

## Práctica 1. Principios POO

Definir los métodos de las siguientes clases y las relaciones entre ellas:

- Médico: es capaz de curar a las Personas. También puede recetar Medicinas.
- Curandero: también sabe curar pero por otros medios, y no puede recetar.
- Hay tres tipos de especialista: Traumatólogo, