Strings e Arrays

POO

Prof. Marcio Delamaro

String

- Uma sequência de caracteres que representa uma informação textual
- Em Java existe um tipo específico para representar strings
- Na verdade, um string é também um objeto, instanciado a partir da classe String

String

- Uma sequência de caracteres que representa uma informação textual
- Em Java existe um tipo específico para representar strings
- Na verdade, um string é também um objeto, instanciado a partir da classe String
- String s = new String(“abc”);
  String p = “abc”;

Algumas coisas que podemos fazer

- ```
  String s = “abc” + “def”;
  String r = “ghi”;
  r += s;
  ```
- ```
  System.out.println(“O valor final é: ” + 25);
  ```

String é objeto

- Possui atributos e métodos...
- ```
  String s = "abc";
  System.out.println("O tamanho eh: " + s.length())
  ```
- ```
  if (s.equals("abc"))
          System.out.println("Sao iguais");
  ```
- ```
  String r = "Meu nome não é Jony".substring(4, 8);
  ```
- ```
  int i = r.indexOf('o');
  ```

Como conhecer uma classe

- A linguagem Java possui uma vasta biblioteca de classes
- API (Application Programming Interface)
- É preciso saber procurar e entender a sua documentação
- http://docs.oracle.com/javase/8/docs/api/index.html
- Se tiver sorte, pode fazer uma busca como: ***api java 8 string***

Exercícios

- Leia um string do teclado usando `EntradaTeclado.leString()`
- Mostre quantas vezes cada vogal aparece
- Transforme e exiba o string só em maiúsculas
- Leia um segundo string e verifique se ele aparece no início ou no fim do primeiro

Aproveitando

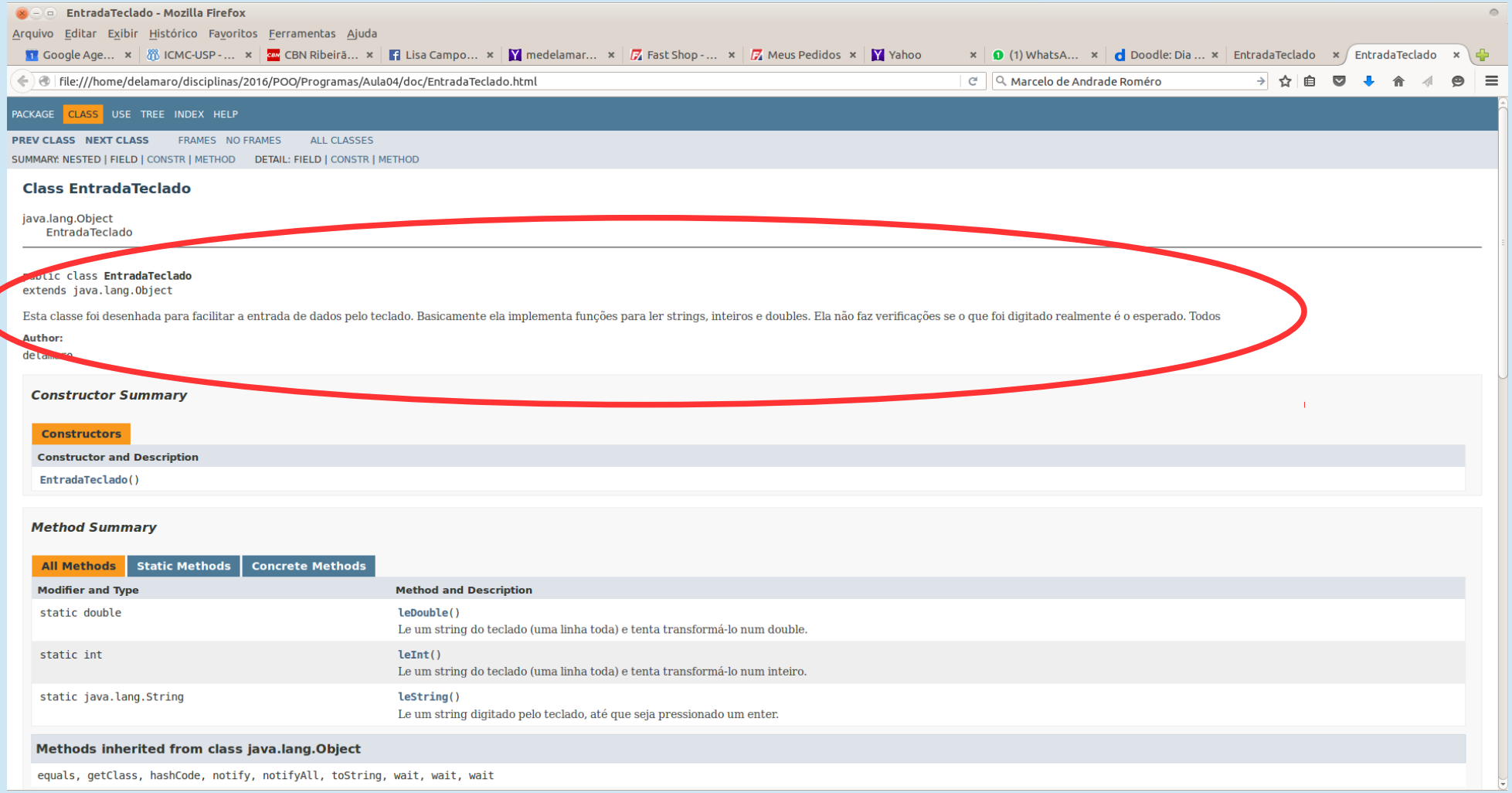
- A documentação que vimos sobre a classe String não é gerada manualmente
- Ela é extraída do código fonte
- É fornecida por meio de comentários
- Nós também podemos criar esse tipo de documentação
- Javadoc

