2
- b) 8
- c) 16
- d) 32
- e) 64

Resposta: C

Representação na Memória

- Variáveis primitivas contêm um valor de tipo primitivo.

```
int a = 2;
```

a 

```
int b = a;
```

b 

- Já as variáveis de referência, que armazenam um objeto, salvam uma referência ao objeto!

Representação na Memória

1 String nome, apelido;

2 nome = "João";

3 apelido = nome;

4 nome = null;

5 System.out.println(apelido);

- Qual o valor final da variável apelido? Teste no eclipse!

Representação na Memória

1 String nome, apelido;

2 nome = "João";

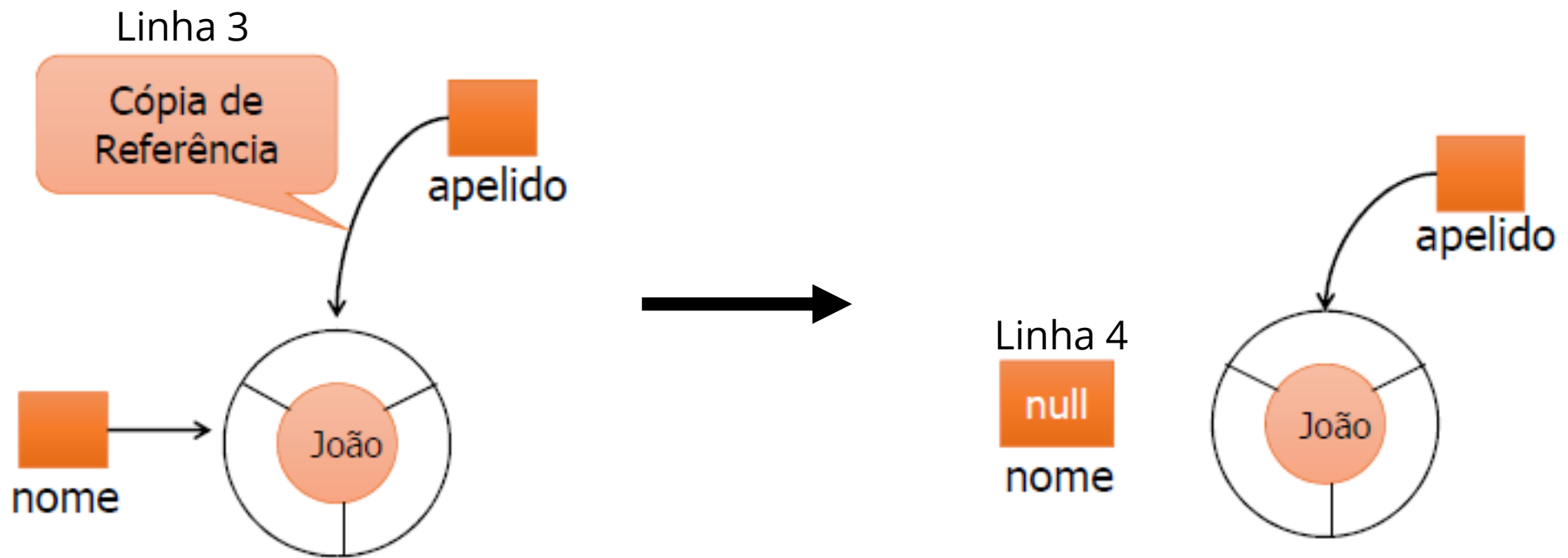
3 apelido = nome;

4 nome = null;

5 System.out.println(apelido);

- Qual o valor final da variável apelido? Teste no eclipse!
- R: João! Porque ainda existirá uma referência apontando para o objeto João.

```
1 String nome, apelido;  
2 nome = "João";  
3 apelido = nome;  
4 nome = null;  
5 System.out.println(apelido);
```



Constantes

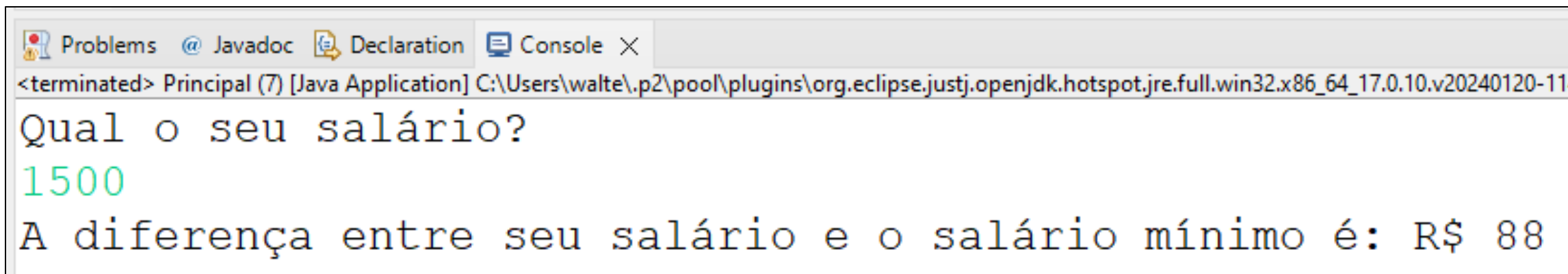
- Lembram que não existe o comando `const` em Java?
- Para declarar uma constante, utilizamos o modificador `final`!
- Sintaxe:

```
final <tipo> <identificador> = <valor> ;
```

- Constantes só podem ser inicializadas 1 vez.
- Aconselha-se o uso de letras maiúsculas no identificador de uma constante.
- Exemplos:
 - `final int LIMITE = 200;`
 - `final String TÍTULO = "abc";`

Exercício

1. Crie uma constante chamada de SALÁRIOMÍNIMO (R\$ 1.412,00) e inicialize o valor dela.
2. Em seguida pergunte ao usuário qual o seu salário e armazene o valor digitado na variável **salário**.
3. Apresente ao usuário a diferença entre o salário dele e o salário mínimo.



```
Problems @ Javadoc Declaration Console ×  
<terminated> Principal (7) [Java Application] C:\Users\walte\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-11  
Qual o seu salário?  
1500  
A diferença entre seu salário e o salário mínimo é: R$ 88
```

Resposta

```
*Principal.java X
1 import java.util.Scanner;
2
3 public class Principal {
4     public static void main(String[] args) {
5         final int SALARIOMINIMO = 1412;
6         Scanner entrada = new Scanner(System.in);
7         System.out.println("Qual o seu salário?");
8         int salario = entrada.nextInt();
9         System.out.println("A diferença "
0             + "entre seu salário e o "
1             + "salário mínimo é: R$ "
2             + (salario - SALARIOMINIMO) );
3         entrada.close();
4     }
5 }
```

Problems @ Javadoc Declaration Console X

<terminated> Principal (7) [Java Application] C:\Users\walte\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-11

Qual o seu salário?

1500

A diferença entre seu salário e o salário mínimo é: R\$ 88

Exercício – Fazer agora

- Identifique no exercício anterior:
 - Quem são as classes?
 - Onde estão as variáveis?
 - Onde estão os métodos?
- O que são classes? O que são objetos? O que são métodos?

Lista de Exercícios 2

PROGRAMAÇÃO ORIENTADA A OBJETOS

AULA 2 – SINTAXE DA LINGUAGEM E VARIÁVEIS

WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR