

Modelado y Diseño de Clases

2.2 Paquetes y Modificadores de Acceso

Programación
Septiembre 2015

Objetivo

- Comprender el concepto de paquete y modificadores de acceso en la POO

Paquetes (1/2)

- ▶ En la POO los paquetes son contenedores que permiten la agrupación de distintas partes (clases, interfaces, enumeraciones) de un sistema de software, y donde estas partes comparten funcionalidades comunes
- ▶ El agrupamiento de clases permite tener mejor organizado el código de un sistema: al considerar características comunes
- ▶ Permite tener un mayor control en los niveles de acceso
- ▶ En los paquetes las clases son únicas en comparación con las de otros paquetes

Paquetes (2/2)

- ▶ Los paquetes proporcionan una forma de ocultar clases, con la finalidad de evitar que otros programas o paquetes accedan a clases que son de uso exclusivo de una aplicación determinada
- ▶ Los paquetes resuelven los conflictos entre los nombres de las clases, esto es porque a medida al crecer el número de clases existe mayor probabilidad de encontrar dos clases diferentes con un mismo nombre
- ▶ Entre sus ventajas se encuentra la reutilización de los nombres de clases, por la independencia de los paquetes
- ▶ Todas las clases deben pertenecer a un paquete, en caso de no especificar ninguno existe un paquete por defecto (sin nombre) donde pertenecerá la clase

Sub-Paquetes

- ▶ Los paquetes son anidados formando jerarquías (similar a los directorios)
- ▶ Los sub-paquetes son derivados de los paquetes principales
- ▶ Los paquetes o sub-paquetes corresponden a directorios
 - ▶ Nombre de paquete = nombre de directorio
 - ▶ Paquete anidados = subdirectorios

Implementación en Java

- ▶ En Java existe la restricción de que una clases debe tener un nombre único, dentro del paquete al que pertenece, no obstante, es posible que dos clases tengan el mismo nombre siempre y cuando pertenezcan a paquetes distintos
- ▶ La declaración de un paquete en Java es utilizando la palabra reservada **package**

```
package biblioteca;  
public class Persona{  
    ...  
}
```

- ▶ Para la incorporación de nuevas clases en el paquete se debe colocar la misma sentencia al comienzo de los archivos que contengan las clases
- ▶ En lo archivos solo pueden existir una sentencia **package**

Implementación en Java

- ▶ Los sub-paquetes también se declaran utilizando la palabra reservada **package**
 - ▶ Para ello, se sigue el nombre del paquete, un punto (.), y posteriormente el nombre del sub-paquete
 - ▶ **package** paquete.sub-paquete;

```
package biblioteca.rh;  
public class Trabajador extends Persona{  
    ...  
}
```

Importar Paquetes en Java

- ▶ La palabra reservada **importa** es utilizada para importar clases desde otro paquete o sub-paquete
- ▶ En java se importan automáticamente las clases del paquete `java.lang`
- ▶ `import paquete.sub-paquete.nombreDeLaClase;` //También se puede usar un comodín

```
package biblioteca.rh;  
import biblioteca.Persona;  
public class Trabajador extends Persona{  
    ...  
}
```

- ▶ En este ejemplo se importa la clase `Persona` del paquete `biblioteca`, en lugar del nombre de la clase se podría utilizar un `*` para indicar que todas las clases de ese paquete serán importadas, por ejemplo:

- ▶ `import paquete.sub-paquete.*;`
- ▶ `import biblioteca.*;`
- ▶ `import biblioteca.rh.*;`

Ejemplo de Paquete (1/4)

- ▶ biblioteca

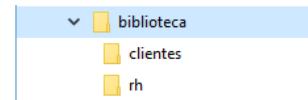
- ▶ Persona.java

- ▶ rh

- ▶ Trabajador.java

- ▶ clientes

- ▶ Socio.java



Nombre	Fecha de modifica...	Tipo	Tamaño
clientes	12/09/2015 12:07 a...	Carpeta de archivos	
rh	12/09/2015 12:09 a...	Carpeta de archivos	
Persona.class	12/09/2015 12:08 a...	Archivo CLASS	2 KB
Persona.java	12/09/2015 12:01 a...	Archivo JAVA	1 KB

