Padronização do Código

- PADRONIZE e DOCUMENTE seu código!
  - ✓ O maior beneficiado será você mesmo
    - Diferenças no estilo de programação dificultam a compreensão do código
  - ✓ A qualidade do software passa pela **padronização** do código fonte
    - Padronização de código é cada vez mais utilizada nas empresas
    - Conta pontos nas certificações
  - ✓ Um código **bem documentado** possibilita uma manutenção mais eficiente
    - A quantidade de informação varia de acordo com a necessidade, ao longo do código

- Orientações sobre padronização e documentação do código
  - ✓ ***Code Conventions for the Java Programming Language***
    - http://www.oracle.com/technetwork/java/codeconv-138413.html

Padronização do Código

- **Documentação**: no início de **todo código fonte**
  - ✓ Utilize o padrão abaixo
  - ✓ Informações: autor, matrícula, data início, data da última alteração, nome do programa, função do código
  - ✓ Não acentue os comentários!

```
/* ***************************************************************
* Autor............: nome(s) do(s) autor(es) do codigo
* Matricula........: numero de matricula(s)
* Inicio...........: data de inicio da codificacao
* Ultima alteracao.: data da ultima alteracao realizada no codigo
* Nome.............: Nome do programa
* Funcao...........: descricao do que eh o programa
*************************************************************** */
int main (int argc, char* argv[]) {
  if (valor==10) { //valor eh o tamanho da mensagem recebida
    //faça algo aqui
  }//fim do if
}//fim do programa principal
```

Padronização do Código

- **Documentação**: no início de **toda sub-rotina/método**
  - ✓ Utilize o padrão abaixo, contendo: nome do método, função, parâmetros recebidos, valor retornado
  - ✓ Pode-se utilizar também o JavaDoc

```
/* ************************************************************
* Metodo: nome do metodo
* Funcao: testa os valores para transmissao
* Parametros: descricao dos parametros recebidos
* Retorno: descricao do valor retornado
************************************************************ */
void testaValor (void) {
  if (valor==10) { //valor eh o tamanho da mensagem recebida
    //faça algo aqui
  }//fim do if
  else {
    //faça outra coisa aqui
  }//fim do else
}//fim do metodo testaValor
```

Padronização do Código

- **Padronização**: identação
  - ✓ Não deve ser por tabulação e sim espaçada
  - ✓ Utilizar 2 (dois) espaços

```
/* ************************************************************
* Metodo: comparaValores
* Funcao: compara os valores para transmissao
* Parametros: descricao dos parametros recebidos
* Retorno: descricao do valor retornado
************************************************************ */
void comparaValores (void) {
  if (valor==10) { //valor eh o tamanho da mensagem recebida
    //faça algo aqui
  }//fim do if
  else {
    //faça outra coisa aqui
  }//fim do else
}//fim do metodo comparaValores
```

Padronização do Código

- **Padronização**: variáveis
  - ✓ Utilizar iniciais em letra minúscula e os espaços são letras maiúsculas
  - ✓ Mesmo para acrônimos como IP, usar ip ou Ip
  - ✓ Evitar a abreviação dos nomes das variáveis

```
/*****************************************************************
* Metodo: testaErro
* Funcao: testa a corretude dos valores para transmissao
* Parametros: mensagemEnviada=texto contendo a mensagem que sera
              testada
* Retorno: void
***************************************************************** */
void testaErro (int mensagemEnviada) {
  static final int TAXA_DE_ERRO = 60;
  int propagacaoDoErro
  propagacaoDoErro = probabilidadeGeral;
  //resto do codigo aqui
}//fim do metodo testaErro
```

Padronização do Código

- **Padronização**: constantes
  - ✓ Palavras em letras maiúsculas e separadas pelo *underscore* (“_”)

```
/*****************************************************************
* Metodo: testaErro
* Funcao: testa a corretude dos valores para transmissao
* Parametros: mensagemEnviada=texto contendo a mensagem que sera
              testada
* Retorno: void
***************************************************************** */
void testaErro (int mensagemEnviada) {
  static final int TAXA_DE_ERRO = 60;
  int propagacaoDoErro
  propagacaoDoErro = probabilidadeGeral;
  //resto do codigo aqui
}//fim do metodo testaErro
```