Antes da declaração de classe

- Usamos sempre /** comentário */
- Antes da declaração da classe colocamos um descrição geral sobre ela

```
/**
* Esta classe foi desenhada para facilitar a entrada de dados pelo
* teclado. Basicamente ela implementa funções para ler strings, inteiros
* e doubles. Ela não faz verificações se o que foi digitado realmente é o
* esperado.
* @author delamaro
*
*/
public class EntradaTeclado {
```

Resultado


EntradaTeclado - Mozilla Firefox
Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda
file:///home/delamaro/disciplinas/2016/POO/Programas/Aula04/doc/EntradaTeclado.html
PACKAGE CLASS USE TREE INDEX HELP
PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
Class EntradaTeclado
java.lang.Object
EntradaTeclado
public class EntradaTeclado
extends java.lang.Object
Esta classe foi desenhada para facilitar a entrada de dados pelo teclado. Basicamente ela implementa funções para ler strings, inteiros e doubles. Ela não faz verificações se o que foi digitado realmente é o esperado. Todos
Author:
Constructor Summary
Constructors
Constructor and Description
EntradaTeclado()
Method Summary
All Methods
Static Methods
Concrete Methods
Modifier and Type
Method and Description
static double
leDouble()
Le um string do teclado (uma linha toda) e tenta transformá-lo num double.
static int
leInt()
Le um string do teclado (uma linha toda) e tenta transformá-lo num inteiro.
static java.lang.String
leString()
Le um string digitado pelo teclado, até que seja pressionado um enter.
Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Antes dos métodos

- Antes de cada método colocamos uma descrição do que ele faz
- Quais os parâmetros que recebe
- O que ele retorna
- Quais exceções pode lançar

