Pacotes e Encapsulamento

Prof. Bruno Gomes

bruno.gomes@ifrn.edu.br

Programação Orientada a Objetos

© TemplatesWise.com

Introdução

- Permite o agrupamento de classes em uma coleção chamada *pacote*
- Um pacote é uma coleção de classes e interfaces que provem proteção de acesso e gerenciamento de espaços de nomes
- Organiza o trabalho e separa seu trabalho das bibliotecas de código
- A biblioteca Java é distribuída em pacotes (Ex.: *java.util*, *java.lang*)
- Todos os pacotes Java estão dentro das hierarquias de pacote *java* e *javax*

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Introdução

- Objetivos:
  - Garantir a singularidade dos nomes de classe (Evita conflito de nomes)
  - Facilidade de determinar classes relacionadas
  - Proteção de acesso
- Sun recomenda que utilize o nome de domínio da Internet da empresa, para garantir um nome de pacote único

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Classes ficam em pacotes

- As classes ficam em pacotes
  - o pacote faz parte do nome da classe
    - java.util.Scanner
    - java.util.ArrayList
  - Alguns pacotes padrão do JAVA
    - java.lang: classes fundamentais – importado automaticamente
    - java.util: classes utilitárias
    - java.io: classes para entrada e saída
    - java.net: classes para uso em rede (TCP/IP)
    - ... e muito mais!

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Pacotes

- Para usar uma classe que pertence a um outro pacote é necessário usar o nome completo:

```
java.util.Scanner sc = new java.util.Scanner(System.in);
```

- ou importar a classe:

```
import java.util.Scanner;
...
Scanner sc = new Scanner();
```

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Pacotes

- Classes do pacote **java.lang** são importadas por padrão:
  - **String, StringBuilder, Integer, Double, System, entre outras**

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Localização de classes

- A **JVM precisa saber onde** encontrar uma determinada classe
  - Classe está em um arquivo **.class**
  - A máquina virtual não procura no sistema de arquivos inteiro
  - Classes da **API padrão estão em um** lugar fixo e a **JVM sabe onde** encontrar (instalação da JVM)
  - Precisamos informar onde estão nossas classes

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

CLASSPATH

- A JVM procura no CLASSPATH
  - opção -cp da chamada a máquina virual
    - java -cp c:\classes Classe (Windows)
    - java -cp /home/aluno/classes (Unix)
  - ... ou definição da variável de ambiente CLASSPATH
    - set CLASSPATH=c:\classes (Windows)
    - export CLASSPATH=/home/aluno/classes (Unix)
  - Diretório classes deverá conter os arquivos com as classes (.class)
    - pacotes são diretórios

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Classes em pacotes

- No arquivo que escrevemos o código fonte:
  - É preciso informar em qual pacote a classe esta
    - no início do arquivo que define a classe:
      - package nomePacote;
      - Exemplo: package br.ifrn.tads.poo;
    - Se não especificado classe pertence ao pacote default (padrão), que não tem nome
      - Desaconselhável usar pacote padrão

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Classes em pacotes

- Inserir os arquivos fontes em um sub-diretório que corresponde ao nome de pacote completo
- Exemplo:
  - Classe no pacote ifrn.tads.poo, deve estar no subdiretório ifrn/tads/poo

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Classes em pacotes

- Considere a classe Racional
  - pacote ifrn.tads.poo;

```
package ifrn.tads.poo;

public class Racional{
...
}
```

- Ao ser compilada, o arquivo .class deverá ser colocado no diretório ifrn/tads/poo da raiz do CLASSPATH
  - C:\classes\ifrn\tads\poo por exemplo

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Compilação

- Opção -d do compilador informa raiz do diretório onde a classe será colocada
  - javac -d c:\classes Racional.java
  - Este comando compila o arquivo Racional.java e coloca o **.class no diretório** correspondente ao pacote a partir de c:\classes
    - c:\classes\ifrn\tads\poo\Racional.class
- – Se o CLASSPATH estiver definido para c:\classes a JVM saberá onde encontrar a classe Racional

