

# Pacotes, arquivos jar e implantação

IFC Bnu - Prof. Hylson



## Capítulo 17

*Todas as imagens destes slides foram obtidas do livro exibido acima ou em busca no google especificando filtro de licença Creative Commons*

# SIERRA, 2005

1. Pacotes
2. Arquivos .jar
3. Java web start



Pacotes

# Lance seu código

Você criou uma  
obra de arte!



O programa  
funciona mesmo!

# Lance seu código

Você criou uma  
obra de arte!

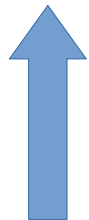


O programa  
funciona mesmo!

Como disponibilizar?  
Quem vai utilizar?

# Lance seu código

Organizar, empacotar e implantar



Como disponibilizar?  
Quem vai utilizar?

# Lance seu código

*Lançar não é adeus: sempre haverá a manutenção...*

Organizar, empacotar e implantar



Como disponibilizar?  
Quem vai utilizar?

# Formas de disponibilizar

Local





# Formas de disponibilizar

Local



Local + remoto

*Jnlp*

# Formas de disponibilizar

Local



Local + remoto

*Jnlp*

Remoto

**SimpleServlet**

Access servlet via [GET](#) request.

Access servlet via POST request.

# Formas de disponibilizar

Local



Local + remoto

*Jnlp*

Remoto

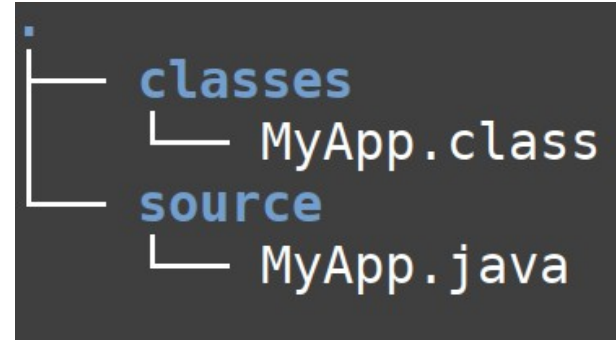
**SimpleServlet**

Access servlet via GET request.

Access servlet via **POST** request.

# Local

Passo 1: compilar código  
separando fontes e  
arquivos compilados



```
/source$ javac -d ../classes MyApp.java
```

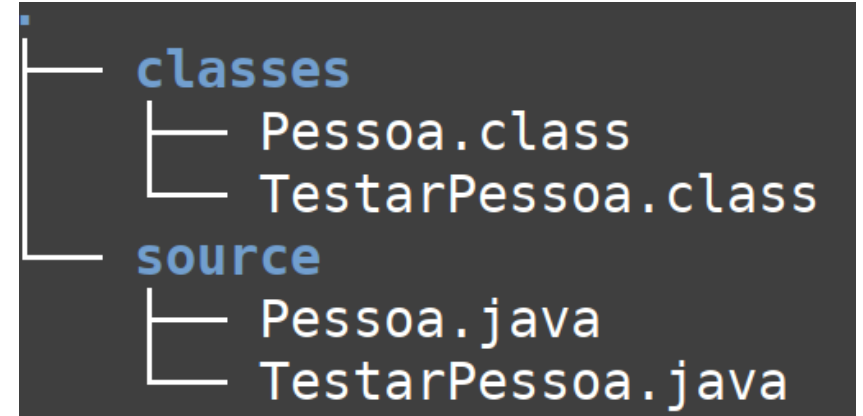
```
/classes$ java MyApp
```

ex01



# Local

Passo 1, exemplo 2

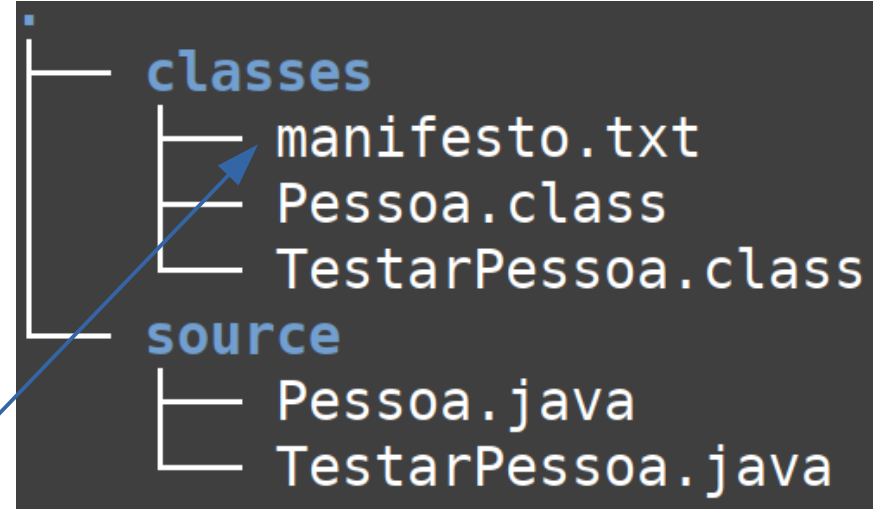


```
/source$ javac -d ../classes *.java
```

```
/classes$ java TestarPessoa
```

# Local

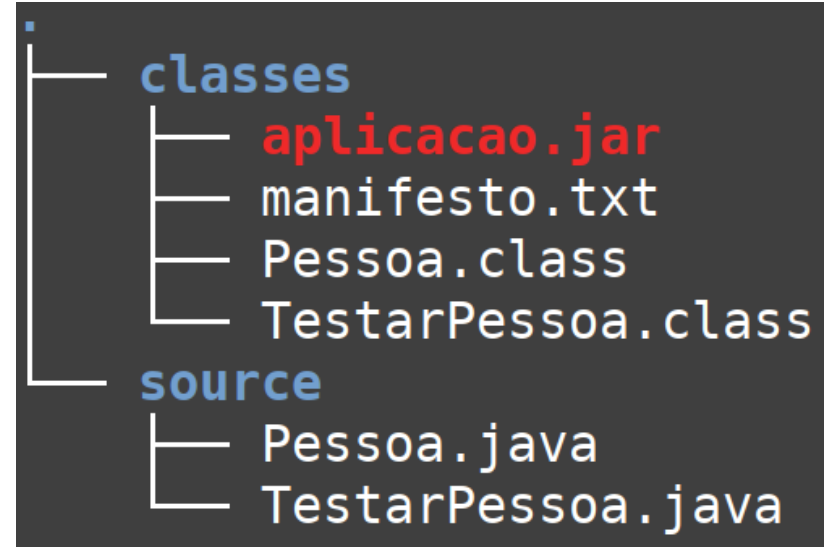
Passo 2: criar arquivo texto dentro da pasta classes, contendo nome da classe que possui o método main()