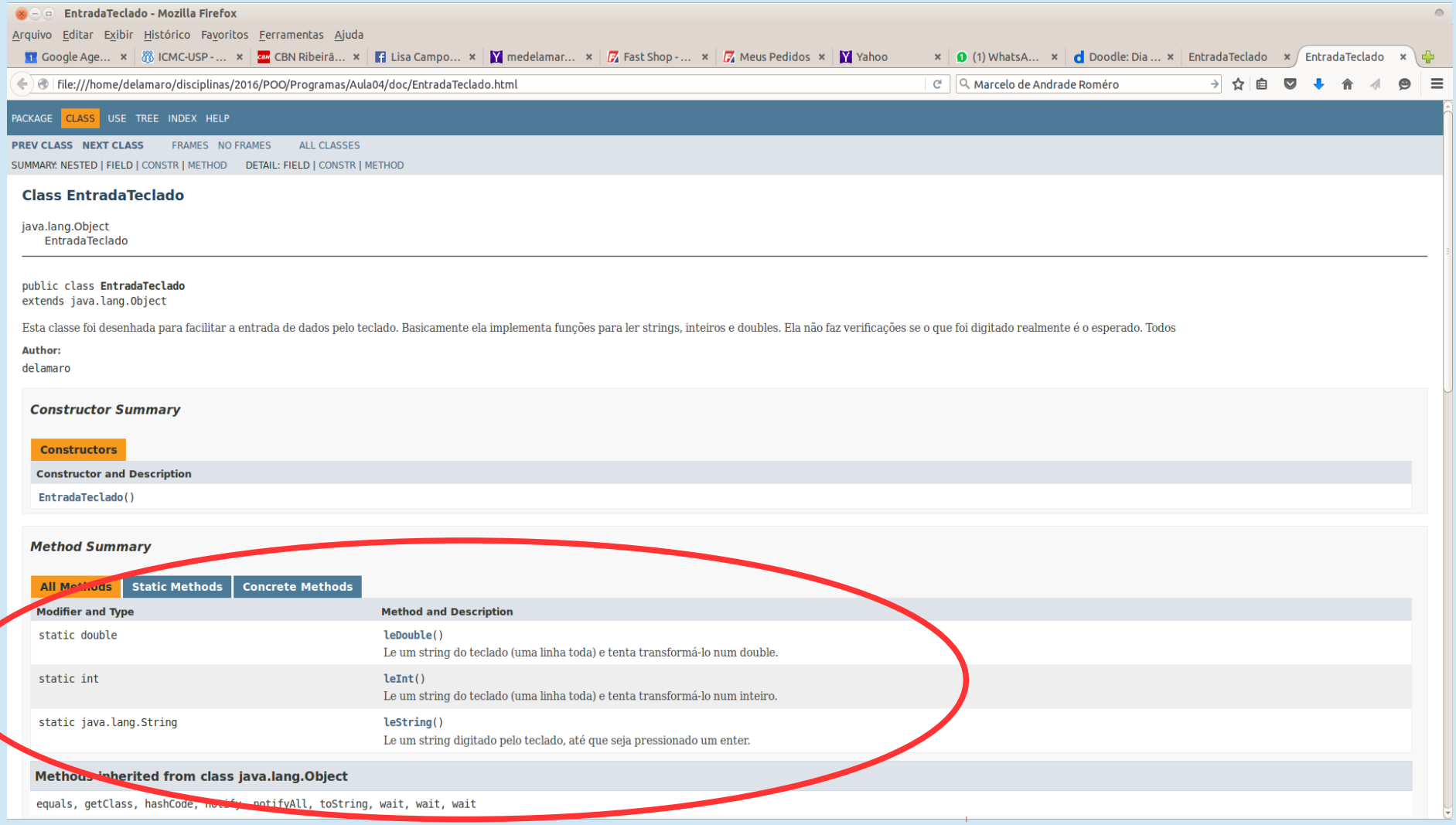
Antes dos métodos

```
/**
* Le um string digitado pelo teclado, até que seja
* pressionado um enter. Ou seja, le a linha toda.
* @return o string que foi digitado pelo usuário.