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Usando Pacotes

```
import ifrn.tads.poo.Racional;
public class Main{
...
Racional r1 = new Racional();
...
}
```

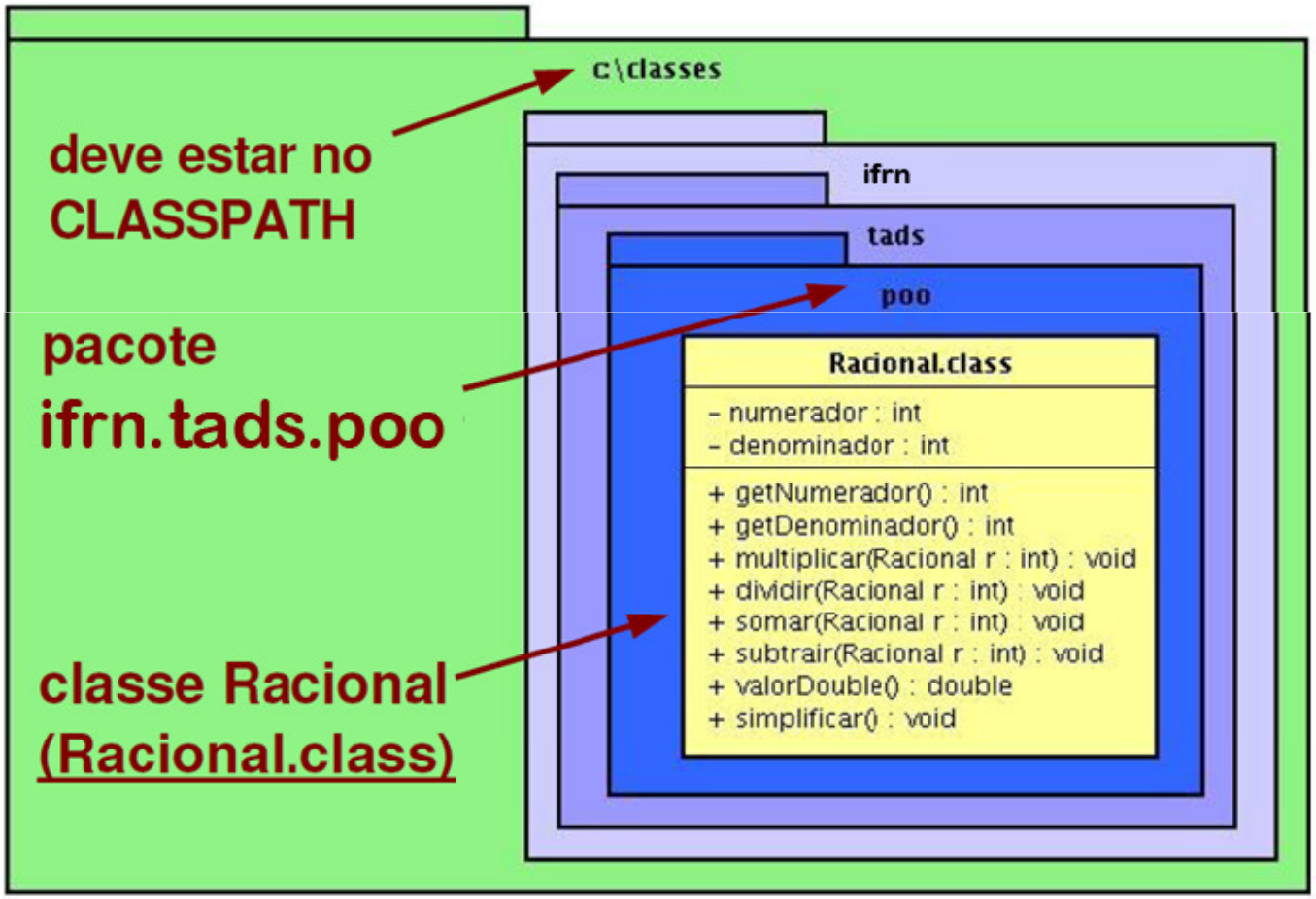
Importando

Nome Completo

```
public class Main{
...
ifrn.tads.poo.Racional r1;
r1 = new ifrn.tads.poo.Racional();
...
}
```