Main-class: TestarPessoa

# Local

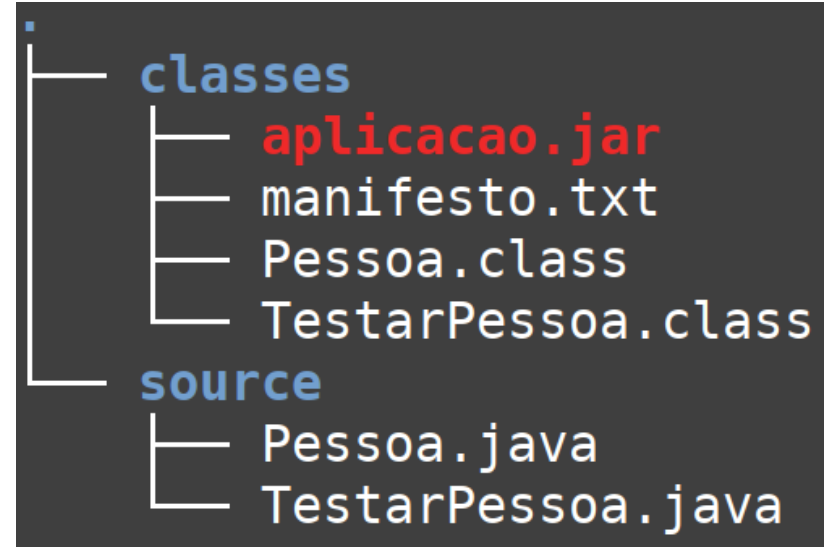
Passo 3: criar o arquivo  
“jar”



```
/classes$ jar -cvmf manifesto.txt aplicacao.jar *.class
added manifest
adding: Pessoa.class(in = 238) (out= 186)(deflated 21%)
adding: TestarPessoa.class(in = 1039) (out= 575)(deflated 44%)
```

# Local

Teste da execução:



```
$ java -jar classes/aplicacao.jar
```

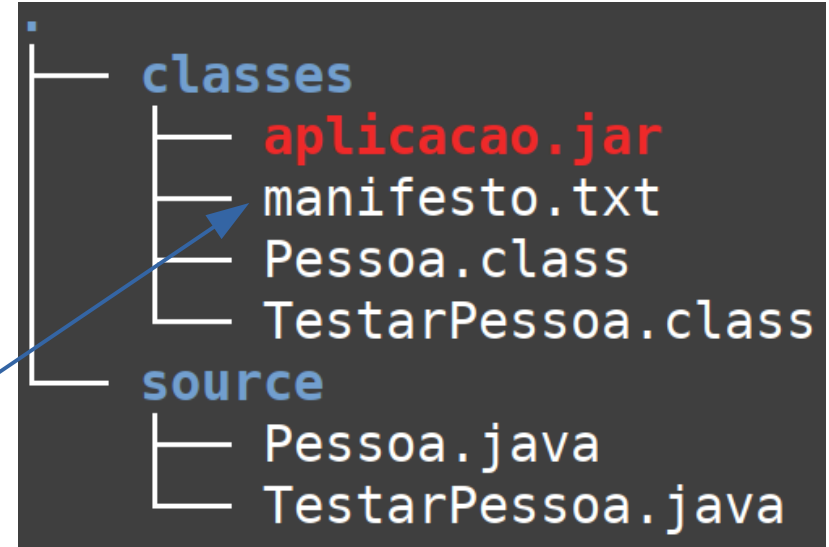


# Local

## Dentro do arquivo jar:

*O conteúdo do arquivo  
texto é inserido no  
arquivo MANIFEST.MF*

```
$ jar -tf classes/aplicacao.jar  
META-INF/  
META-INF/MANIFEST.MF  
Pessoa.class  
TestarPessoa.class
```



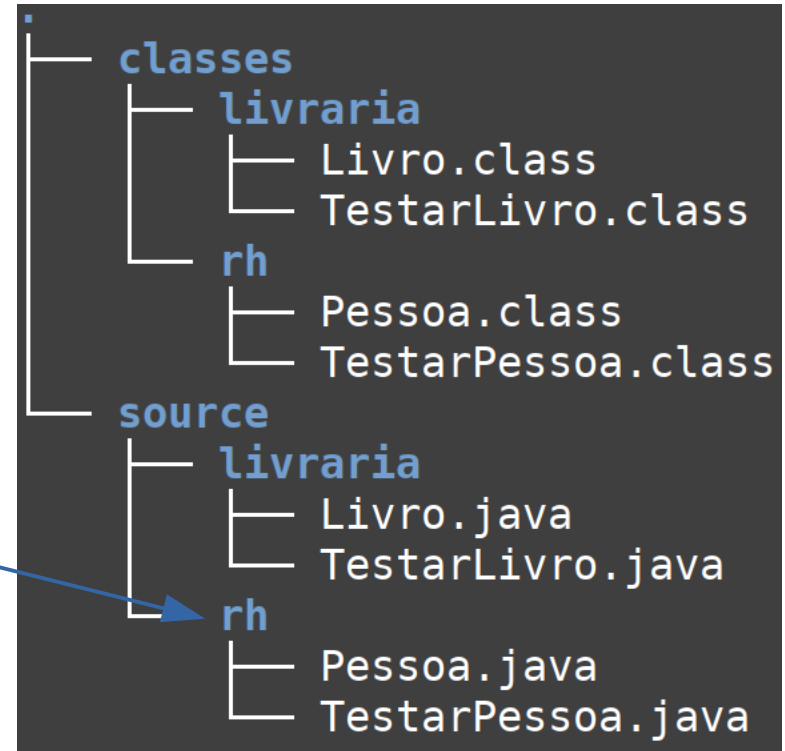
ex02



# Local com código fonte

Comando **package** para organizar classes

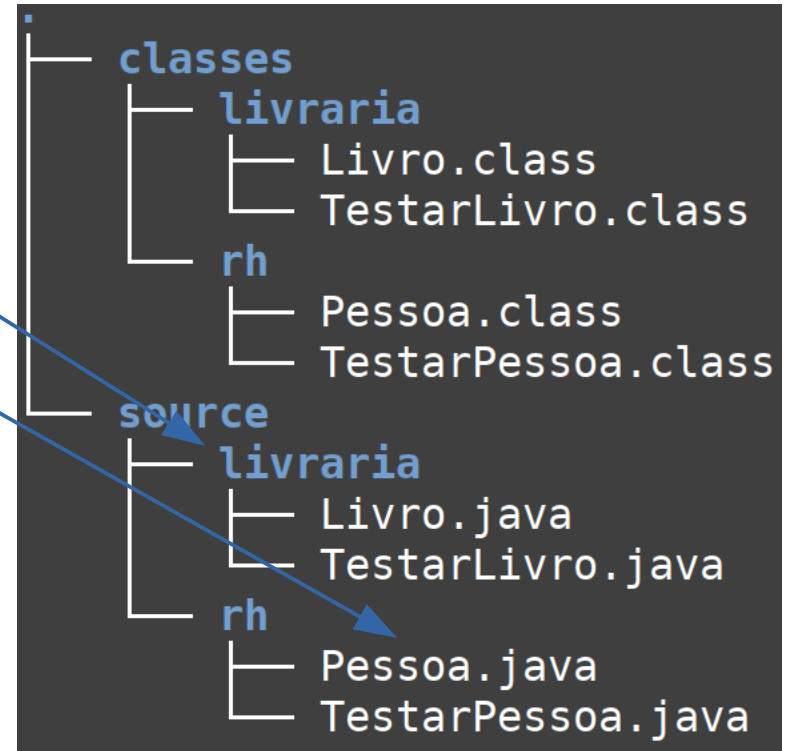
```
package rh;  
public class Pessoa {  
    ...
```

