

Revisão

- `java.lang`
 - Funcionalidades essenciais à linguagem Java
- `java.io`
 - Leitura e escrita em arquivos
- `java.util`
 - Trabalhar com coleção de objetos

java.lang

- Object
- System
 - out
 - in
- Wrapper classes
 - Integer, Double, Float, Boolean, Byte, Short, Long ...

java.io

- Pacote com classes que ajudam na manipulação de fluxos de dados, como arquivos por exemplo...

java.io

- Exemplo
 - Vamos escrever os dados de uma Pessoa num arquivo e depois ler esse arquivo e armazenar em Pessoa novamente!
 - Pessoa
 - nome: string
 - idade: int
 - sexo: char
 - telefone: String

java.util

- Pacote responsável por funcionalidades como collections, vetores, etc...
- Documentação*

java.util

- Arrays & Vectors
 - Arrays e Vectors apesar de possuírem conceitos semelhantes eles funcionam de formas diferentes...

java.util

- Exemplo

- Arrays possuem o seu tamanho fixados na hora da sua criação...

- `int numeros[] = new int[5];`

- Collections podem ter seus tamanhos alterados

- `Vector nomes = new Vector();`

- `numeros.addElement("casa");`

java.util

- Obs:
 - Vectors trabalham sobre Object!
 - Tipos primitivos não são permitidos
 - Wrapper Classes

Como armazenar e recuperar um objeto Pessoa em um Vector ?

java.util

- **Vectors**

- void add(Object o);
- int size();
- Object elementAt(int index);
- Object firstElement();
- Object lastElement();
- void removeElementAt(int index);
- Enumeration elements();

java.util

- Crie um Vector e insira 4 números diferentes..depois percorra esse vector e escreva esses números na tela

JavaDoc

- Ferramenta para gerar documentação dos códigos em java

JavaDoc

- Comentário em java são feitos de 3 formas
- `//`
- `/* */`
- `/** */` - comentários javadoc

JavaDoc

- Comentários javadoc serão interpretados pelo javadoc e usados para gerar documentação do sistema

JavaDoc

```
/** classe casa que define uma estrutura de  
casa*/
```

```
public class casa {  
}
```

JavaDoc

Os comentários javadoc podem ter tags HTML

JavaDoc

Além de tags comentários javadoc podem ter anotações que geram informações específicas sobre o código

ex.

@author autor

@version versao do sistema

JavaDoc

Anotações mais comuns

@author

@version

@param

@return

@throws | @exception