- ▶ Para compilar las clases que están en el paquete debemos colocarnos en el directorio padre del paquete y compilar desde esa ruta

Ejemplo de Paquete (2/4)

- ▶ Clase Persona en la raíz de paquete biblioteca
- ▶ biblioteca
 - ▶ Persona.java
 - ▶ rh
 - ▶ Trabajador.java
 - ▶ clientes
 - ▶ Socio.java

```
File Edit Selection Find View Goto Tools Project Preferences Help
Persona.java x Socio.java x Trabajador.java x Main.java x
1 package biblioteca;
2 public class Persona{
3     //Características
4     private String nombre;
5     private int edad;
6     private String telefono;
7     private String correo;
8
9     public Persona(){
10    }
11
12    public Persona(String nombre, int edad, String telefono, String correo){
13        this.nombre = nombre;
14        this.correo = correo;
15        this.edad = edad;
16        this.telefono = telefono;
17    }
18
19    public String getNombre(){
20        return nombre;
21    }
22
23
24    public String getCorreo(){
25        return correo;
26    }
27
28
29    public int getEdad(){
30        return edad;
31    }
32
```

Ejemplo de Paquete (3/4)

► Clase Trabajador en el subdirectorio y sub-paquete rh

► También es necesario importar el paquete padre biblioteca debido a que se utiliza la clase Persona, la cual está ubicada en ese nivel

