

IPC1

Vacaciones Diciembre 2017

Agenda

- Control de acceso
- Clases abstractas y métodos
- Interfaces
- Paquetes
- Ejemplos prácticos proyecto 1

Control de acceso (I)

Método	Descripción
private	Un miembro privado es accesible sólo para la clase en la que está definido.
protected	La propia clase, las subclases y todas las clases dentro del mismo paquete tienen acceso a los miembros calificados con este método de acceso.
public	Todas las clases tienen acceso a los miembros públicos de la clase.
package	Sólo las clases del mismo paquete que la clase tienen acceso a los miembros

Control de acceso (II)

	public	protected	(sin modificador)	private
Clase	SI	SI	SI	SI
Subclase en el mismo paquete	SI	SI	SI	NO
No-Subclase en el mismo paquete	SI	SI	SI	NO
Subclase en diferente paquete	SI	SI/NO (*)	NO	NO
No-Subclase en diferente paquete (Universo)	SI	NO	NO	NO

(*) Los miembros (variables y metodos) de clase (static) si son visibles. Los miembros de instancia no son visibles.

Clases abstractas y métodos

Una clase que declara la existencia de métodos pero no la implementación de dichos métodos, se considera una clase abstracta.

Una clase abstracta puede contener métodos no-abstractos pero al menos uno de los métodos debe ser declarado abstracto.

Para declarar una clase o un método como abstractos, se utiliza la palabra reservada **abstract**.

```
abstract class Drawing
{
    abstract void miMetodo(int var1, int var2);
    String miOtroMetodo( ){ ... }
}
```

Una clase abstracta no se puede instanciar pero si se puede heredar y las clases hijas serán las encargadas de agregar la funcionalidad a los métodos abstractos. Si no lo hacen así, las clases hijas deben ser también abstractas.

Clases abstractas

Algunas veces, una clase que se ha definido representa un concepto abstracto y como tal, no debe ser instanciada. Por ejemplo, la comida en la vida real. ¿Has visto algún ejemplar de comida? No. Lo que has visto son ejemplares de manzanas, pan, y chocolate.

Comida representa un concepto abstracto de cosas que son comestibles. No tiene sentido que exista una instancia de comida. Similarmente en la programación orientada a objetos, se podría modelar conceptos abstractos pero no querer que se creen objetos de ellos.

Una clase abstracta es una clase que sólo puede tener subclases--no puede ser instanciada.

Métodos abstractos

Métodos que no tienen implementación

Primero se debe declarar una clase abstracta, ObjetoGrafico, para proporcionar las variables miembro y los métodos que van a ser compartidos por todas las subclases, como la posición actual y el método **moverA()**.

También se deberían declarar métodos abstractos como **dibujar()**, que necesita ser implementado por todas las subclases, pero de manera completamente diferente (no tiene sentido crear una implementación por defecto en la superclase).

```
abstract class ObjetoGrafico {
    int x, y;
    . . .
    void moverA(int nuevaX, int nuevaY) {
        . . .
    }
    abstract void dibujar();
}

class Circulo extends ObjetoGrafico {
    void dibujar() {
        . . .
    }
}

class Rectangulo extends ObjetoGrafico {
    void dibujar() {
        . . .
    }
}
```

Interfaces (I)

Una *interfaz* es una especie de plantilla para la construcción de clases. Normalmente una interfaz se compone de un conjunto de declaraciones de **cabeceras de métodos** (sin implementar, de forma similar a un método abstracto) que especifican un **protocolo de comportamiento** para una o varias clases. Además, una clase puede implementar una o varias interfaces: en ese caso, la clase debe proporcionar la declaración y definición de **todos** los métodos de cada una de las interfaces

```
/**
 * Declaracion de la interfaz Modificacion
 */
public interface Modificacion {
    void incremento(int a);
}
```


Interfaces (II)

Una *interfaz* puede parecer similar a una *clase abstracta*, pero existen una serie de diferencias entre una interfaz y una clase abstracta:

- **Todos** los métodos de una interfaz se declaran implícitamente como abstractos y públicos.
- Una clase abstracta no puede *implementar* los métodos declarados como abstractos, una interfaz no puede *implementar* ningún método (ya que todos son abstractos).
- Una interfaz no declara variables de instancia.
- Una clase puede implementar varias interfaces, pero sólo puede tener una clase ascendiente directa.
- Una clase abstracta pertenece a una jerarquía de clases mientras que una interfaz **no** pertenece a una jerarquía de clases. En consecuencia, clases sin relación de herencia pueden implementar la misma interfaz.

Paquetes

Los paquetes son una forma de organizar grupos de clases. Un paquete contiene un conjunto de clases relacionadas bien por finalidad, por ámbito o por herencia.

Los paquetes resuelven el problema del conflicto entre los nombres de las clases. Al crecer el número de clases crece la probabilidad de designar con el mismo nombre a dos clases diferentes.

Las clases tienen ciertos privilegios de acceso a los miembros dato y a las funciones miembro de otras clases dentro de un mismo paquete.

La palabra reservada *import*

Para importar clases de un paquete se usa el comando **import**. Se puede importar una clase individual

```
import java.awt.Font;
```

o bien, se puede importar las clases declaradas públicas de un paquete completo, utilizando un asterisco (*) para reemplazar los nombres de clase individuales.

```
import java.awt.*;
```

Los paquetes estándar

Paquete	Descripción
java.applet	Contiene las clases necesarias para crear applets que se ejecutan en la ventana del navegador
java.awt	Contiene clases para crear una aplicación GUI independiente de la plataforma
java.io	Entrada/Salida. Clases que definen distintos flujos de datos
java.lang	Contiene clases esenciales, se importa implícitamente sin necesidad de una sentencia import .
java.net	Se usa en combinación con las clases del paquete java.io para leer y escribir datos en la red.
java.util	Contiene otras clases útiles que ayudan al programador

Dudas proyecto 1

Recordar fecha de entrega el día 26 de diciembre antes de las 22:00, se calificará el día 27 a partir de las 9:00 exclusivamente desde el repositorio y en la máquina de la persona que está calificando.