

Especialização em Engenharia de Software

Programação Orientada a Objetos

Pacotes

Sérgio Soares
scbs@cin.ufpe.br

Tipos de Módulos em Java

■ Classes

- agrupam definições de métodos, atributos, inicializadores, etc.
- definem tipos

■ Pacotes

- agrupam definições de **classes** relacionadas
- estruturam sistemas de grande porte, facilitando a localização das classes
- oferece um nível mais alto de abstração
 - há mais classes do que pacotes

Pacotes e Diretórios

- As classes de um pacote são definidas em arquivos com o mesmo cabeçalho:

```
package nomeDoPacote;
```
- Cada pacote é associado a um diretório do sistema operacional:
 - os arquivos `.class` das classes do pacote são colocados neste diretório
 - é recomendável que o código fonte das classes do pacote também esteja neste diretório

Nomeando Pacotes

- O nome de um pacote é parte do nome da sua pasta associada: o pacote

`exemplos.banco`

deve estar na pasta

`c:\sergio\exemplos\banco`

assumindo que o compilador Java foi informado para procurar classes em

`c:\sergio\`

é o classpath!

Pacotes e Subdiretórios

- Subdiretórios não correspondem a "subpacotes", são pacotes como outros quaisquer
- Por exemplo, não existe nenhuma relação, além de lógica, entre `exemplos` e `exemplos.banco`:

```
package exemplos;  
public class /*...*/
```

```
package exemplos.banco;  
public class /*...*/
```

Pacotes e Visibilidade de Declarações

- **public**
 - atributos, métodos, inicializadores e classes
 - declaração pode ser utilizada (é visível) em qualquer lugar
- **private**
 - atributos, métodos e inicializadores
 - declaração só pode ser utilizada na classe onde ela é introduzida

Pacotes e e Visibilidade de Declarações

■ **protected**

- atributos, métodos e inicializadores
- declaração só pode ser utilizada no pacote onde ela é introduzida, ou nas subclasses da classe onde ela é introduzida

■ **Ausência de modificador**

- atributos, métodos, inicializadores e classes
- declaração só pode ser utilizada no pacote onde ela é introduzida

Reuso de Declarações

- As declarações feitas em um arquivo são visíveis em qualquer outro arquivo do mesmo pacote, a menos que elas sejam privadas
- Qualquer arquivo de um pacote pode usar as definições visíveis de outros pacotes, através do mecanismo de importação de pacotes...

Importação de Pacotes

- Importando definição de tipo específica:

```
package a.b;  
import c.d.NomeDoTipo;  
public class X { /*...*/ }
```

- Importando todas definições de tipo públicas:

```
package a.b;  
import c.d.*;  
public class X { /*...*/ }
```

Importação de Pacotes: Detalhes

- Tanto `NomeDoTipo` quanto `c.d.NomeDoTipo` podem ser usados no corpo de `a.b`
- `a.b` não pode definir um tipo com nome `NomeDoTipo` caso a importação tenha sido específica
- Importação de pacotes não é transitiva nem distribui sobre os arquivos de um pacote

Estruturando Aplicações com Pacotes

- Agrupar classes relacionadas, com dependência (de implementação ou conceitual) entre as mesmas
- Evitar dependência mútua entre pacotes:

```
package a.b;  
import c.d.*;  
/*...*/
```

```
package c.d;  
import a.b.*;  
/*...*/
```

Sugere problemas de modelagem!

Estruturando Aplicações com Pacotes

- Exemplo para um sistema de informação:
 - vários pacotes para as classes da GUI, um para cada conjunto de telas associadas
 - um pacote para a classe fachada e exceções associadas
 - um pacote para cada coleção de negócio, incluindo as classes básicas, coleções de dados, interfaces, e exceções associadas
 - um pacote `sistema.util` contendo classes auxiliares de propósito geral

Pacotes da Biblioteca de Java

- Acesso a Internet e WWW (`java.net`)
- Applets (`java.applet`)
- Definição de interfaces gráficas (`java.awt`)
- Suporte a objetos distribuídos (`java.rmi`)
- Interface com Banco de Dados (`java.sql`)
- Básicos: threads e manipulação de strings (`java.lang`), arquivos (`java.io`), utilitários de propósito geral (`java.util`)

Exercícios

- Estruture os exemplos de contas e bancos usando pacotes
- Implemente a classe `ConjuntoContas` usando a classe `Vector` ou a classe `Hashtable` (ambas do pacote `java.util`).

Especialização em Engenharia de Software

Programação Orientada a Objetos

Pacotes

Sérgio Soares
scbs@cin.ufpe.br