

Universidade Federal de Ouro Preto

CSI032 - Programação de Computadores II

Organização de Códigos

Professor: Dr. Rafael Frederico Alexandre

Contato: rfalexandre@decea.ufop.br

Colaboradores: Renan Saldanha Contatos:

renansaldlinhares@gmail.com

ICEA



Instituto de Ciências Exatas e
Aplicadas - Campus João Monlevade



UFOP

Universidade Federal
de Ouro Preto

Agenda

- 1 Pacotes
- 2 Ferramentas
- 3 Dicas de estudo

Agenda

- 1 Pacotes
- 2 Ferramentas
- 3 Dicas de estudo

Pacotes

- Pacotes são utilizados para organizar as classes semelhantes;
- A grosso modo, são apenas pastas ou diretórios do sistema operacional onde ficam armazenados os arquivos fonte de Java;
- Essenciais para o conceito de encapsulamento, no qual são dados níveis de acesso as classes (Modificadores de acesso);
- Para definir que determinada classe pertence a um pacote deve ser utilizada a palavra reservada `package` e o nome do pacote ao qual a classe pertence.

Pacotes

```
package br.ufop;  
  
public class Cliente {  
  
}
```

Figura: Exemplo de classe pertencente ao pacote br.ufop

Pacotes

- Agora para utilizar uma classe em um pacote diferente é necessário referenciar todo o seu pacote antes de utilizá-la;
- Gera um código maior e mais difícil para leitura e interpretação de outras pessoas que venham a fazer manutenção no código;

Sintaxe

```
nomeCompletoDoPacote variável = new  
nomeCompletoDoPacote();
```

Pacotes

Exemplo de utilização de classes de pacotes diferentes



The screenshot shows an IDE with two tabs: `Cliente.java` and `Main.java`. The `Main.java` file contains the following code:

```
1 package br.ufop.teste;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         br.ufop.Cliente cliente = new br.ufop.Cliente("João", 23, "123456789");
7         System.out.println("Nome: " + cliente.getNome()+
8             " Idade: "+cliente.getIdade()+" CPF: "+cliente.getCpf());
9     }
10 }
11
12 }
13
```

The console output at the bottom shows the result of the program execution:

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jdk1.8.0_221\bin\javaw.exe (8 de set de 2019 14:58:51)
Nome: João Idade: 23 CPF: 123456789
```

Figura: Exemplo de utilização de classes em pacotes diferentes

Pacotes

Import.

- Como reduzir a escrita?
 - 1 Para utilizar classes de um determinado pacote em uma classe de outro pacote, sem a necessidade de fazer toda a referencia ao pacote quando for declarar uma variável reference da classe, é necessário fazer o import da classe;

Pacotes

Import.

- Como reduzir a escrita?
 - 1 Para utilizar classes de um determinado pacote em uma classe de outro pacote, sem a necessidade de fazer toda a referencia ao pacote quando for declarar uma variável reference da classe, é necessário fazer o import da classe;
 - 2 A importação da classe é feita pela palavra reservada **import**;

Pacotes

Import.

- Como reduzir a escrita?
 - ① Para utilizar classes de um determinado pacote em uma classe de outro pacote, sem a necessidade de fazer toda a referencia ao pacote quando for declarar uma variável reference da classe, é necessário fazer o import da classe;
 - ② A importação da classe é feita pela palavra reservada **import**;
 - ③ Para que o pacote seja importado para uma classe em outro pacote, o modificador de acesso da classe deve está como public.

Pacotes

Import.

- Como reduzir a escrita?
 - ① Para utilizar classes de um determinado pacote em uma classe de outro pacote, sem a necessidade de fazer toda a referencia ao pacote quando for declarar uma variável reference da classe, é necessário fazer o import da classe;
 - ② A importação da classe é feita pela palavra reservada **import**;
 - ③ Para que o pacote seja importado para uma classe em outro pacote, o modificador de acesso da classe deve está como public.

Sintaxe

```
import nomeCompletoDoPacote.classeASerImportada;
```

Pacote

Exemplo import



```
1 package br.ufop.teste;
2
3 import br.ufop.Cliente;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         Cliente cliente = new Cliente("João", 23, "123456789");
9         System.out.println("Nome: " + cliente.getNome()+
10             " Idade: "+cliente.getIdade()+" CPF: "+cliente.getCpf());
11     }
12 }
13
14 }
15 }
```

Problems @ Javadoc Declaration Console

<terminated> Main (1) [Java Application] C:\Program Files\Java\jdk1.8.0_221\bin\javaw.exe (8 de set de 2019 15:07:09)

Nome: João Idade: 23 CPF: 123456789

Figura: Exemplo de utilização do Import

Pacote

Boas práticas de programação

Organização

É muito importante manter a ordem! Primeiro, aparece uma (ou nenhuma) vez o package; depois, pode aparecer um ou mais import's; e, por último, as declarações de classes.

Curinga [?]

É possível "importar um pacote inteiro" (todas as classes do pacote, exceto os subpacotes) através do coringa * : (import java.util.*). Importar todas as classes de um pacote não implica em perda de performance em tempo de execução, mas pode trazer problemas com classes de mesmo nome! Além disso, importar de um em um é considerado boa prática, pois facilita a leitura para outros programadores.

Pacote

Exercícios

- ① Crie um sistema que calcule a comissão para funcionários de um determinado tipo de comércio com as seguintes condições:
 - O comércio possui gerente e vendedor;
 - Os vendedores recebem uma comissão de 10% sobre vendas totais no mês até de R\$ 1000,00, 15% para vendas acima de R\$ 1000,00 e menores que R\$ 5000,00 e 20% para vendas acima de R\$ 5000,00;
 - O gerente recebe ao final uma comissão de 5% sobre o salário;
 - São vendidos três tipos de produtos, cada produto deve conter nome, preço;
 - Todos os funcionarios devem ter nome, registro, salário base e devem conter um método que retorne o salário com a comissão;

Agenda

- 1 Pacotes
- 2 Ferramentas
- 3 Dicas de estudo

Ferramentas Java

- 1 Uma aplicação em java tem uma quantidade muito grande de classes, como enviar essa quantidade de classes para o cliente de maneira prática?

Ferramentas Java

- 1 Uma aplicação em java tem uma quantidade muito grande de classes, como enviar essa quantidade de classes para o cliente de maneira prática?
- 2 Compactar todos os arquivos em um arquivo só;

Ferramentas Java

- 1 Uma aplicação em java tem uma quantidade muito grande de classes, como enviar essa quantidade de classes para o cliente de maneira prática?
- 2 Compactar todos os arquivos em um arquivo só;
- 3 O formato de compactação padrão é o **zip** com extensão **.jar**

Ferramentas Java

JAR

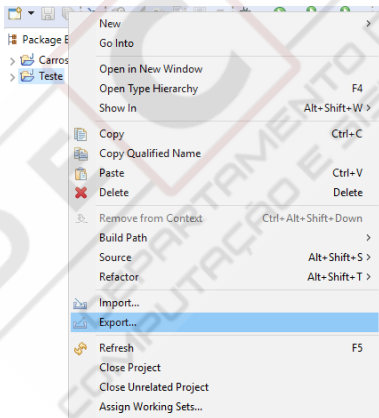
Conceito [?]

O arquivo jar ou Java ARchive, possui um conjunto de classes (e arquivos de configurações) compactados, no estilo de um arquivo zip. O arquivo jar pode ser criado com qualquer compactador zip disponível no mercado, inclusive o programa jar que vem junto com o JDK.