Padronização do Código

- **Padronização**: métodos
  - ✓ Mesmas regras de nomes de variáveis

```
/*************************************************************
* Metodo: cadastroUnico
* Funcao: calcula ano de ingresso no curso a partir do numero
*         de matricula do aluno
* Parametros: matriculaAluno eh o numero de matricula do aluno
* Retorno: void
************************************************************* */
void ClasseAluno::cadastroUnico (int matriculaAluno) {
  matricula = matriculaAluno;
  anoIngresso = ano;
  //resto do codigo aqui
}//fim do metodo cadastroUnico
```

Padronização do Código

- **Padronização**: classes
  - Devem começar com letras maiúsculas e sem espaços

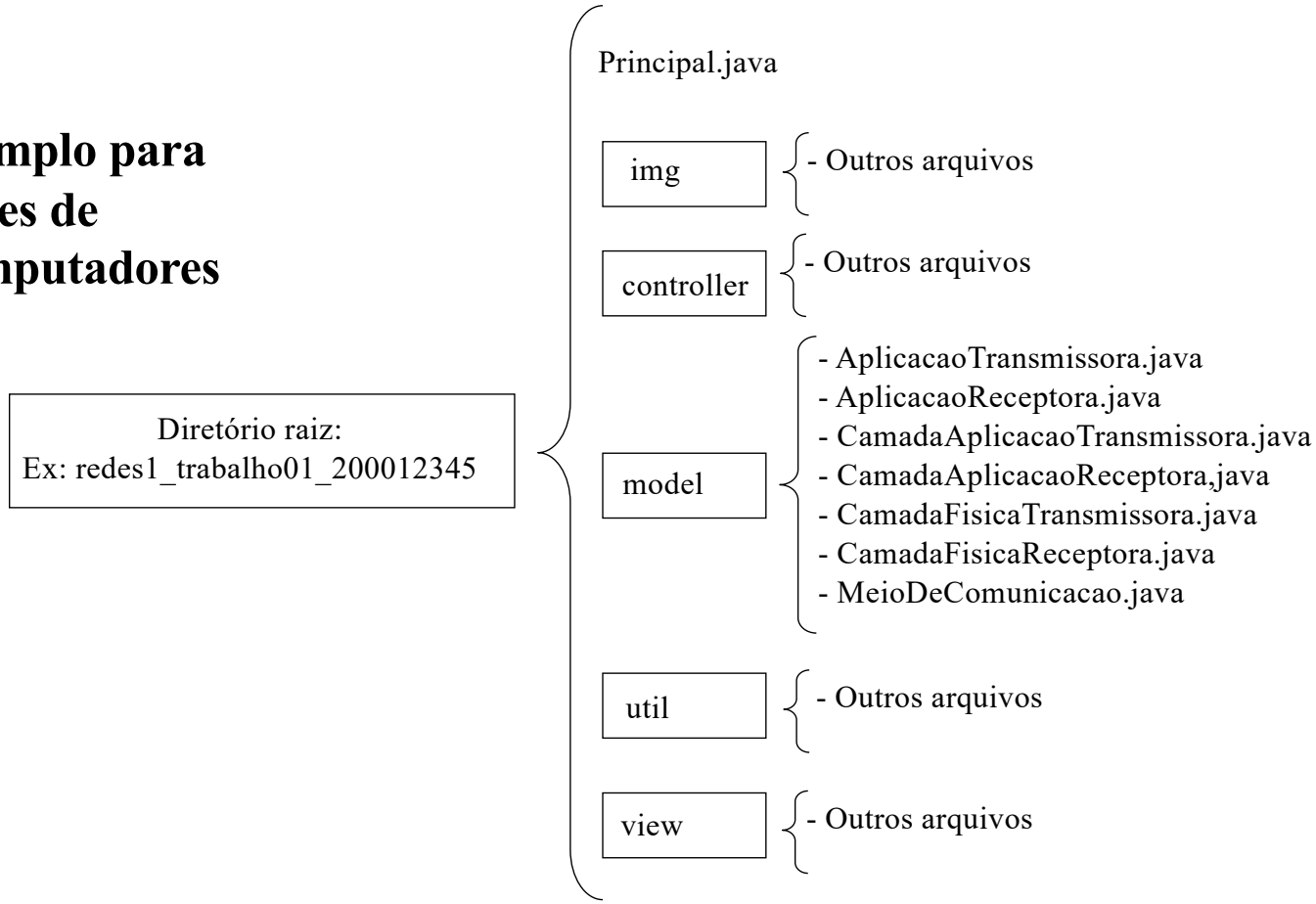
```
/*************************************************************
* Metodo: cadastroUnico
* Funcao: calcula ano de ingresso no curso a partir do numero
*         de matricula do aluno
* Parametros: matriculaAluno eh o numero de matricula do aluno
* Retorno: void
************************************************************* */
void ClasseAluno::cadastroUnico (int matriculaAluno) {
  matricula = matriculaAluno;
  anoIngresso = ano;
  //resto do codigo aqui
}//fim do metodo cadastroUnico
```