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Pacotes



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

ENCAPSULAMENTO

Projeto Orientado a Objetos

- **Objetivos:**
  - Robustez
    - Sistemas confiáveis, tolerante a falhas
  - Adaptabilidade
    - Capacidade de reagir conforme o contexto
  - Reutilizabilidade
    - Reutilização do software

- **Princípios:**
  - Abstração
  - Encapsulamento
  - Modularidade

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Elementos do modelo de objetos

- Abstração
  - Decompor um sistema complicado em suas partes fundamentais
  - Descrevê-las em uma linguagem simples e precisa
    - Atribuir-lhes um nome e descrever suas funcionalidades
  - Abstrair os detalhes desnecessários
- Encapsulamento
  - É o processo de esconder todos os detalhes de um objeto que não contribuem para suas características essenciais

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Visibilidade

- Proteção de acesso
  - Proteger o interior da classe
- Explicitar o que usuários (da classe) precisam saber
- Pode ser:
  - **private:** Apenas membros da classe têm acesso
  - **protected:** Membros da classe e subclasses
  - **public:** Todos têm acesso
  - **default:** Apenas membros do mesmo pacote

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Proteção de Acesso

- Atributos que fazem parte da implementação
  - Declare-os como **private**
- Nem todos os métodos fazem parte da interface
  - Métodos que servem para auxiliar outros métodos
    - Declare-os **private**
- Deixe **public** apenas o que o cliente deve saber

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Proteção de Acesso

- Quando utilizamos atributos privados (private), o acesso a esses membros se dá através dos **métodos acessadores** (getters e setters):
- GET
  - Verifica um valor em algum campo ou atributo de uma classe
- SET
  - Modificar um valor em algum campo ou atributo de uma classe

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Proteção de Acesso

- Observação:

Criar os métodos GET ou SET somente quando for precisar!!!

Proteção de Acesso

- Exemplo:

```
public class Pessoa {
        private String nome;
        private int cpf;

        public Pessoa(String nome, int cpf) {
                this.nome = nome;
                this.cpf = cpf;
        }

        public String getNome() {
                return nome;
        }
        public void setNome(String nome) {
                this.nome = nome;
        }
        public int getCpf() {
                return cpf;
        }
        public void setCpf(int cpf) {
                this.cpf = cpf;
        }
}
```

Método toString()

- Serve para adicionar um texto que irá representar um objeto
- Se tentar imprimir diretamente um objeto, irá ser impresso as coordenadas do objeto
- Sintaxe:

```
public String toString(){
    return "texto";
}
```

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Método toString()

- Façam um teste:
  - Crie uma Classe Pessoa, com um atributo nome (String).
  - Na mesma classe, crie um método main, e crie um objeto da Classe Pessoa.
  - Imprima este objeto, sem acessar nenhum método

```
Pessoa pessoa = new Pessoa();
System.out.println(pessoa);
```

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Exemplo:

```
public class Pessoa {
        private String nome;
        private int cpf;

        public Pessoa(String nome, int cpf) {
                this.nome = nome;
                this.cpf = cpf;
        }

        public String getNome() {
                return nome;
        }
        public void setNome(String nome) {
                this.nome = nome;
        }
        public int getCpf() {
                return cpf;
        }
        public void setCpf(int cpf) {
                this.cpf = cpf;
        }

        public String toString(){
                return "Nome: "+nome+"\nCPF: "+cpf;
        }
}
```