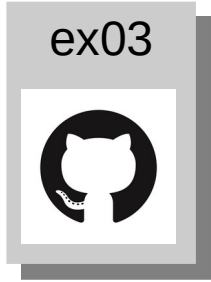


# Local com código fonte

```
package livraria;  
  
import rh.Pessoa;  
  
public class Livro {  
    public String nome;  
    public String autores;  
    public Pessoa proprietario;  
    ...  
}
```

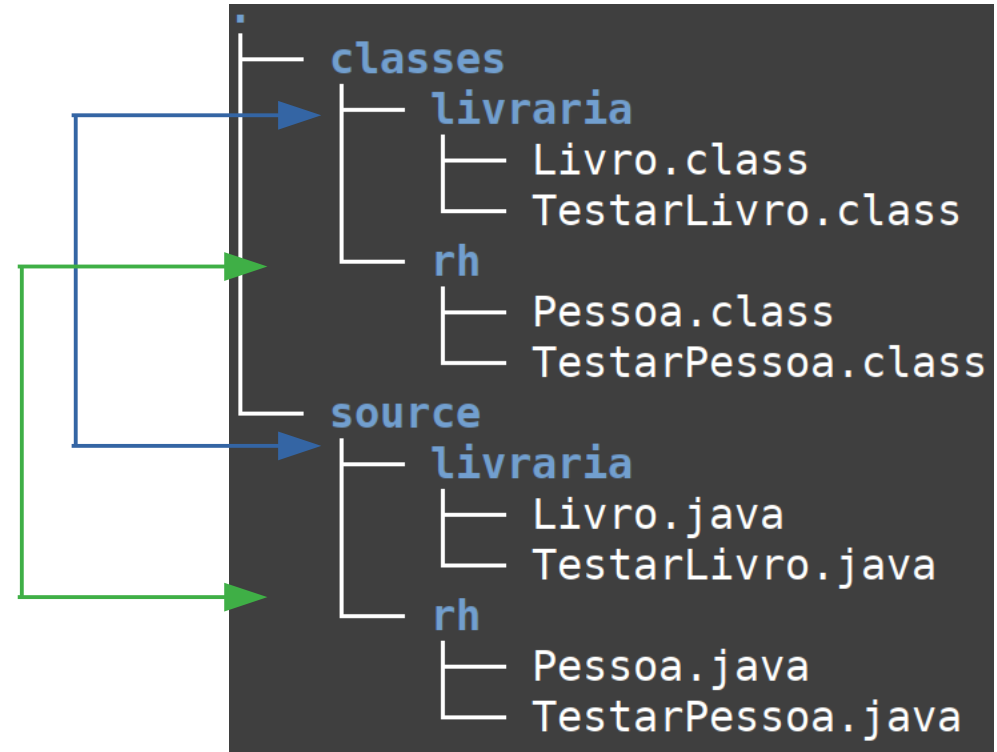
Comando **import** para  
usar classes que estão  
em outros pacotes





# Local com código fonte

*O compilador gera as classes nos diretórios especificados nos fontes pelos pacotes*

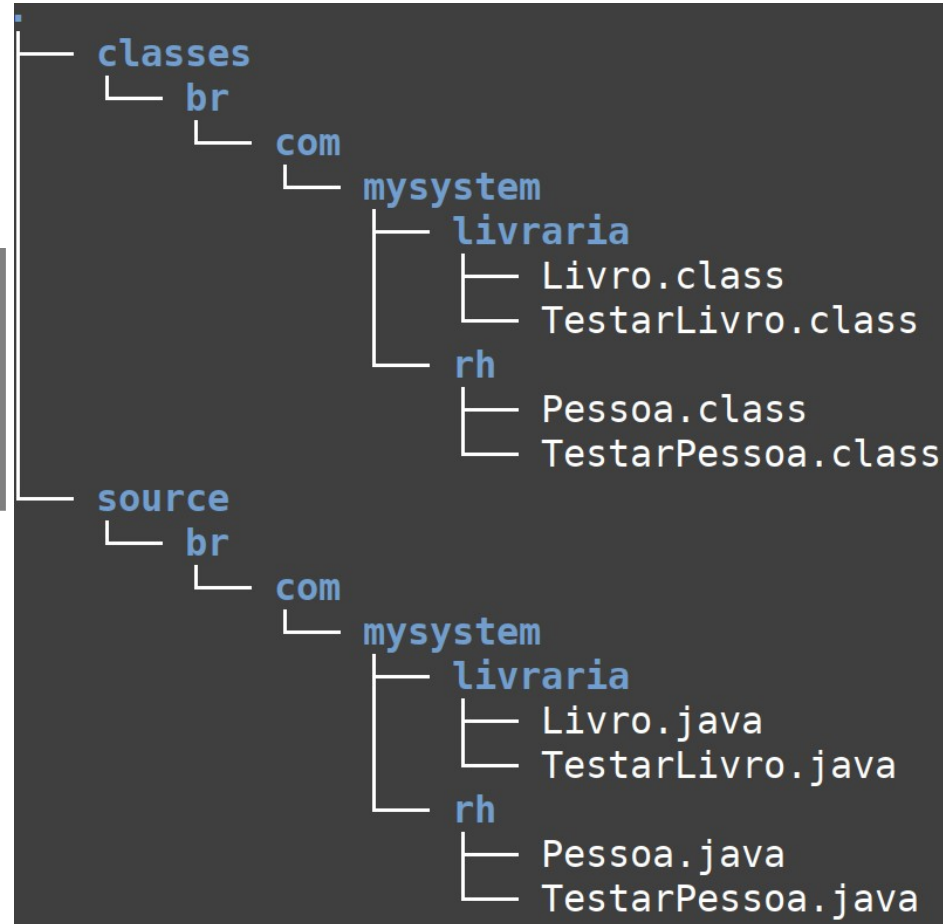
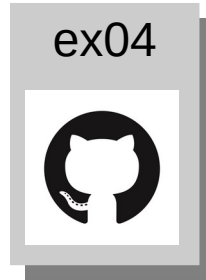


# Pacotes “mais” identificados

Sugestão: uso de domínio na ordem contrária.

