Práctica 5: Packages y herencia

Ejercicio 5.1 Packages

En el ejercicio 3.2 se implementaron los 3 primeros puntos de menú de la aplicación de gestión de matrices. Se debe utilizar como punto de partida la solución contenida en el fichero Ejer6_matrix2.java. Realizar ahora los siguientes cambios.

1. Comprobar el funcionamiento y entender el programa.

2. En lugar de utilizar la clase propia Matrix.java, utilizar la que está incluida en el package Jama. Para ello incluir la línea:

```
import Jama.Matrix;
```

Y en la variable classpath, añadir el fichero Jama-1.0.2.jar:

```
set classpath=%classpath%;.;Q:\Java\Jama-1.0.2.jar
```

Esta sentencia se puede añadir al fichero .bat donde se define la variable PATH.

3. Substituir las dos instancias de Matrix, matrizA y matrizB, por un "Vector" de objetos Matrix. Para ello se define el Vector con la sentencia:

```
  static Vector<Matrix> matrices = new Vector<Matrix>();
```

Y se utilizarán los siguientes métodos de Vector:

```
void addElement(Object)   Añade un elemento, en nuestro caso una matriz al vector
int size()                Devuelve el tamaño del vector, se utilizará al listar
elementAt(int i)          Devuelve el elemento de la posición i, en nuestro caso
                          devuelve un objeto del tipo Matrix
remove(int i)             Borra el elemento en la posición i y reajusta el tamaño
```

En concreto, en el método Opcion1(), quedará de la forma:

```
matrices.addElement(LeerMatriz());
matrices.addElement(LeerMatriz());
```

Cada una de las lineas se podría haber escrito como:

```
Matrix m1 = LeerMatriz();
matrices.addElement(m1);
```

4. Generalizar el programa para que gestione un número indefinido de matrices. Para ello la opción 1, lee una sola matriz y la añade al vector. Al multiplicar 2 matrices, pregunta cuales son las matrices que se deseen multiplicar.

La selección de la matriz se puede realizar con un método `int SeleccionMatriz()` que pregunta el número de matriz y lo devuelve a la función que va a realizar la operación. Por ejemplo para conseguir la primera matriz de la multiplicación, se tiene la llamada:

```
int m1 = SeleccionMatriz();
```

5. Implementar las restantes funciones de gestión de matrices: crear matriz traspuesta, calcular matriz inversa, borrar matriz y calcular el determinante. La lista de métodos de Matrix se puede ver en la documentación de Jama (http://math.nist.gov/javanumerics/jama/).

Ejercicio 5.2 Herencia

Las matrices del ejercicio anterior se identifican por su número. Se desea darle un nombre a cada matriz pero sin modificar la clase Matrix del package Jama. Para ello crear una nueva clase NMatrix que extiende de Matrix. El código completo de dicha clase es el siguiente:

```
import Jama.Matrix;

public class NMatrix extends Matrix {
    String name;

    public NMatrix (String nombre, double[][] A, int m, int n) {
        super(A, m, n);
        name = nombre;
    }

    public void print (int w, int d) {
        System.out.println("Matrix " + name);
        super.print(w,d);
    }
}
```

Utilizar dicha clase en el ejercicio de gestión de matrices, de forma que al crear una nueva matriz se pida su nombre y al imprimirse se muestre (no es necesario ningún cambio en el proceso de impresión). No es necesario utilizar NMatrix en el resto de funciones; con esto se logra observar que al imprimir las matrices, unas las imprime con el nuevo método y otras con el original (sin una línea previa para poner el nombre de la matriz).