Ferramentas Java

Criando um arquivo .jar

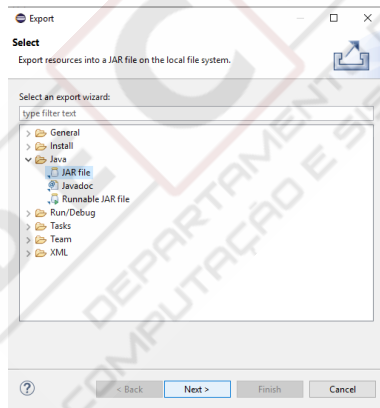
Clique com o botão direito em cima do nome do seu projeto e selecione a opção Export.



Ferramentas Java

Criando um arquivo .jar

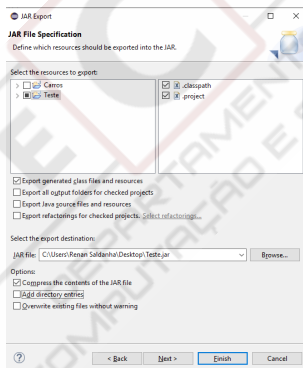
Na tela Export, selecione a opção "JAR file" e aperte o botão "Next".



Ferramentas Java

Criando um arquivo .jar

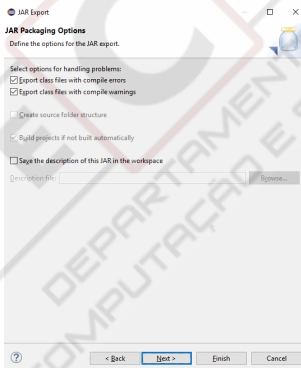
Na opção "JAR file:", selecione o local que deseja salvar o arquivo JAR. E aperte "Next".



Ferramentas Java

Criando um arquivo .jar

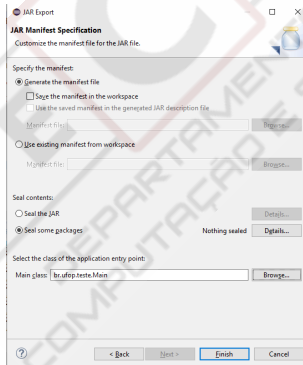
Na tela abaixo, simplesmente clique em next, pois não há nenhuma configuração a ser feita.



Ferramentas Java

Criando um arquivo .jar

Na tela abaixo, na opção "select the class of the application entry point", você deve escolher qual classe será a classe que vai rodar automaticamente quando você executar o JAR. imagem.



Ferramentas Java

Criando um arquivo .jar

É comum dar um nome mais significativo aos JARs, incluindo nome da empresa, do projeto e versão.

Exemplos de arquivos .jar

hibernate-jpamodelgen-5.4.4.jar

ufop-sistema-de-matricula-1.0.0.jar

Ferramentas Java

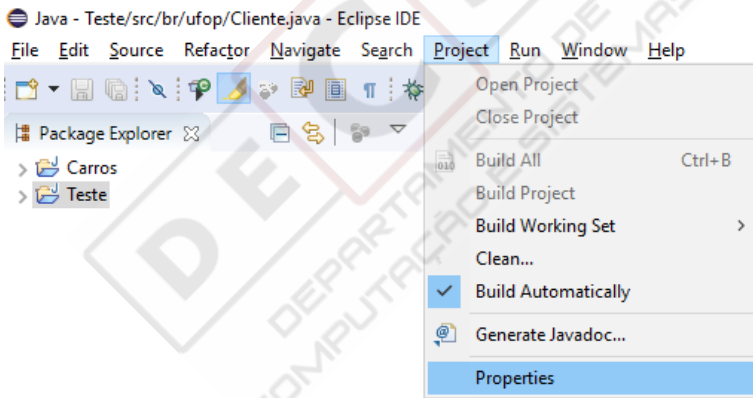
Exercício

- 1 Gere um jar do exercício anterior com o arquivo de manifesto.

Ferramentas Java

Importando um arquivo .jar

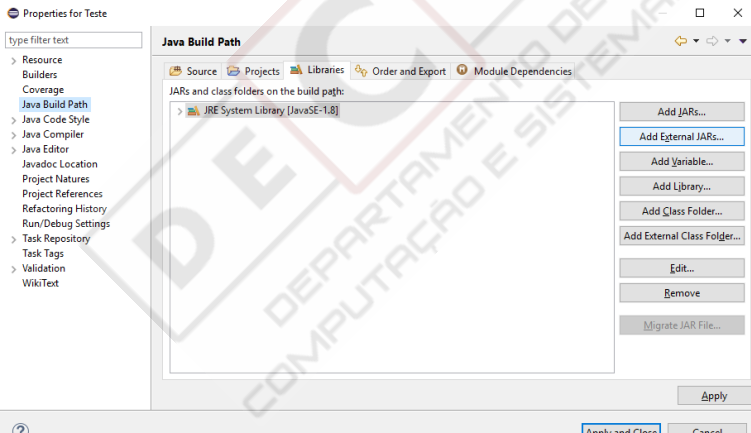
Vá ao menu Project selecione a opção Properties.



Ferramentas Java

Importando um arquivo .jar

Selecionar a opção Java Build Path, depois selecionar a aba Libraries e, finalmente, clicar no botão Add External Jars...



Ferramentas Java

javadoc

- 1 Como vamos saber o que cada classe tem no Java?
- 2 Utilizando a Documentação do Java, também referido como javadoc;
- 3 <https://docs.oracle.com/javase/8/docs/api/>

Ferramentas Java

javadoc

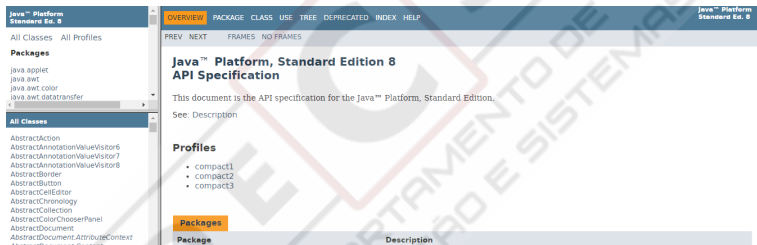
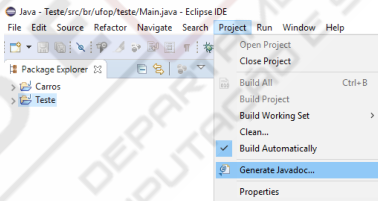


Figura: Página de consulta da documentação java.

Ferramentas Java

Gerando um javadoc

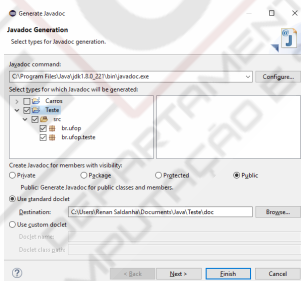
Na barra de menu, selecione o menu Project, depois a opção "Generate Javadoc...". (apenas disponível se estiver na perspectiva Java, mas você pode acessar o mesmo wizard pelo export do projeto)



Ferramentas Java

Gerando um javadoc

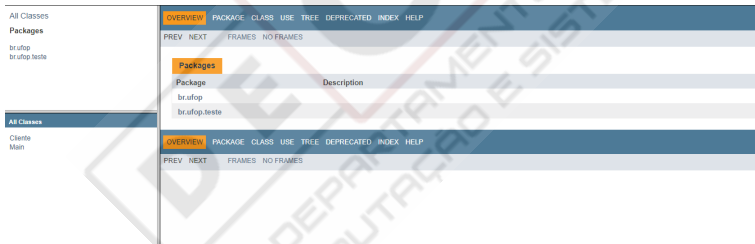
Em seguida, aparecerão as opções para gerar a documentação do seu sistema, selecione todas as classes do seu sistema e deixe as outras opções como estão.



Ferramentas Java

Gerando um javadoc

Abra a documentação através do caminho que você marcou e abra o arquivo index.html.



Ferramentas Java

Adicionando comentários no javadoc

- 1 Os comentários são adicionados ao código sobre forma de comentário;
- 2 O texto deve ser aberto por `/**` e fechado com `*/`;
- 3 Cada linha deve iniciar com `*`

Exemplo de comentário de documentação