* @throws IOException Essa exceção é lançada quando o
* método não consegue ler o string a partir do teclado.

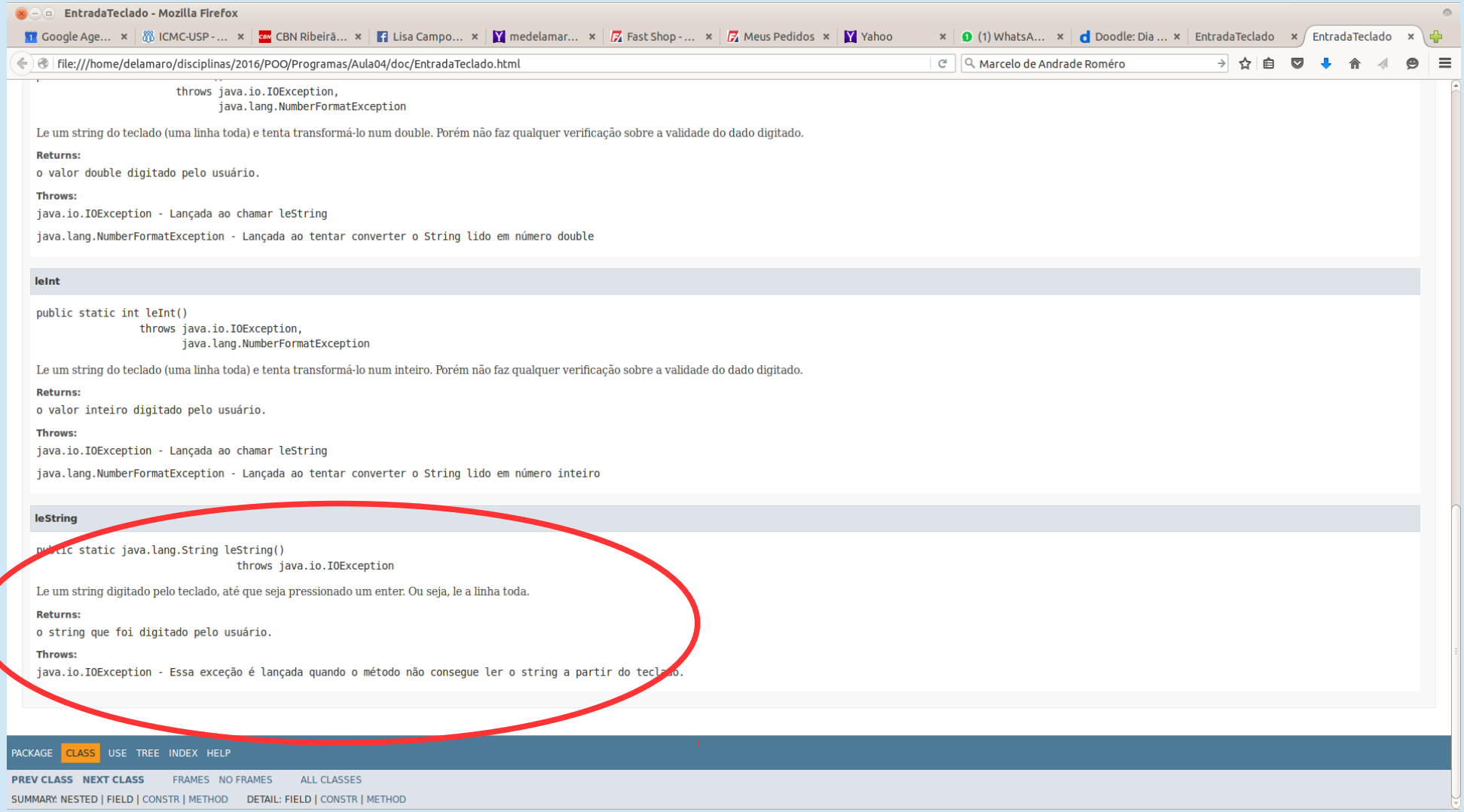
*/

public static String leString() throws IOException {
```