Exemplo:

`mssystem.com.br`

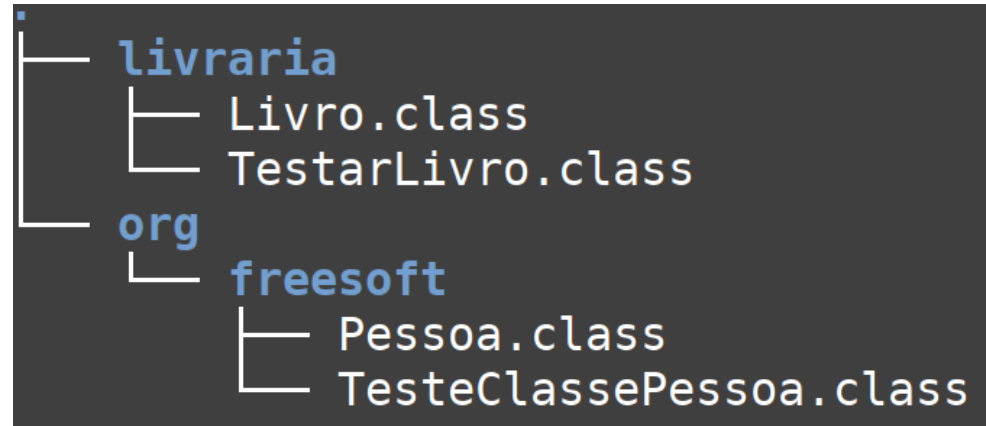


```
/source$ javac -d ../classes ./br/com/mssystem/rh/*.java
```

```
/source$ javac -d ../classes ./br/com/mssystem/livraria/*.java
```

# Evitando conflitos

Pessoa de freesoft.org sendo utilizado. A “outra” classe Pessoa não foi compilada.



Pessoa: nome, telefone

```
/source$ javac -d ../classes org/freesoft/*.java
```

```
/source$ javac -d ../classes livraria/*.java
```

ex05



# Utilizando a “outra” classe

```
package livraria;  
  
import br.com.mysystem.rh.Pessoa;  
  
public class TestarLivro {  
    ...  
}
```

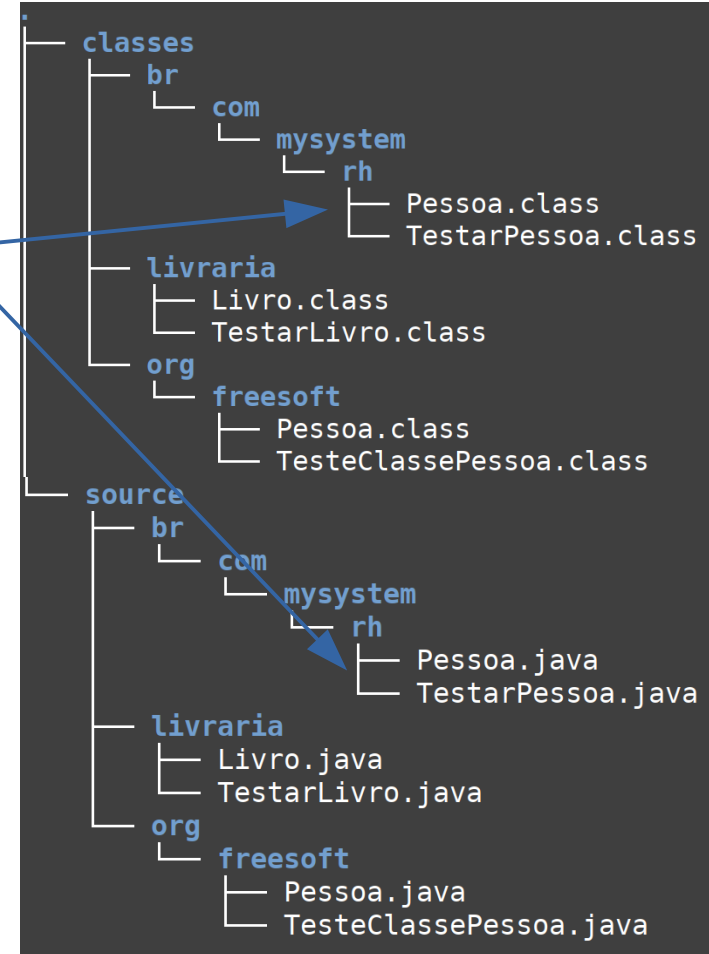
***Ambas estão compiladas, mas apenas uma está sendo usada***

```
/source$ javac -d ../classes br/com/mysystem/rh/*.java
```

```
/source$ javac -d ../classes org/freesoft/*.java
```

```
/source$ javac -d ../classes livraria/*.java
```

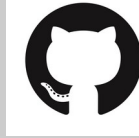
```
/classes$ java livraria.TestarLivro
```



# Utilizando a “outra” classe

```
package livraria;  
  
import br.com.mysystem.rh.Pessoa;  
  
public class TestarLivro {  
    ...  
}
```

ex06



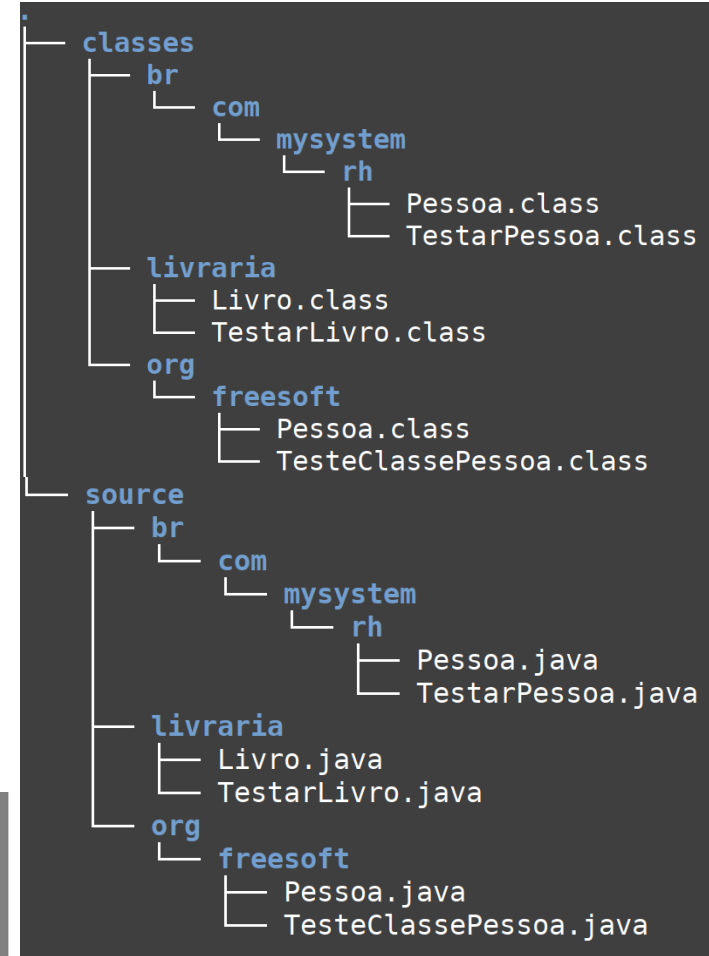
```
/source$ javac -d ../classes br/com/mysystem/rh/*.java
```

```
/source$ javac -d ../classes org/freesoft/*.java
```

```
/source$ javac -d ../classes livraria/*.java
```

```
/classes$ java livraria.TestarLivro
```

