

CLASES EN JAVA

PARTICULARIDADES

Ámbito en las clases

```
class Ambitos {  
    int atributo;
```

```
    ...
```

```
    void metodo() {
```

```
        int varLocal;
```

```
        ...
```

```
        while(...) {
```

```
            int varBloque;
```

```
            ...
```

```
        } //del while
```

```
        ...
```

```
    } //del método
```

```
    ...
```

```
} //de la clase
```

← ámbito de la clase: podemos utilizar atributo, ~~varLocal~~ y ~~varBloque~~

← ámbito del método: podemos utilizar atributo, varLocal y ~~varBloque~~

■ ámbito del while: podemos utilizar atributo, varLocal y varBloque

Ocultación de atributos

```
public class Ambito {  
    int edad; //atributo entero  
  
    void metodo() {  
        double edad; //variable local. Oculta al atributo edad (entero)  
        edad = 8.2; //variable local, no el atributo de la clase  
        ...  
    }  
}
```

Objeto this

```
public class Ambito {  
    int edad; //atributo entero  
  
    void metodo() {  
        double edad; //oculta el atributo edad (entero)  
  
        edad = 20.0; //variable local, no el atributo  
        this.edad = 30; //atributo de la clase  
    }  
}
```

Constructores

```
public class Estudiante {  
    String nombre;  
    int edad;  
    String direccion;  
  
    Estudiante(String nombre, int edad, String direccion) {  
        this.nombre = nombre;  
        this.edad = edad;  
        this.direccion = direccion;  
    }  
}
```

```
public class principal {  
    public static void main(String[] args) {  
        // error  
        Estudiante e1 = new Estudiante();  
  
        System.out.println(e1.nombre);  
    }  
}
```

```
public class principal {  
    public static void main(String[] args) {  
        Estudiante e1 = new Estudiante("Rodomiro", 99, "Calle Rafael de Utrera");  
  
        System.out.println(e1.nombre);  
    }  
}
```

Constructor sobrecargado

```
public class Estudiante {  
    String nombre;  
    int edad;  
    String direccion;  
  
    Estudiante(String nombre, int edad, String direccion) {  
        this.nombre = nombre;  
        this.edad = edad;  
        this.direccion = direccion;  
    }  
  
    Estudiante() {}  
}
```

```
public class principal {  
    public static void main(String[] args) {  
        Estudiante e1 = new Estudiante();  
  
        System.out.println(e1.nombre);  
    }  
}
```

```
public class principal {  
    public static void main(String[] args) {  
        Estudiante e1 = new Estudiante("Rodomiro", 99, "Calle Rafael de Utrera");  
  
        System.out.println(e1.nombre);  
    }  
}
```

Atributos y métodos estáticos

```
public class Estudiante {  
    static int numero_estudiantes = 0;  
  
    String nombre;  
    int edad;  
    String direccion;  
  
    Estudiante() {  
        numero_estudiantes++;  
    }  
}
```

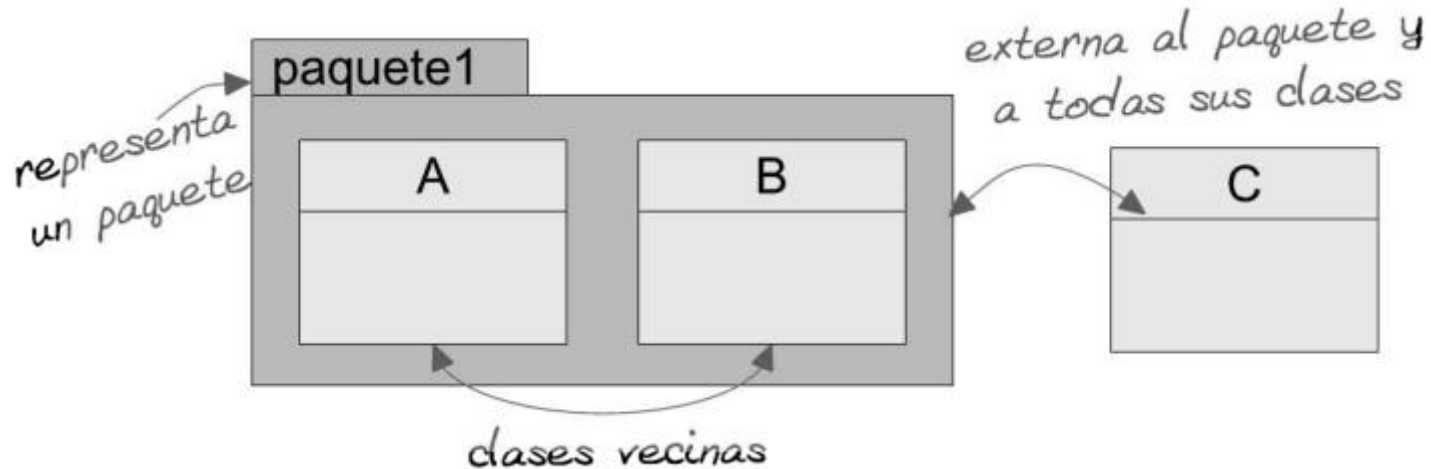
```
public class principal {  
    public static void main(String[] args) {  
        Estudiante e1 = new Estudiante();  
        Estudiante e2 = new Estudiante();  
        Estudiante e3 = new Estudiante();  
  
        System.out.println("# de estud: " + e1.num_estud);  
        System.out.println("# de estud: " + e2.num_estud);  
        System.out.println("# de estud: " + e3.num_estud);  
        System.out.println("# de estud: " +  
        Estudiante.num_estud);  
  
    }  
}
```