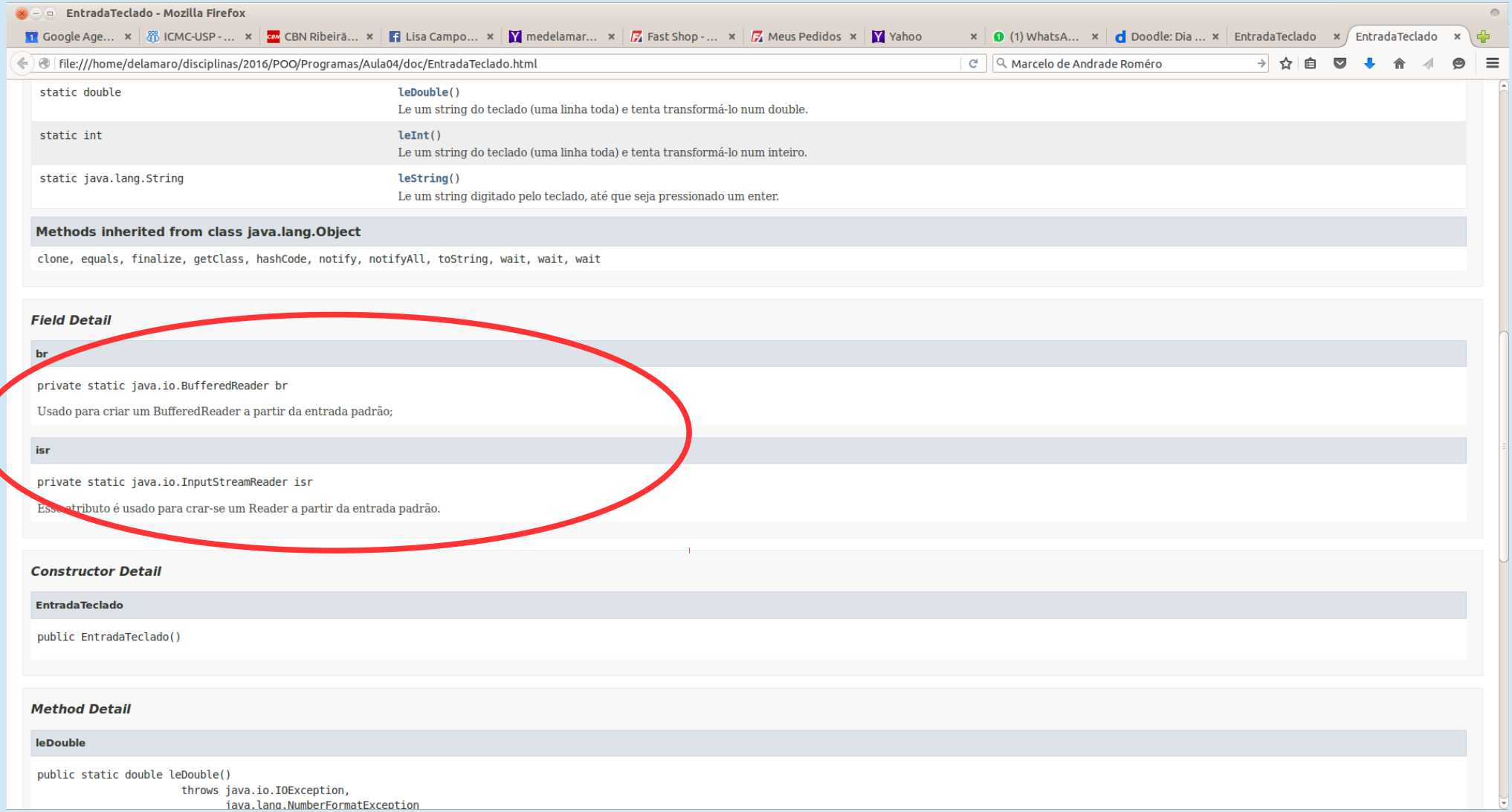
Resultado


EntradaTeclado - Mozilla Firefox
Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda
file:///home/delamaro/disciplinas/2016/POO/Programas/Aula04/doc/EntradaTeclado.html
PACKAGE CLASS USE TREE INDEX HELP
PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
Class EntradaTeclado
java.lang.Object
EntradaTeclado
public class EntradaTeclado
extends java.lang.Object
Esta classe foi desenhada para facilitar a entrada de dados pelo teclado. Basicamente ela implementa funções para ler strings, inteiros e doubles. Ela não faz verificações se o que foi digitado realmente é o esperado. Todos
Author:
delamaro
Constructor Summary
Constructors
Constructor and Description
EntradaTeclado()
Method Summary
All Methods
Static Methods
Concrete Methods
Modifier and Type
Method and Description
static double
leDouble()
Le um string do teclado (uma linha toda) e tenta transformá-lo num double.
static int
leInt()
Le um string do teclado (uma linha toda) e tenta transformá-lo num inteiro.
static java.lang.String
leString()
Le um string digitado pelo teclado, até que seja pressionado um enter.
Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Resultado


EntradaTeclado - Mozilla Firefox
file:///home/delamaro/disciplinas/2016/POO/Programas/Aula04/doc/EntradaTeclado.html
throws java.io.IOException,
java.lang.NumberFormatException
Le um string do teclado (uma linha toda) e tenta transformá-lo num double. Porém não faz qualquer verificação sobre a validade do dado digitado.
Returns:
o valor double digitado pelo usuário.
Throws:
java.io.IOException - Lançada ao chamar leString
java.lang.NumberFormatException - Lançada ao tentar converter o String lido em número double
leInt
public static int leInt()
throws java.io.IOException,
java.lang.NumberFormatException
Le um string do teclado (uma linha toda) e tenta transformá-lo num inteiro. Porém não faz qualquer verificação sobre a validade do dado digitado.
Returns:
o valor inteiro digitado pelo usuário.
Throws:
java.io.IOException - Lançada ao chamar leString
java.lang.NumberFormatException - Lançada ao tentar converter o String lido em número inteiro
leString
public static java.lang.String leString()
throws java.io.IOException
Le um string digitado pelo teclado, até que seja pressionado um enter. Ou seja, le a linha toda.
Returns:
o string que foi digitado pelo usuário.
Throws:
java.io.IOException - Essa exceção é lançada quando o método não consegue ler o string a partir do teclado.
PACKAGE CLASS USE TREE INDEX HELP
PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Atributos – Resultado