Exercício

- Colocar os atributos como private, criar os métodos acessadores e o método toString para as Classes do exercício da aula passada:
  - Endereco
  - Pessoa
  - Produto
- Criar os pacotes *ifrn.tads.modelo* e *ifrn.tads.negocio*

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

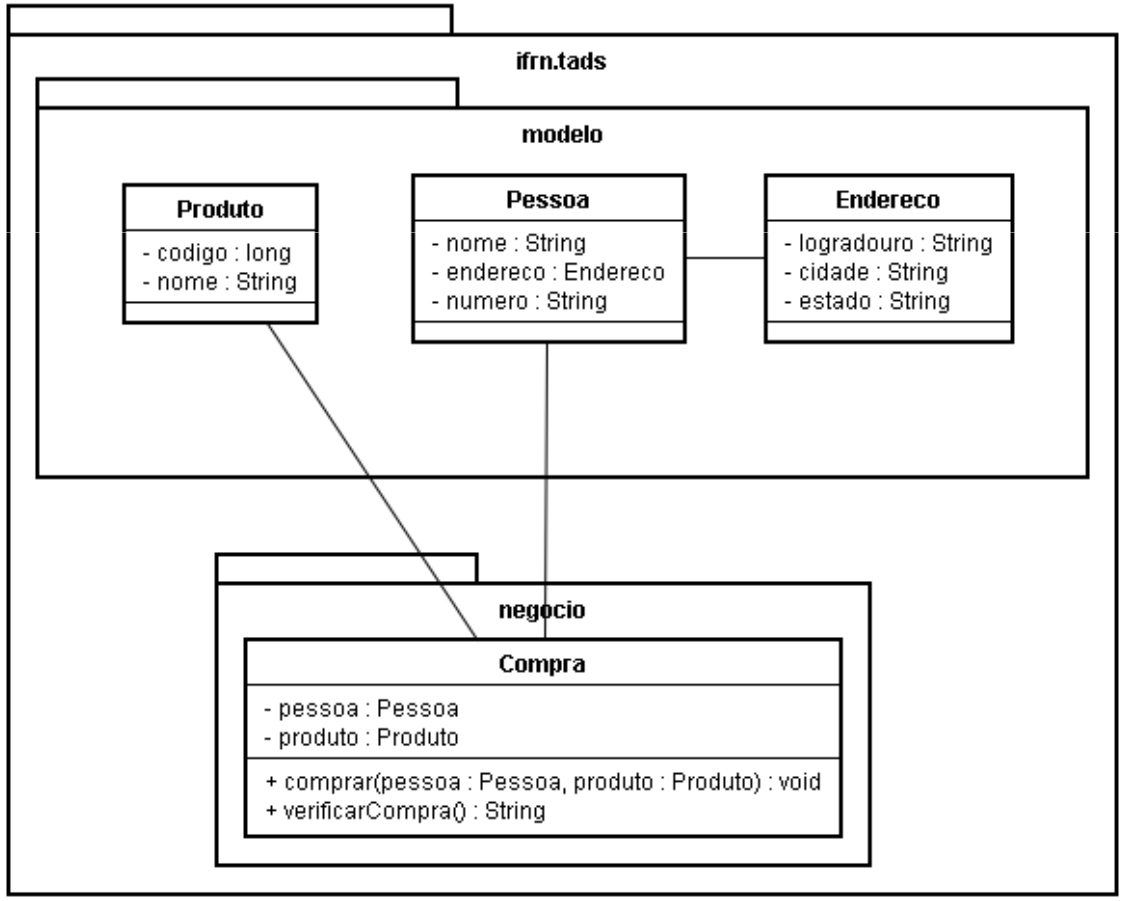
Exercício

- Inserir as Classes nos determinados pacotes:
- *ifrn.tads.modelo:*
  - Endereco
  - Pessoa
  - Produto
- *ifrn.tads.negocio:*
  - Compra

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Exercício

- Representação:



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central