► biblioteca

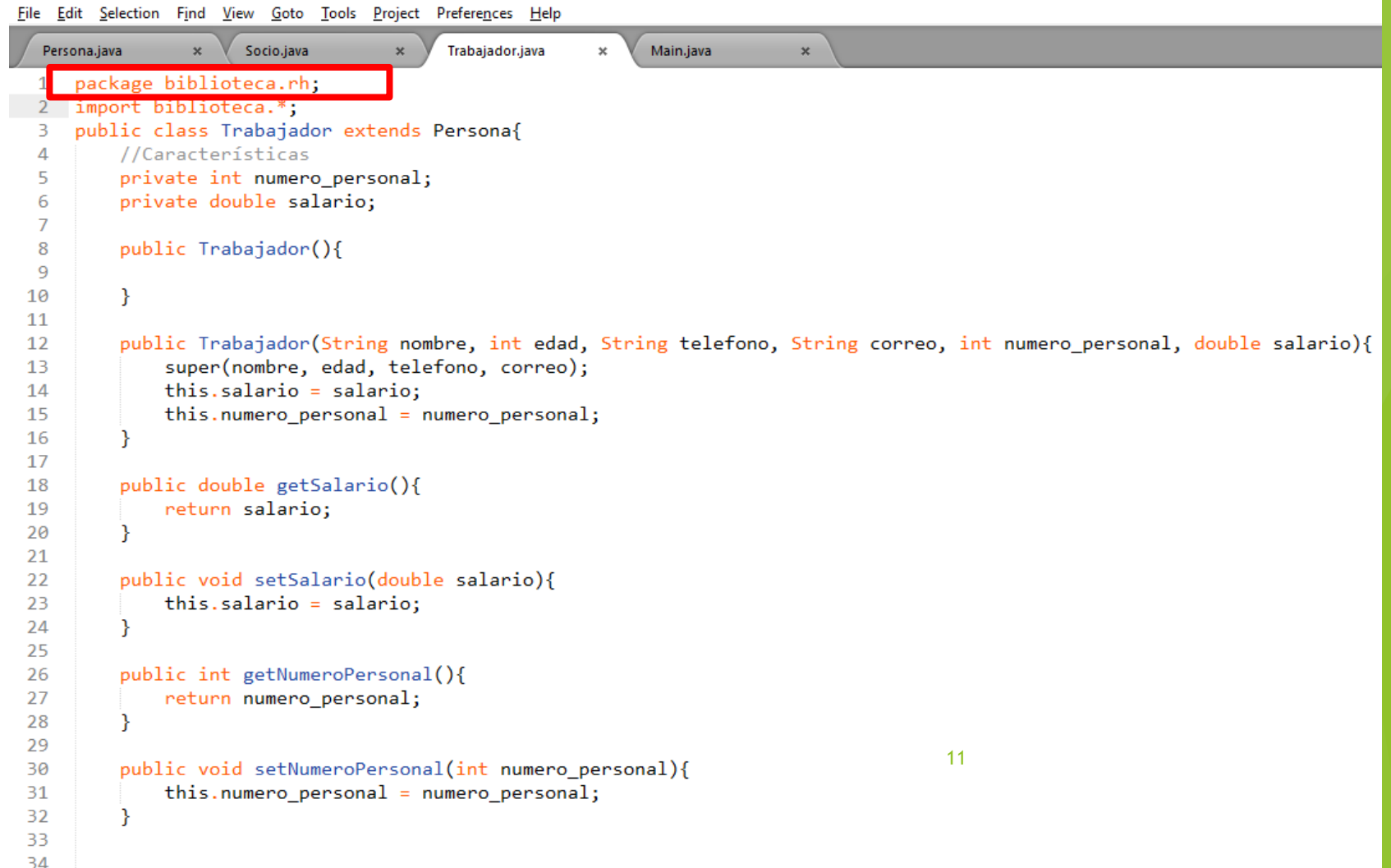
► Persona.java

► rh

► Trabajador.java

► clientes

► Socio.java



```
File Edit Selection Find View Goto Tools Project Preferences Help
Persona.java x Socio.java x Trabajador.java x Main.java x
1 package biblioteca.rh;
2 import biblioteca.*;
3 public class Trabajador extends Persona{
4     //Características
5     private int numero_personal;
6     private double salario;
7
8     public Trabajador(){
9
10    }
11
12    public Trabajador(String nombre, int edad, String telefono, String correo, int numero_personal, double salario){
13        super(nombre, edad, telefono, correo);
14        this.salario = salario;
15        this.numero_personal = numero_personal;
16    }
17
18    public double getSalario(){
19        return salario;
20    }
21
22    public void setSalario(double salario){
23        this.salario = salario;
24    }
25
26    public int getNumeroPersonal(){
27        return numero_personal;
28    }
29
30    public void setNumeroPersonal(int numero_personal){
31        this.numero_personal = numero_personal;
32    }
33
34
```

Ejemplo de Paquete (4/4)

- ▶ Clase Socio en el subdirectorio y sub-paquete clientes
- ▶ Es necesario importar el paquete padre biblioteca debido a que se utiliza la clase Persona, la cual está ubicada en ese nivel

- ▶ biblioteca
 - ▶ Persona.java
 - ▶ rh
 - ▶ Trabajador.java
 - ▶ clientes
 - ▶ Socio.java

C:\Programacion\práctica 3 - Paquetes\biblioteca\clientes\Socio.java - Sublime Text 2 (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```
Persona.java x Socio.java x Trabajador.java x Main.java x
1 package biblioteca.clientes; /iniciar con package
2 import biblioteca.*;
3 public class Socio extends Persona{
4     //Características
5     private int numero_socio;
6
7     public Socio(String nombre, int edad, String telefono, String correo, int numero_socio){
8         super(nombre, edad, telefono, correo);
9         this.numero_socio = numero_socio;
10    }
11
12    //Operaciones
13    public void mostrar(){
14        super.mostrar();
15        System.out.println(getNombre() + ", " + numero_socio);
16    }
17
18 }
```

Compilación desde Comandos

- Para compilar las clases que están en el paquete debemos colocarnos en el directorio padre del paquete y compilar desde esa ruta

```
C:\WINDOWS\system32\cmd.exe

C:\Programacion\práctica 3 - Paquetes>tree
Listado de rutas de carpetas para el volumen OS
El número de serie del volumen es 00000075 C0A5:1BCF
C:.
├── biblioteca
│   ├── clientes
│   └── rh
└──

C:\Programacion\práctica 3 - Paquetes>javac biblioteca\Persona.java
C:\Programacion\práctica 3 - Paquetes>javac biblioteca\clientes\Socio.java
C:\Programacion\práctica 3 - Paquetes>javac biblioteca\rh\Trabajador.java
```

Uso de Paquetes (1/2)

- ▶ La clase ejecutable puede estar fuera o dentro del paquete
- ▶ 1) Ejecución desde fuera del paquete:
 - ▶ En este ejemplo la clase se coloca en el directorio padre del paquete (Main no estará en ningún paquete-observa que no lleva la sentencia package)
 - ▶ práctica 3 - Paquetes

- ▶ Main.java
- ▶ biblioteca
 - ▶ Persona.java
 - ▶ rh
 - ▶ Trabajador.java
 - ▶ clientes
 - ▶ Socio.java

C:\WINDOWS\system32\cmd.exe