Atributos y métodos estáticos

```
public class Estudiante {  
    static int num_estud = 0;  
  
    String nombre;  
    int edad;  
    String direccion;  
  
    Estudiante() {  
        numero_estudiantes++;  
    }  
  
    static int dame_num_estud() {  
        return num_estud;  
    }  
  
    static int dame_nombre() {  
        return nombre; // error  
    }  
}
```

```
public class principal {  
    public static void main(String[] args) {  
        Estudiante e1 = new Estudiante();  
        Estudiante e2 = new Estudiante();  
        Estudiante e3 = new Estudiante();  
  
        System.out.println("# de estud.: " + e1.dame_num_estud());  
        System.out.println("# de estud.: " + Estudiante.dame_num_estud());  
    }  
}
```

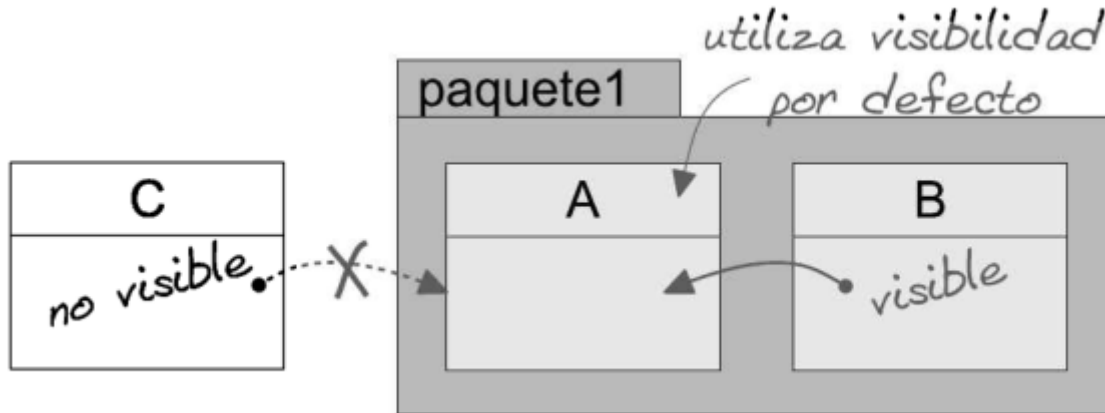

Visibilidad entre clases



Visibilidad entre clases

```
package paquete1;
```

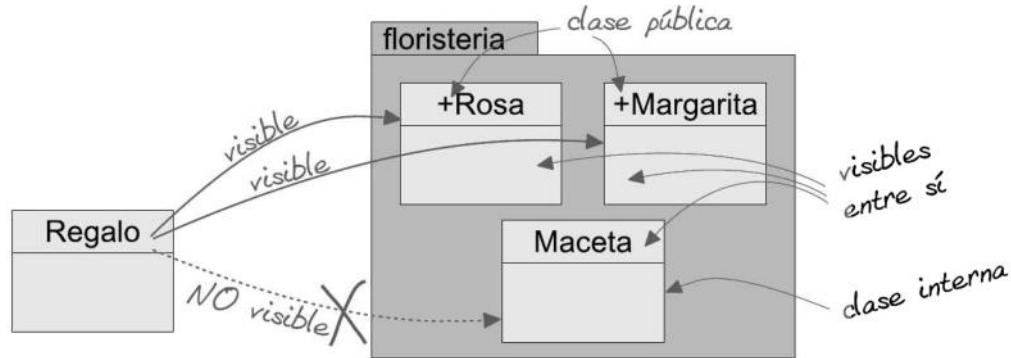
```
class A { //sin modificador de acceso  
    ...  
}
```



Visibilidad entre clases

```
package paquete1;  
  
public class A { //clase marcada como pública  
    ...  
}
```

```
import paquete1.A; //ahora C puede utilizar la clase A  
  
class C {  
    ...  
}
```



```
package floristeria;  
  
public class Rosa {  
    ...  
}
```

```
package floristeria;  
  
public class Margarita {  
    ...  
}
```

```
package floristeria;  
  
class Maceta {  
    ...  
}
```

Modificadores public y private

```
public class Estudiante {  
    String nombre;  
    private int edad;  
  
    int dame_edad() {  
        return edad;  
    }  
  
    String dame_nombre() {  
        return nombre;  
    }  
}
```

```
public class principal {  
    public static void main(String[] args) {  
        Estudiante e1 = new Estudiante();  
  
        System.out.println(e1.nombre);  
        System.out.println(e1.edad); // error: es private  
        System.out.println(e1.dame_edad()); // Claro que sí guapi  
    }  
}
```

Enumerados

```
public enum DiaDeLaSemana {  
    LUNES,  
    MARTES,  
    MIERCOLES,  
    JUEVES,  
    VIERNES,  
    SABADO,  
    DOMINGO  
}
```

```
public static void main(String[] args) {  
    DiaDeLaSemana dia_programacion = DiaDeLaSemana.VIERNES;  
  
    System.out.println(dia_programacion);  
}
```

```
public static void main(String[] args) {  
    Scanner t = new Scanner(System.in);  
    String dia = t.nextLine(); // Introducimos VIERNES  
    DiaDeLaSemana dia_programacion =  
        DiaDeLaSemana.valueOf(dia);  
  
    System.out.println(dia_programacion);  
}
```