*Classes em pacotes:  
**execução usando o  
nome completo***





# Empacotando como biblioteca

*A livraria vai usar  
uma classe Pessoa*



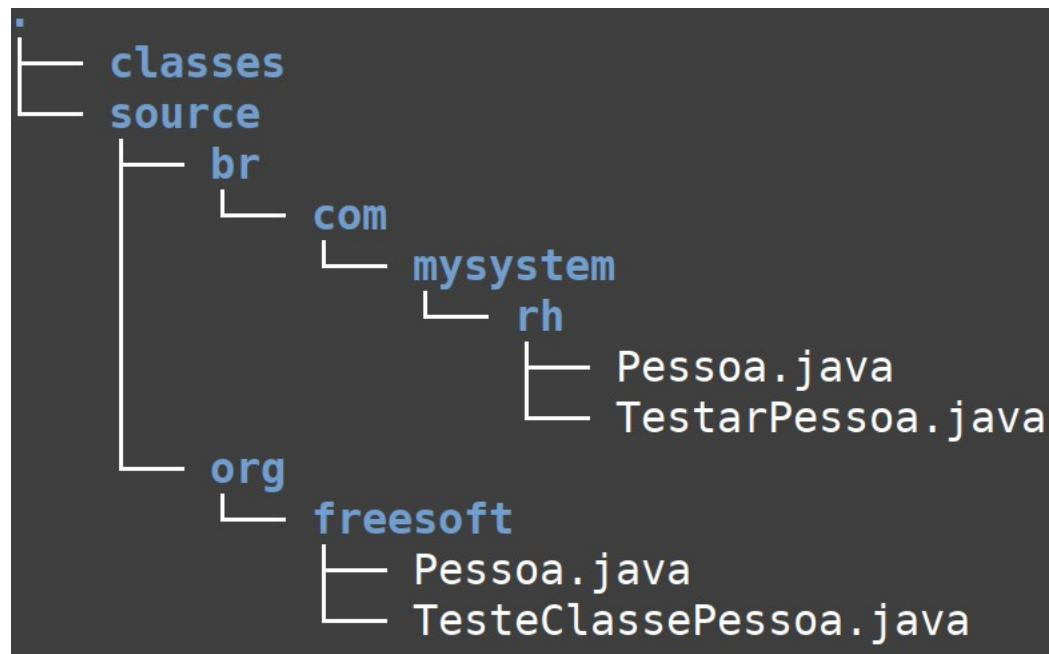
Cliente

*As duas implementações  
da classe Pessoa vão  
estar em um .jar*

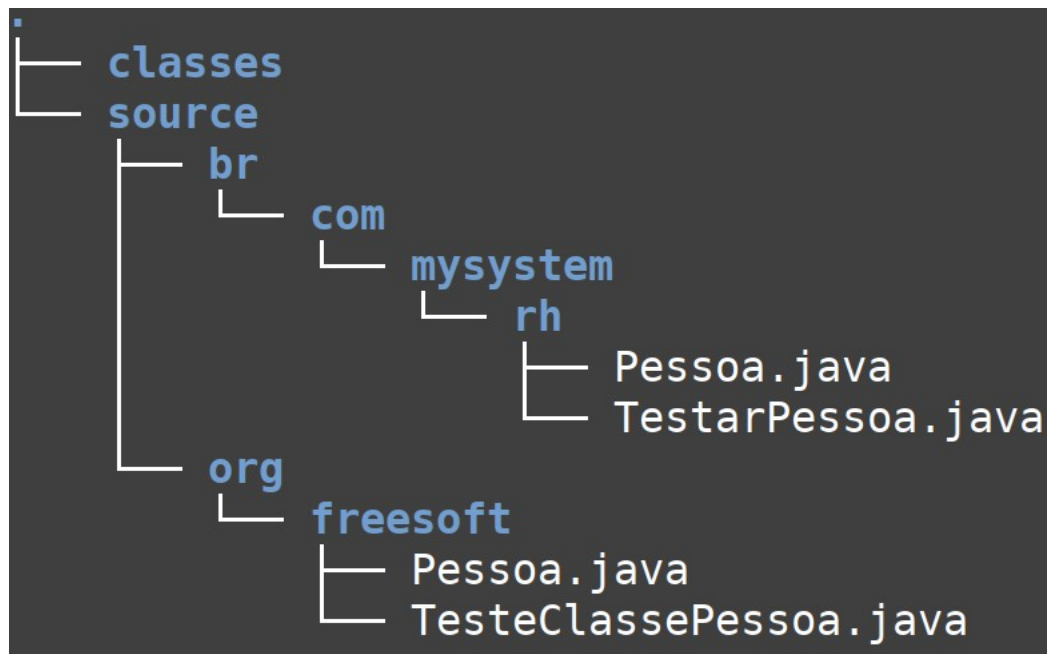


Biblioteca

# Empacotando como biblioteca



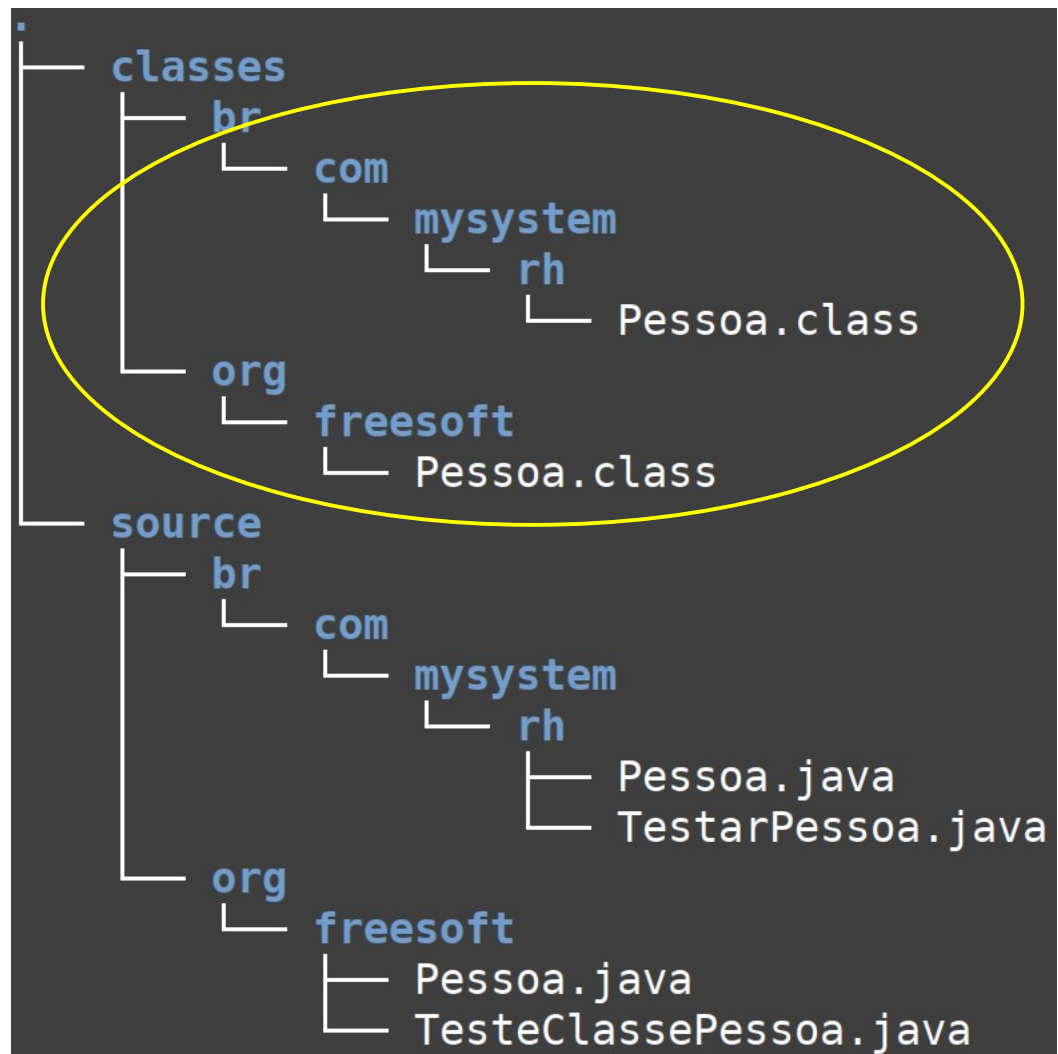
# Empacotando como biblioteca



Compilando:

```
/source$ javac -d ../classes br/com/mysystem/rh/Pessoa.java org/freesoft/Pessoa.java
```

