

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
Facultad de Ciencias



Introducción a las Ciencias de la Computación

Práctica 6: Paquetes

Profesora:

Amparo López Gaona

Ayudante:

Víctor Emiliano Cruz Hernández

Ayudante de Laboratorio:

Kevin Jair Torres Valencia

Objetivos

El objetivo de esta práctica es que el alumno comprenda el concepto de paquetes para agrupar clases y al final sea capaz de crear sus propios paquetes.

Desarrollo

Los paquetes en Java (packages) son la forma en la que Java nos permite agrupar de alguna manera lógica los componentes de nuestra aplicación que estén relacionados entre sí.

Un paquete puede contener una o más definiciones de interfaces y clases, distribuyéndose habitualmente como un archivo. De este modo, los paquetes en Java ayudan a darle una buena organización a la aplicación ya que permiten modularizar o categorizar las diferentes estructuras que componen nuestro software.

Un paquete Java se genera sencillamente incluyendo la palabra clave `package` al inicio del código en los que se definen las clases que formarán parte del mismo.

Por ejemplo:

```
package mi_paquete;
```

Java usa el sistema de archivos para administrar paquetes, con cada paquete almacenado en su propio directorio. Por ejemplo, los archivos `.java` y `.class` para cualquier clase que declare que forma parte de "mi_paquete" deben almacenarse en un directorio llamado `mi_paquete`.

Importante: El nombre del directorio tiene que ser igual al nombre del paquete.

Para utilizar los elementos de un paquete es necesario importar este en el módulo de código en curso, usando para ello la sentencia `import`.

```
import paquete.nombredeclase;
```

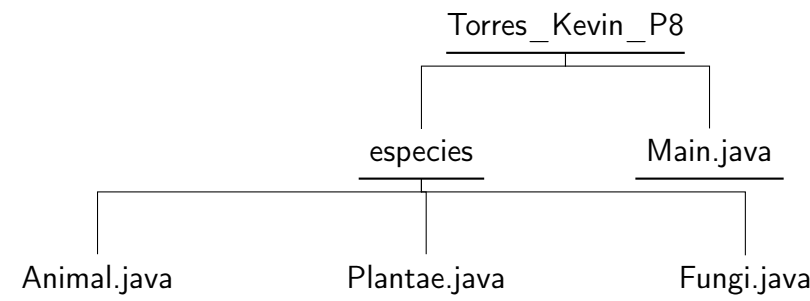
En el siguiente ejemplo estaríamos importando todas las clases contenidas dentro del paquete "mi_paquete": `import mi_paquete.*;`

Desarrollo

Problema: El Laboratorio de Especies.

1. En el directorio donde se realizara la practica crear un archivo llamado `Main.java` y un directorio llamada `especies`.
2. Dentro del directorio `especies` crear tres clases llamadas: `Animal.java`, `Plantae.java` y `Fungi.java`.
3. Juntar en un paquete llamado `especies` a las tres clases.
`package especies;`
4. Asignar los siguientes atributos a cada clase: `nombreComun`, `nombreCientífico`, `Reino`, `Clase` y `Orden`.
5. Para cada una de las clases, crear un constructor que reciba los atributos y un método `toString` para mostrar el objeto en terminal. Además, crea un método `equals` en cada clase.
6. Importar en el archivo `Main.java` todas las clases del paquete `especies`.
7. Programar en el archivo `Main.java` una solicitud de un numero al usuario:
 - Si se pasa el numero 1, se creara un objeto de la clase `Animal` y se mostrara en terminal.
 - Si se pasa el numero 2, se creara un objeto de la clase `Plantae` y se mostrara en terminal.
 - Si se pasa el numero 3, se creara un objeto de la clase `Fungi` y se mostrara en terminal.
 - Si se pasa otro numero, se enviara un mensaje explicando que el numero pasado no es válido.

A continuación se muestra la jerarquía deseada:



Formato de Entrega

1. Las prácticas serán entregadas en parejas y solo un integrante debe entregarla.
2. Cada práctica (sus archivos y directorios) deberá estar contenida en un directorio llamado EquipoX_pY, donde:
 - (a) X es el número de equipo correspondiente.
 - (b) Y es el número de la práctica.

Por ejemplo: Equipo01_p01

3. NO incluir los archivos .class dentro del directorio a entregar.
4. Los archivos de código fuente deben estar documentados.
5. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier práctica plagiada total o parcialmente será penalizada con cero para los involucrados.
6. La práctica se debe entregar en el apartado de Github Classroom correspondiente.
7. El horario y día de entrega se acordará en la clase de laboratorio y no deberá sobrepasar 2 clases de laboratorio.