CURSO: Ciência da Computação
PERÍODO: 4º.
DISCIPLINA: Técnicas Alternativas de Programação

DATA: ____ / ____ / 2013
PROFESSOR: Andrey
AULA: 13

APRESENTAÇÃO

O objetivo desta aula é apresentar e discutir os conceitos de Pacotes em Java.

DESENVOLVIMENTO

PACOTES (PACKAGES)

No desenvolvimento de grandes aplicações é preciso organizar as classes de maneira a evitar problemas com nomes duplicados de classes e localizar o código da classe de forma eficiente. Em Java, a solução para esse problema está na organização de classes e interfaces em pacotes.

Podemos entender pacotes como um conjunto de classes que na maioria das vezes estão relacionadas através de um propósito comum. Em alguns pontos, os pacotes reproduzem a ideia de bibliotecas usadas em outras linguagens de programação. Ao criar um pacote em Java devemos armazenar o conjunto de classes num diretório em comum e estabelecer sua denominação que permitira a inclusão destas classes em outros programas Java. Com a utilização de diretivas import (importar). Vamos aprender a importar classes existentes em outros pacotes e como criar um pacote.

IMPORTANDO CLASSES DO JAVA

Devemos usar a seguinte sintaxe:

```
import nomedopacote.nomedosubpacote.NomeDaClasse;
```

Um exemplo real seria:

```
import java.io.IOException;
```

Todos os pacotes iniciados pela palavra "java" fazem parte da API padrão e são reservados à Sun. No caso acima estamos indicando que iremos utilizar a classe *IOException* do pacote "io", que faz parte do Java API onde temos as classes que permitem aos programas manipular entrada e saída de dados. Quando fazemos isso, estamos informando ao compilador que as classes da API padrão devem ser utilizadas, sendo assim, o compilador faz a busca nos seus diretórios setados na instalação. Se precisarmos de mais de uma classe do mesmo pacote podemos importar desta forma:

```
import java.io.*;
```

Esta sintaxe não implica na utilização de todas as classes do io no programa, apenas usaremos aqueles que nosso programa necessitar.

CRIANDO UM PACOTE

Definir um pacote consiste na criação de um diretório apropriado onde ficarão as classes que irão participar do mesmo. As classes adicionadas devem ser iniciadas com a seguinte sintaxe:

```
package nomedopacote;
```

Um exemplo de uma criação de uma classe pertencente a um pacote seria:

```
// ij.exemplos.parte7.Oi.java  
  
package ij.exemplos.parte7;  
  
public class Oi {  
  
    public static void main (String args[]) {  
        System.out.print("Oi");  
    }  
}
```

Com isso podemos dizer que a classe "Oi" faz parte do pacote `ij.exemplos.parte7`. O arquivo gerado `.class` deve estar na árvore de diretório de nome `ij/exemplos/parte7/` para que a classe possa ser utilizada. Vamos informar ao compilador a raiz do diretório `ij` e torna a classe acessível em outros diretórios do sistema, através de qualquer diretório do sistema podemos fazer sua execução, fazendo:

```
$ java -classpath /home/controlc/ ij.exemplos.parte7.Oi
```

Perceba que opção `classpath` indica a raiz dos pacotes definidos pelo usuário seguido pelo nome completo da classe. Para especificar vários diretórios basta utilizar o `(;)` para separar eles.

As classes pertencentes a pacotes podem ser importadas para outros programas reutilizando a mesma. Para compilar utilizamos a mesma ideia anterior, indicando o diretório raiz dos pacotes que serão utilizados, fazendo:

```
$ javac -classpath /home/controlc/ Arquivo.java
```

Para evitar a digitação constante do `classpath` e da lista de diretórios, podemos atribuir no `user/system profile`.

Duas dicas para a criação de pacotes:

1. Nos nomes dos pacotes devemos utilizar apenas letras minúsculas.
2. Quando formos utilizar pacotes que serão distribuídos, recomenda-se que a denominação do pacote utilize o nome reverso do domínio na internet da organização que originou o pacote, pois os nomes de domínios na internet são únicos.

Um exemplo seria:

Domínio: www.controlc.eti.br

Pacote: caixas

Então teríamos, `br.eti.controlc.caixas`.

ATIVIDADE

1. Construa um pacote com a sua aplicação de cadastro de cliente.
2. Compile o exemplo de acesso à base de dados importando o driver.

BIBLIOGRAFIA BÁSICA

FURGERI, SÉRGIO. *Java 2 Ensino Didático. Desenvolvendo e Implementando Aplicações*. ed. Érica. São Paulo, 2002.

DEITEL, H. M. e DEITEL, P. J.. *Java, como Programar*. Ed. Bookman. Porto Alegre. 2001.

ARNOLD Ken, GOSLING James: "The Java Programming Language Second Edition", Addison-Wesley, 1997.