Arrays

- Um array em Java também é um tipo especial de objeto
- Precisa sempre ser instanciado
- Os elementos do array podem ser tipos primitivos ou objetos
- ```
  int[] vetInt = new int[20];
  int t = vetInt.length;
  ```
  ```

Coisas que podemos fazer

- `int[] v2 = new int[] {0 , 1963, 0, 0, 0} ;`
- `String[] vs = new String[10];`
- `String[] xv = new String[] { "abc", "def", ""};`
- `Random[] vetRand = new Random[4];`

Coisas que podemos fazer

- `int[] v2 = new int[] {0 , 1963, 0, 0, 0} ;`
- `String vs = new String[10];`
- `String[] xv = new String[] { "abc", "def", ""};`
- `Random[] vetRand = new Random[4];`
- ATENÇÃO: NÃO HÁ NADA NO ARRAY (null)
-
```
vetRand[0] = new Random();
vetRand[1] = new Random(110);
…
```
-
```
String vs = null;
System.out.println(vs);
```

Matrizes

- Um array pode ter mais do que uma dimensão
- int[][] m = new int[5][3];

Matrizes

- Um array pode ter mais do que uma dimensão
- `int[][] m = new int[3][5];`

m[0][0]	m[0][1]	m[0][2]	m[0][3]	m[0][4]
m[1][0]	m[1][1]	m[1][2]	m[1][3]	m[1][4]
m[2][0]	m[2][1]	m[2][2]	m[2][3]	m[2][4]

Matrizes

- Um array pode ter mais do que uma dimensão
- `int[][] m = new int[3][5];`

m[0][0]	m[0][1]	m[0][2]	m[0][3]	m[0][4]
m[1][0]	m[1][1]	m[1][2]	m[1][3]	m[1][4]
m[2][0]	m[2][1]	m[2][2]	m[2][3]	m[2][4]

- `int[][] m = new int[][] { {1,2,3}, {4,5,6}, {7,8,9} };`

1	2	3
4	5	6
7	8	9

A verdade

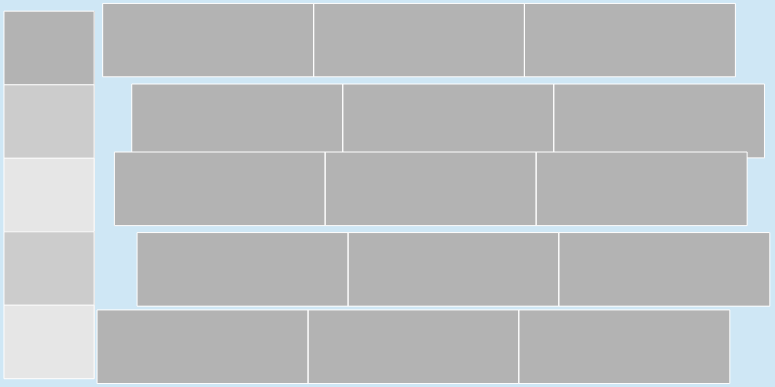
- Uma matriz é na verdade um array
- Cada elemento do array é um array

```
int[][] m = new int[5][3];
```

A verdade

- Uma matriz é na verdade um array
- Cada elemento do array é um array

```
int[][] m = new int[5][3];
```



O que podemos fazer

- ```
  m = new int[][] { {1, 2, 3} ,
   {4,5}, {6} };
  ```

- ```
  m = new int[5][];
  for (i = 0; i < 5; i++)
        m[i] = new int[5-i];
  ```

Treinando

- 1. Crie uma classe para representar uma pessoa, com os atributos privados de nome, idade e altura. Crie os métodos públicos necessários para sets e gets e também um método para imprimir os dados de uma pessoa.

- 2. Crie uma classe Agenda que pode armazenar 10 pessoas e seja capaz de realizar as seguintes operações:
  - void armazenaPessoa(String nome, int idade, float altura);
  - void removePessoa(String nome);
  - int buscaPessoa(String nome); // informa em que posição da agenda está a pessoa
  - void imprimeAgenda(); // imprime os dados de todas as pessoas da agenda
  - void imprimePessoa(int i); // imprime os dados da pessoa que está na posição "i" da agenda

Exercício

- Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio, excluindo o térreo, capacidade do elevador, e quantas pessoas estão presentes nele.
- A classe deve também disponibilizar os seguintes métodos:
  - Construtor: que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
  - Entra: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
  - Sai: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
  - Sobe: para subir um andar (não deve subir se já estiver no último andar);
  - Desce: para descer um andar (não deve descer se já estiver no térreo);
- Encapsular todos os atributos da classe (criar os métodos set e get).

Prática

- Crie uma classe Televisão e uma classe ControleRemoto que pode controlar o volume e trocar os canais da televisão.
- O controle de volume permite aumentar ou diminuir a potência do volume de som em uma unidade de cada vez;
- O controle de canal também permite aumentar e diminuir o número do canal em uma unidade, porém, também possibilita trocar para um canal indicado;
- Também devem existir métodos para consultar o valor do volume de som e o canal selecionado.