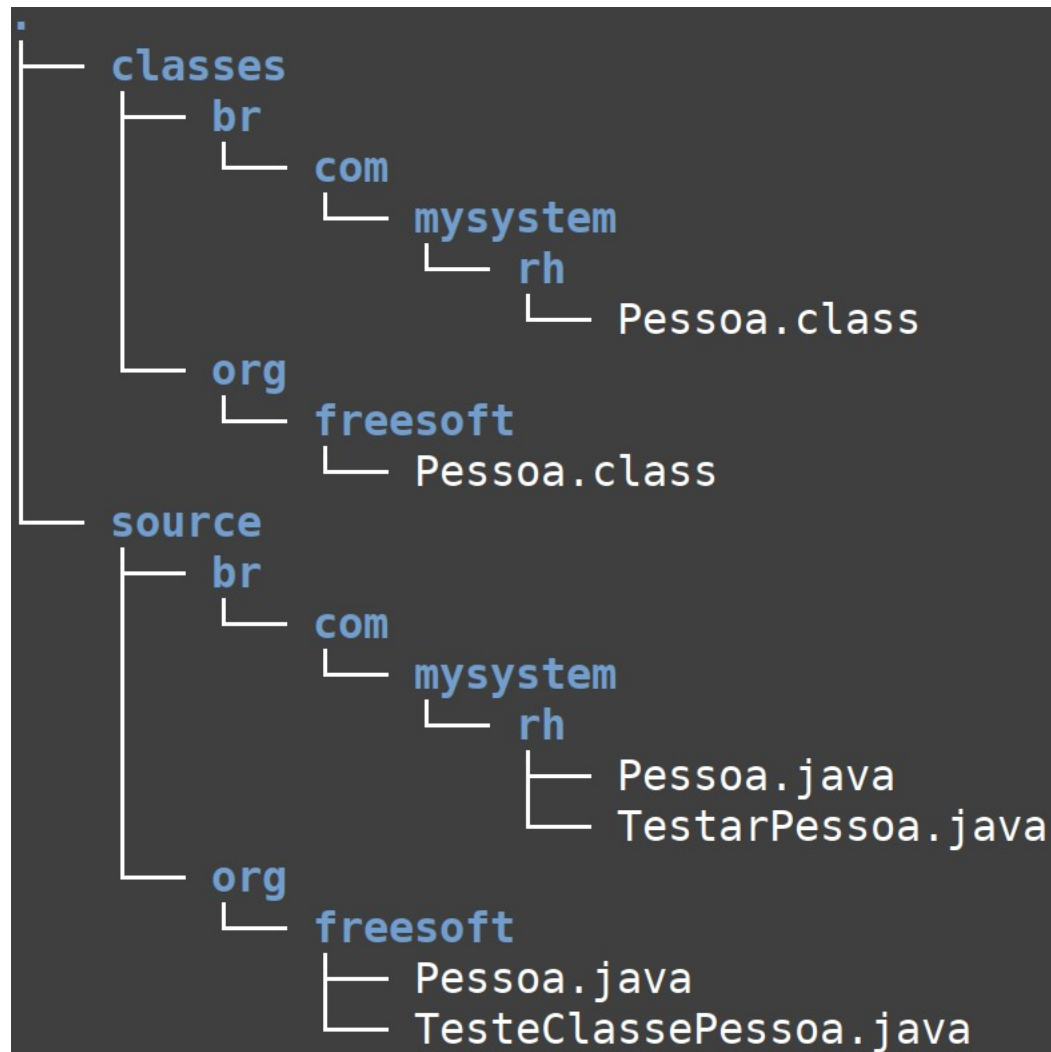
# Empacotando como biblioteca



# Empacotando como biblioteca

*-cf : não necessita  
arquivo manifesto  
informando main()*

## Empacotando:

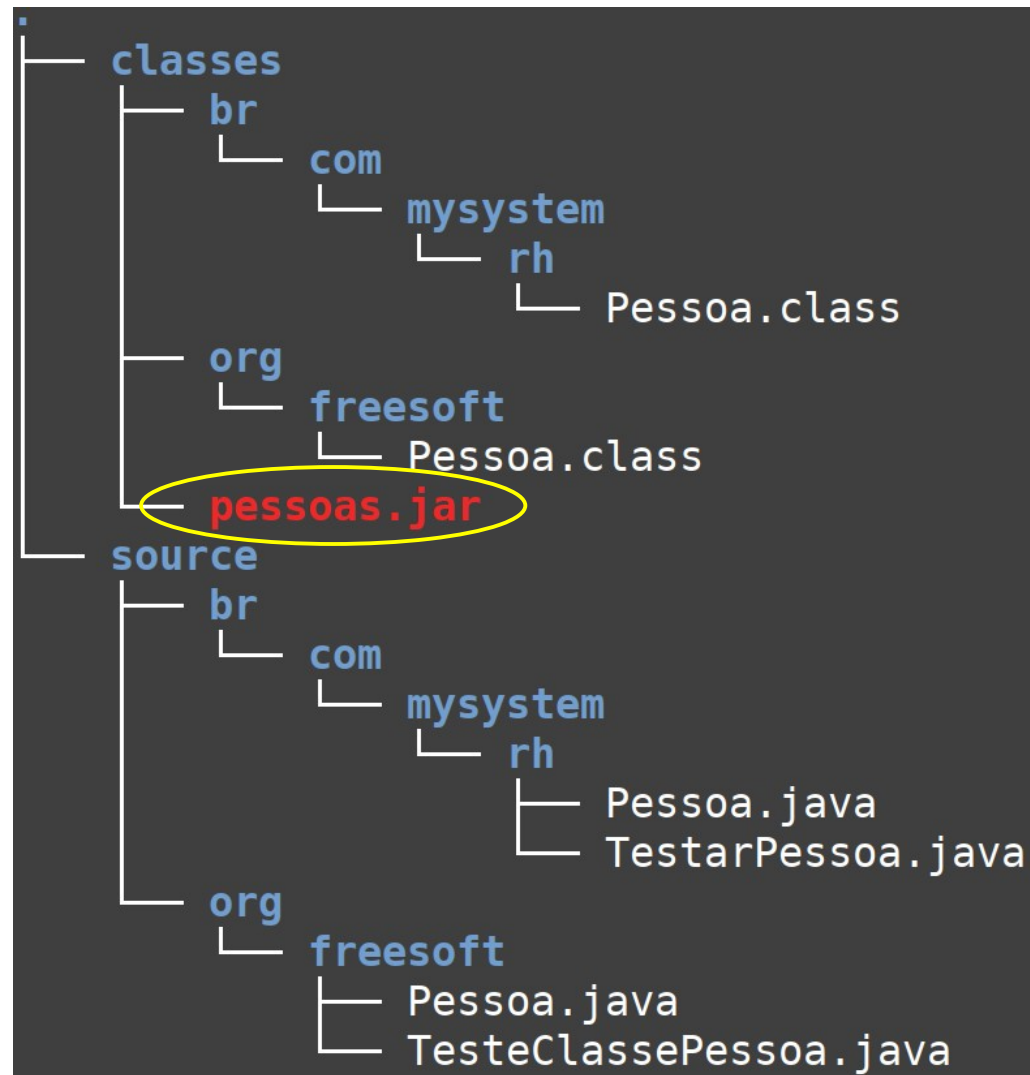


/classes\$ jar -cf pessoas.jar br/com/mysystem/rh/Pessoa.class org/freesoft/Pessoa.class

# Empacotando como biblioteca

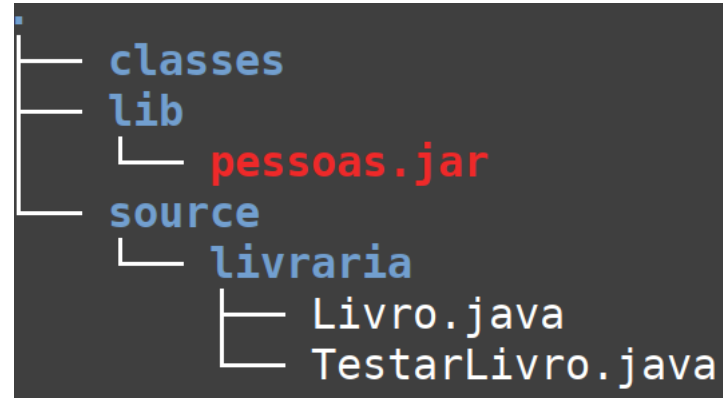
Classes “Pessoa”  
disponíveis como  
biblioteca

ex07