```
C:\Programacion\práctica 3 - Paquetes>dir Main.java
El volumen de la unidad C es OS
El número de serie del volumen es: C0A5-1BCF

Directorio de C:\Programacion\práctica 3 - Paquetes
12/09/2015  12:10 a. m.                688 Main.java
              1 archivos                688 bytes
              0 dirs  51,853,737,984 bytes libres
```

```
C:\Programacion\práctica 3 - Paquetes>tree
Listado de rutas de carpetas para el volumen OS
El número de serie del volumen es 00000057 C0A5:1BCF
C:.
├── biblioteca
│   └── clientes
│       └── rh
```

```
C:\Programacion\práctica 3 - Paquetes>javac Main.java
```

```
C:\Programacion\práctica 3 - Paquetes>java Main
Mi nombre es Juan, edad = 20, telefono = 8101010
38192, 2345.0
```

- ▶ En el código de la clase Main es necesario importar las clases que se utilizarán dentro de la misma (en este caso es biblioteca.rh.Trabajador)

```
1 import biblioteca.rh.Trabajador; //O también se podría utilizar import biblioteca.rh.*;
2
3 //Esta clase no forma parte del paquete biblioteca...
4 public class Main{
5     public static void main(String[] args){
6         Trabajador t = new Trabajador("Juan", 20, "8101010", "juan@uv.mx", 38192, 2345);
7         t.mostrar();
8
9         Trabajador t1 = new Trabajador();
```

Uso de Paquetes (2/2)

2) Ejecución de una clase localizada dentro del paquete (ejecutable):

- ▶ En este ejemplo la clase se coloca en el directorio biblioteca (Main estará en el paquete biblioteca - observa que el código incluye la sentencia package)

C:\WINDOWS\system32\cmd.exe

C:\Programacion\práctica 3 - Paquetes>javac biblioteca\Main.java

C:\Programacion\práctica 3 - Paquetes>java biblioteca.Main

Mi nombre es Juan, edad = 20, telefono = 8101010
38192, 2345.0

- ▶ práctica 3 - Paquetes

- ▶ biblioteca

- ▶ Main.java

- ▶ Persona.java

- ▶ rh

- ▶ Trabajador.java

- ▶ clientes

- ▶ Socio.java

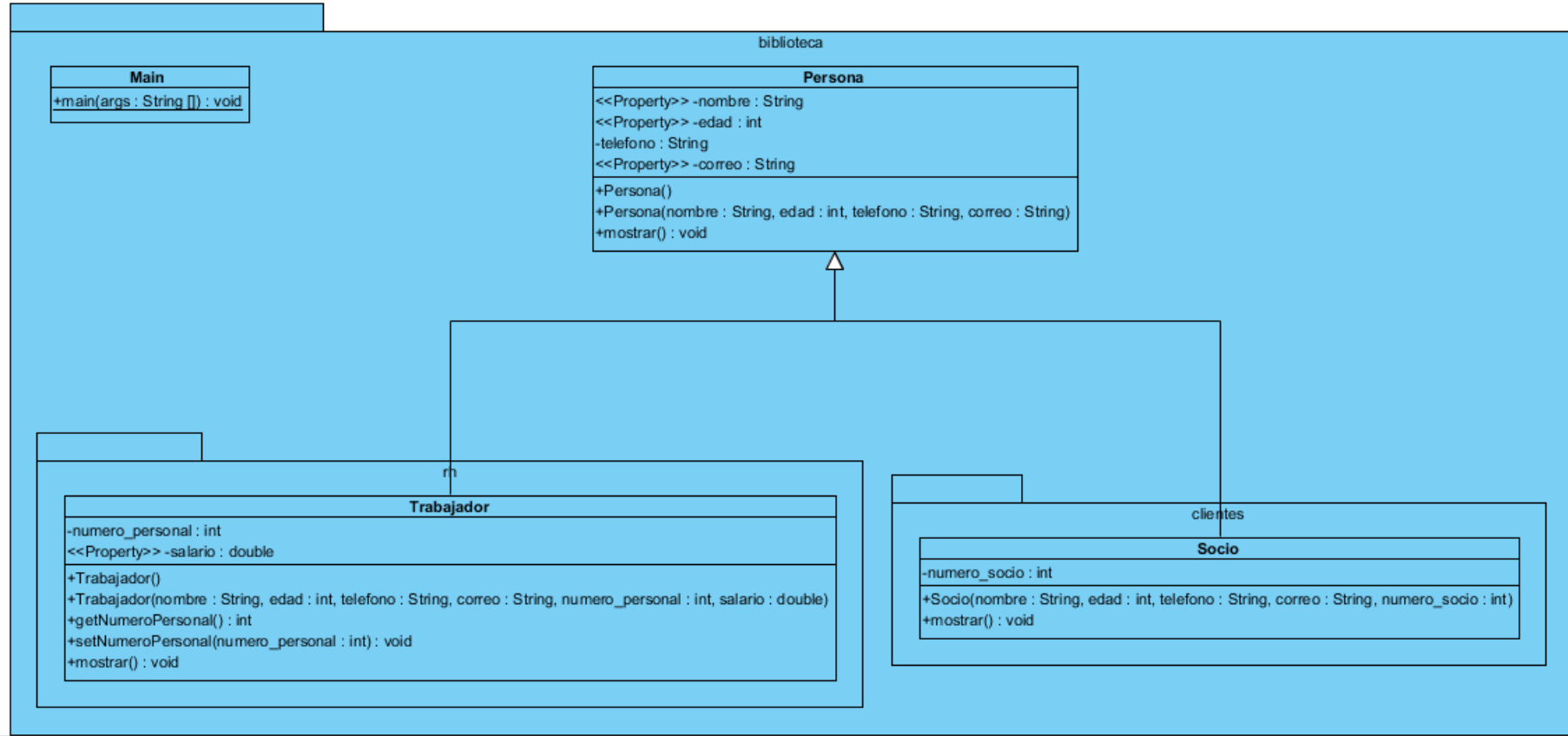
- ▶ En el código de la clase Main es necesario importar las clases que se utilizarán dentro de la misma (en este caso es biblioteca.rh.Trabajador)