```
/**  
 * Isso é um comentário de documentação  
 */
```

Ferramentas Java

Informações no javadoc

Também podemos definir outras informações neste texto, como:

- autor: Deve ser utilizada a tag @author, é utilizada para especificar o autor da classe ou do método em questão.;
- versão: Deve ser utilizada a tag @version, é utilizada para indicar a versão da classe;
- parâmetros: Deve ser utilizada a tag @param, é utilizada para indicar os parâmetros que devem ser passados para um método;
- retorno: Deve ser utilizada a tag @return, é utilizada para mostrar o tipo de retorno de um método;
- throws: Deve ser utilizada a tag @throws, é utilizada para indicar as possíveis exceções de um método.

Ferramentas Java

Exemplo de comentários

```
1 package br.ufop;
2
3 /**
4  *
5  * Classe responsável por representar um cliente
6  * @author Renan Saldanha
7  * @version 0.1.0
8  */
9
10 public class Cliente {
11     private String nome;
12     private int idade;
13     private String cpf;
14
15     public Cliente(String nome, int idade, String cpf) {}
16
17     /**
18      *
19      * Método que retorna o nome de um cliente
20      * @param nome
21      * @return String
22      */
23     public String getNome() {
24         return nome;
25     }
26
27     /**
28      *
29      * Método para adicionar ou editar o nome do cliente
30      * @param nome
31      */
32     public void setNome(String nome) {
33         this.nome = nome;
34     }
35 }
```

Figura: Exemplo de classe com comentário para javaDoc.

Ferramentas Java

Exercício

- 1 Gere o javadoc do exercício feito anteriormente. Coloque todas as informações que julgarem pertinentes.

Agenda

- 1 Pacotes
- 2 Ferramentas
- 3 Dicas de estudo

Pesquisa

Classes

Agora que você já sabe onde encontrar a documentação da linguagem java faça uma pesquisa das seguintes classes:

- A classe Scanner do pacote java.util
- A classe ArrayList do pacote java.util

A resolução dos exercícios dessa aula estão disponíveis em <https://github.com/renansald/TutoriaProgII>



**MUITO
OBRIGADO!!!**

DECSI
DEPARTAMENTO DE
COMPUTAÇÃO E SISTEMAS

Referncias Bibliograficas I