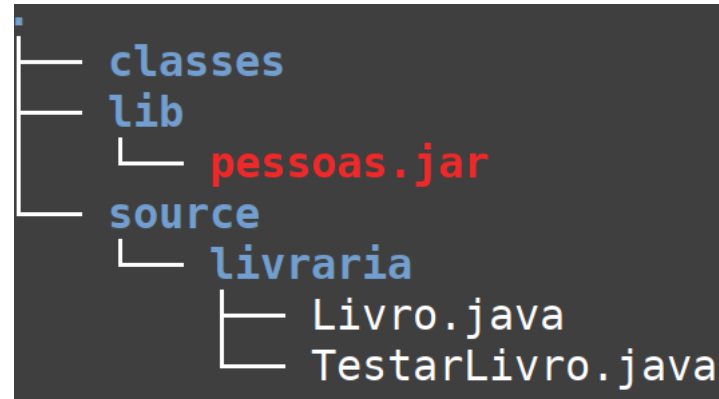
# Empacotando como biblioteca

“Usando a biblioteca”



# Empacotando como biblioteca

“Usando a biblioteca”



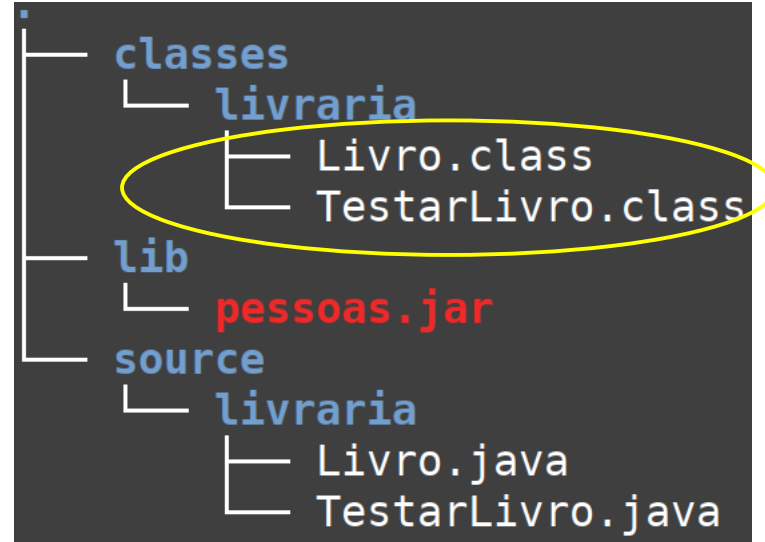
## Compilação da livraria:

```
/source$ javac -d ../classes -cp ../lib/pessoas.jar livraria/*.java
```



# Empacotando como biblioteca

“Usando a biblioteca”



# Empacotando como biblioteca

## “Usando a biblioteca”

“-cp” => Classpath:  
\* arquivo .jar (../lib/)  
\* diretório atual (.)