```
1 package biblioteca;
2 import biblioteca.rh.Trabajador; //O también se podría utilizar import biblioteca.rh.*
3
4 //Esta clase no forma parte del paquete biblioteca...
5 public class Main{
6     public static void main(String[] args){
7         Trabajador t = new Trabajador("Juan", 20, "8101010", "juan@uv.mx", 38192, 2345);
8         t.mostrar();
9
10        Trabajador t1 = new Trabajador();
11    }
```

Modificadores de Acceso

	Clase	Paquete	Subclase	Mundo Entero
public	Sí	Sí	Sí	Sí
protected	Sí	Sí	Sí	No
<i>sin modificador (default)</i>	Sí	Sí	No	No
private	Sí	No	No	No

Ejemplo



Modificadores de Acceso - Ámbito de Atributos y Métodos

- ▶ **Público**
 - ▶ Cuando se declara un atributo o método de tipo **public** todas las clases tienen acceso a él
- ▶ **Protected**
 - ▶ Cuando se declara un atributo o método **protected** sólo se tiene acceso a él desde la propia clase que lo define y las que heredan de él
- ▶ **Private**
 - ▶ Cuando se declara un atributo o método de tipo **private** sólo se tiene acceso a él desde la clase que lo define nada más
- ▶ **Package (sin modificador)**
 - ▶ Tiene acceso a estos atributos o métodos las clases que se encuentren contenidas en el mismo paquete (sub-paquete) que la clase que los define. Es el modificador por default cuando no se coloca ninguno de los anteriores

Visibilidad de Clases

- ▶ Las clases pueden ser declaradas como:
 - ▶ **defecto** (sin modificador de acceso), sólo disponible en su paquete
 - ▶ **public** accesible desde cualquier parte

- ▶ Ejemplo

```
public class A{ //pública
    ...
}

class B{ //acceso default = solo visible desde este paquete
    ...
}
```

- ▶ En cada archivo fuente SOLO PODRÁ existir una clase con acceso público
- ▶ Cuando en un archivo hay una clase **public**, el archivo debe llamarse exactamente igual que la clase