

1.- (2 puntos) Dado el siguiente código

```
abstract class Profesor {  
    private String nombre;  
  
    Profesor(String nombre) {  
        this.nombre=nombre;  
    }  
    abstract void impartirDocencia();  
}  
  
interface Investigador {  
    public void investigar();  
}  
  
class ProfesorSecundaria extends Profesor {  
  
    ProfesorSecundaria(String nombre) {  
        super(nombre);  
    }  
  
    @Override  
    void impartirDocencia() {  
        System.out.println("Dando clase en un instituto");  
    }  
  
    void vigilarRecreo() {  
        System.out.println("Vigilando el recreo en un instituto");  
    }  
}  
  
class ProfesorUniversidad extends Profesor implements Investigador {  
  
    ProfesorUniversidad(String nombre) {  
        super(nombre);  
    }  
  
    @Override  
    void impartirDocencia() {  
        System.out.println("Dando clase en una facultad");  
    }  
}
```

```
@Override  
public void investigar() {  
    System.out.println("Investigando en una facultad");  
}  
  
class ProfesorUniversidadDoctor extends ProfesorUniversidad {  
  
    ProfesorUniversidadDoctor(String nombre) {  
        super(nombre);  
    }  
  
    @Override  
    public void investigar() {  
        super.investigar();  
        System.out.println("Como doctor");  
    }  
}
```

Indicar los errores relacionados con compatibilidad de tipos presentes en el siguiente programa principal. Se deberá indicar el número de línea, si se produce en tiempo de compilación o ejecución y dar una muy breve descripción de la causa.

```
1. public class Enero22 {  
2.     public static void main(String[] args) {  
3.         ((ProfesorUniversidad) new ProfesorUniversidadDoctor("Manolo")).investigar();  
4.         Profesor prof1=new ProfesorUniversidad("Pepe");  
5.         prof1.investigar();  
6.         Investigador inv1=prof1;  
7.         ProfesorSecundaria secu1=(ProfesorSecundaria) (Profesor) new ProfesorUniversidad("Juan");  
8.     }  
9. }
```

LÍNEA

MOMENTO

EXPLICACIÓN

5

Compilacióñ

La clase Profesor no realiza la interfaz  
ni dispone del método investigar

6

Compilacióñ

La clase Profesor no realiza la interfaz luego  
una variable de tipo Investigador no podrá  
referenciar a un Profesor

El siguiente ejercicio aparece con comentarios y que es imposible pegarlo aquí sin usar el trabajo de otra persona, está relacionado con los errores de visibilidad. Se hará en Proyecto-Ruby

3.-(2 puntos) Dado el siguiente código

```
interface Dobblable {
    void doblar();
}

class Mapa implements Dobblable {
    @Override
    public void doblar(){
        //...
    }
}

class Alfombra implements Dobblable {
    @Override
    public void doblar(){
        //...
    }
}

class Kebab implements Dobblable {
    @Override
    public void doblar(){
        //...
    }
}
```

Este ejercicio se centra en la operación para enrollar mapas, alfombras y kebabs. Sobre esta operación se proporciona la siguiente información:

- La operación de enrollado, cuyo código no se proporciona, se debe asumir que tiene cierta complejidad (unas 100 líneas de código) y que se basa siempre en el uso progresivo e iterativo del método `doblar` del elemento que se desea enrollar.
- El enrollado de un mapa, de un kebab y de una alfombra se hace exactamente de la misma forma (usando, como ya se ha comentado, el método `doblar` de estos elementos).
- Se desea crear las clases `EnrolladorDeMapas`, `EnrolladorDeAlfombras` y `EnrolladorDeKebaps`, de forma que dispongan todas de un **método de instancia** llamado `enrollar` que se aplique a un **parámetro del tipo adecuado en cada caso** (mapa, alfombra y kebab respectivamente).
- Es importante que un `EnrolladorDeMapas` solo permita enrollar mapas, el `EnrolladorDeAlfombras` solo permita enrollar alfombras y el `EnrolladorDeKebaps` solo permita enrollar kebabs. **De esta comprobación debe encargarse el compilador.**
- Es probable que en el futuro aparezcan más elementos `doblables` y que también se tengan que crear clases que permitan enrollarlos.

Proporcionar una implementación, utilizando el lenguaje de programación Java, para las tres clases indicadas. Del método `enrollar`, solo se deberá proporcionar la cabecera completa (incluyendo el parámetro).

Se leerá en Proyecto-Serie