*Linux usa dois pontos (:)*  
*Windows usa ponto-e-vírgula (;)*



## Execução:

```
/classes$ java -cp ../lib/pessoas.jar:. livraria.TestarLivro
```

ex08



# Java Web Start

# Java web start

Usuário clica  
em link para  
arquivo .jnlp

# Java web start

Usuário clica  
em link para  
arquivo .jnlp



Java web start inicia  
aplicativo auxiliar,  
que lê o jnlp

# Java web start

Usuário clica  
em link para  
arquivo .jnlp



Java web start inicia  
aplicativo auxiliar,  
que lê o jnlp



App auxiliar  
solicita ao servidor  
web o arquivo .jar

# Java web start

Usuário clica  
em link para  
arquivo .jnlp



Java web start inicia  
aplicativo auxiliar,  
que lê o jnlp



App auxiliar  
solicita ao servidor  
web o arquivo .jar



Java web  
start inicia  
o .jar obtido

# Java web start



Descontinuado a partir do Java 9

Support for Deployment Technology and JavaFX*				
Version	GA Date	Java Web Start Support Until	Java Plugin (Applets) Support Until	JavaFX Support Until
6	December 2006	October 2017	October 2017	N/A
7	July 2011	October 2017	October 2017	July 2019
8	March 2014	December 2030	March 2019	March 2025
9 and later		N/A	N/A	N/A

<https://www.oracle.com/java/technologies/java-se-support-roadmap.html>





# Java web start

×  

[Todas](#) [Notícias](#) [Imagens](#) [Vídeos](#) [Shopping](#) [Mais](#) Fr

Aproximadamente 32.900 resultados (0,26 segundos)

 Resultados para **Água Verde, Blumenau** · [Escolher região](#) ⋮

 **openwebstart.com**  
<https://openwebstart.com> · [Traduzir esta página](#) ⋮

**Java Web Start is dead. Long live OpenWebStart ...**

This is why we decided to create **OpenWebStart**, an open source reimplementation of the Java Web Start technology. Our replacement provides the most commonly used ...

**Download**  
User Guide - Details - Support - FAQs - ...

**Details**

# Java web start

Alternativa...



# Java web start

## OpenWebStart

**Latest stable release: 1.9.1**

### Windows

Compatibility:

Windows 10 or higher (\*)

OpenWebStart\_windows-  
x64\_1\_9\_1.exe

OpenWebStart\_windows-  
x32\_1\_9\_1.exe

### macOS

Compatibility:

macOS 10.15 (Catalina) or  
higher (\*)

OpenWebStart\_macos-  
aarch64\_1\_9\_1.dmg

OpenWebStart\_macos-  
x64\_1\_9\_1.dmg

### Linux

Compatibility:

Ubuntu 18.04 LTS or higher (\*)

OpenWebStart\_linux\_1\_9\_1.deb

# Java web start

## OpenWebStart

### OpenWebStart support options

Java and your operating system get updates, and so does OpenWebStart. Thanks to the support of our awesome sponsors, OpenWebStart will be updated and further developed as long as needed.

You can of course use OpenWebStart as an open source solution free of charge but we'd also like to talk to you about support options to guarantee the lasting development.

#### Community

**free**

Community usage -  
Devs help devs

Public bug fixes  
only

#### Developer

**1200<sup>EUR</sup>**

Small companies  
up to 5 developers.  
Price per  
developer and  
year.

#### Basic

**6000<sup>EUR</sup>**

Companies with  
more than 5  
developers. Price  
per company and  
year.

#### Premium

**30000<sup>EUR</sup>**

Companies which  
need fixed SLA's.  
Price per company  
and year.

# Java web start

## OpenWebStart

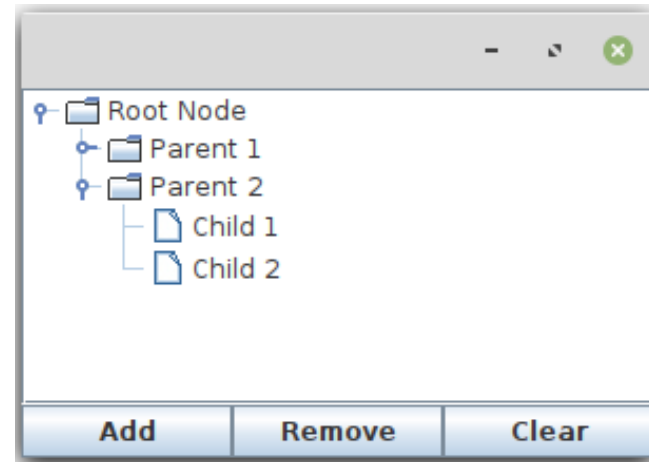
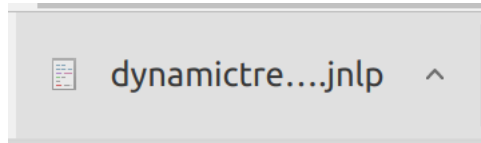
Example (click on link to view example)	Zip File (contains all source files necessary for the example plus NetBeans IDE project metadata)	Where Described
<a href="#">Dynamic Tree Demo</a>	<a href="#">webstart_ComponentArch_DynamicTreeDemo</a>	<a href="#">Developing a Java Web Start Application</a> <a href="#">Deploying a Java Web Start Application</a>

<https://docs.oracle.com/javase/tutorial/deployment/webstart/examplesIndex.html>

# Java web start

## Alternativa...

Example (click on link to view example)	Zip File (contains all source files necessary for the example plus NetBeans IDE project metadata)	Where Described
<a href="#">Dynamic Tree Demo</a>	<a href="#">webstart_ComponentArch_DynamicTreeDemo</a>	<a href="#">Developing a Java Web Start Application</a> <a href="#">Deploying a Java Web Start Application</a>



# Java web start

Overview .....	4
Release Cadence and Licensing.....	4
Java Deployment Roadmap.....	5
Applets.....	5
Java Web Start and Pre-Installed Java Runtime Environments.....	5
Java UI.....	6
JavaFX .....	6
Swing and AWT .....	6

## ***Java client roadmap***

<https://www.oracle.com/technetwork/java/javase/javaclientroadmapupdatev2020may-6548840.pdf>

# Java web start

Oracle has been encouraging application developers and users to transition away from Web Start and encouraging non-commercial consumers to remove any unused or non-supported Oracle JRE installations from their desktops.

<https://www.oracle.com/technetwork/java/javase/javaclientroadmapupdatev2020may-6548840.pdf>



# FIM

# Obrigado!



**INSTITUTO FEDERAL**  
Catarinense  
Campus Blumenau



*Capítulo 17*

Pacotes, arquivos jar e implantação

<https://github.com/hvescovi/dw2ed/tree/main/fund/java/pacote>

