

7.C. Herencia.

1. Herencia.

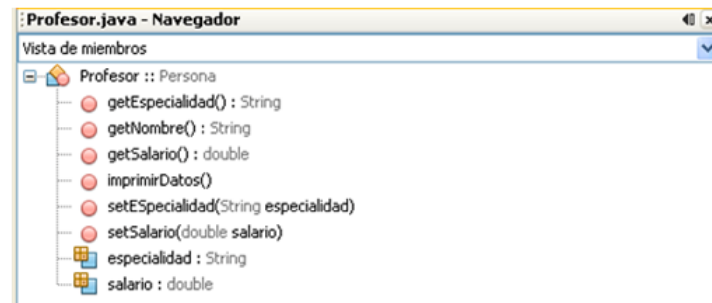
1.4. Utilización de miembros heredados (II). Métodos.

Del mismo modo que se heredan los **atributos**, también se heredan los **métodos**, convirtiéndose a partir de ese momento en otros **métodos** más de la **clase derivada**, junto a los que hayan sido definidos específicamente.

En el ejemplo de la clase **Persona**, si dispusiéramos de métodos **get** y **set** para cada uno de sus tres atributos (**nombre**, **apellidos**, **fechaNacim**), tendrías seis métodos que podrían ser heredados por sus **clases derivadas**. Podrías decir entonces que la clase **Alumno**, derivada de **Persona**, tiene diez métodos:

- Seis por ser **Persona** (**getNombre**, **getApellidos**, **getFechaNacim**, **setNombre**, **setApellidos**, **setFechaNacim**).
- Oros cuatro más por ser **Alumno** (**getGrupo**, **setGrupo**, **getNotaMedia**, **setNotaMedia**).

Sin embargo, sólo tendrías que definir esos cuatro últimos (los **específicos**) pues los **genéricos** ya los has heredado de la **superclase**.



Autoevaluación

En Java los métodos heredados de una superclase deben volver a ser definidos en las subclases. ¿Verdadero o Falso?

- ☐ Verdadero
- ☐ Falso

Ejercicio resuelto

Dadas las clases **Persona**, **Alumno** y **Profesor** que has utilizado anteriormente, implementa métodos **get** y **set** en la clase **Persona** para trabajar con sus tres atributos y en las clases **Alumno** y **Profesor** para manipular sus cinco atributos (tres heredados más dos específicos), teniendo en cuenta que los métodos que ya hayas definido para **Persona** van a ser heredados en **Alumno** y en **Profesor**.

Solución:

Una posible solución:

1. Clase **Persona**.

```
public class Persona {  
  
    protected String nombre;  
  
    protected String apellidos;  
  
    protected GregorianCalendar fechaNacim;  
  
  
  
    // Método getNombre
```

```
public String getNombre (){
```

```
    return nombre;
```

```
}
```

```
// Método getApellidos
```

```
public String getApellidos (){
```

```
    return apellidos;
```

```
}
```

```
// Método getFechaNacim
```

```
public GregorianCalendar getFechaNacim (){
```

```
    return this.fechaNacim;
```

```
}
```

```
// Método setNombre
```

```
public void setNombre (String nombre){
```

```
    this.nombre= nombre;
```

```
}
```

```
// Método setApellidos
```

```
public void setApellidos (String apellidos){
```

```
    this.apellidos= apellidos;
```

```
}
```

```
// Método setFechaNacim
```

```
public void setFechaNacim (GregorianCalendar fechaNacim){
```

```
    this.fechaNacim= fechaNacim;
```

```
}
```

2. Clase Alumno.

Al heredar de la clase **Persona** tan solo es necesario escribir métodos para los nuevos atributos (**métodos especializados** de acceso a los **atributos especializados**), pues los **métodos genéricos** (de acceso a los **atributos genéricos**) ya forman parte de la clase al haberlos heredado.

```
public class Alumno extends Persona {
```

```
    protected String grupo;
```

```
    protected double notaMedia;
```

```
// Método getGrupo
```

```
public String getGrupo (){
```

```
    return grupo;
```

```
}
```

```

// Método getNotaMedia

public double getNotaMedia (){

    return notaMedia;

}

// Método setGrupo

public void setGrupo (String grupo){

    this.grupo= grupo;

}

// Método setNotaMedia

public void setNotaMedia (double notaMedia){

    this.notaMedia= notaMedia;

}

}

```

Aquí tienes una demostración práctica de cómo la herencia permite una reutilización eficiente del código, evitando tener que repetir atributos y métodos. Sólo has tenido que escribir cuatro métodos en lugar de diez.

3. Clase **Profesor**.

Seguimos exactamente el mismo procedimiento que con la clase **Alumno**.

```

public class Profesor extends Profesor {

    String especialidad;

    double salario;

    // Método getEspecialidad

    public String getEspecialidad (){

        return especialidad;

    }

    // Método getSalario

    public double getSalario (){

        return salario;

    }

    // Método setSalario

    public void setSalario (double salario){

        this.salario= salario;

    }

}

```

```
// Método setEspecialidad
```

```
public void setEspecialidad (String especialidad){
```

```
    this.especialidad= especialidad;
```

```
}
```

```
}
```