Padronização do Código

➪ **Padrão de Projeto**: caso utilize, opte pelo MVC

✓ Caso contrário, colocar todos os arquivos no diretório raiz

Exemplo para Redes de Computadores

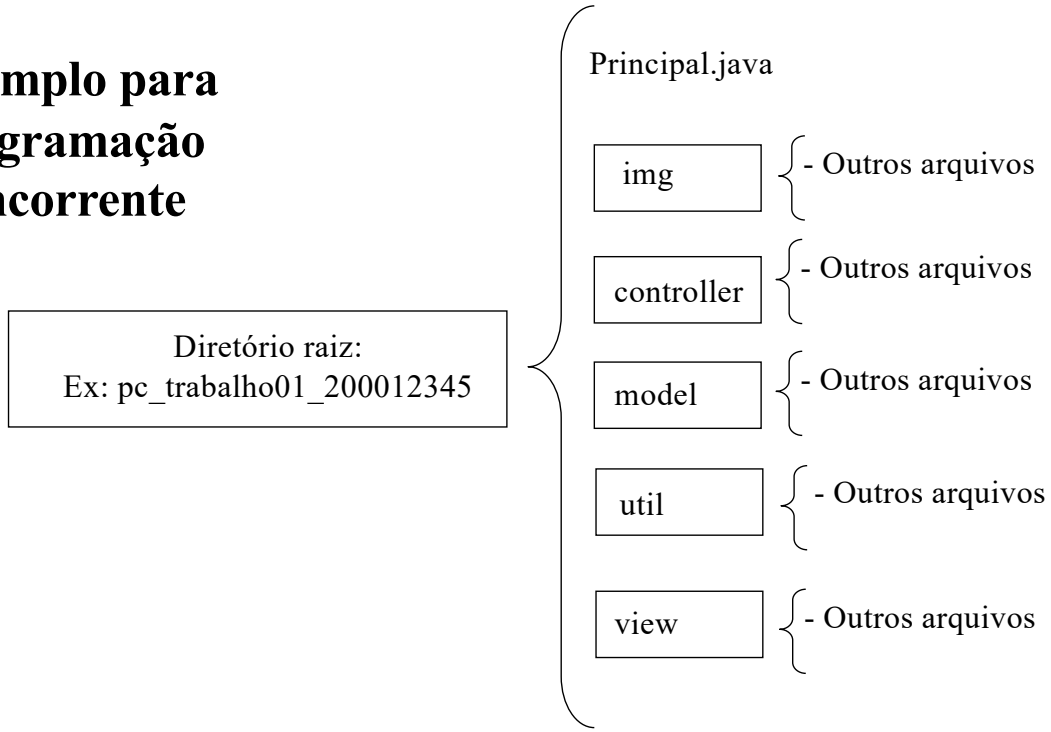


Padrão de Projeto

➪ Caso utilize, opte pelo MVC (veja abaixo)

✓ Caso contrário, colocar todos os arquivos no diretório raiz

Exemplo para Programação Concorrente



Padronização do Código

- RESUMO: o que será obrigatório
  - ✓ Comentário no inicio de todo código fonte (**conforme padronizado**)
  - ✓ Caso utilize um padrão de projeto, opte pelo MVC (**conforme padronizado**)
  - ✓ Identação do código fonte (**conforme padronizado**)

- **Contudo, independente do exigido pratique as boas técnicas de programação!**

```
/* ***************************************************************
* Autor............: nome(s) do(s) autor(es) do codigo
* Matricula........: numero de matricula(s)
* Inicio...........: data de inicio da codificacao
* Ultima alteracao.: data da ultima alteracao realizada no codigo
* Nome.............: Nome do programa
* Funcao...........: descricao do que eh o programa
*************************************************************** */
```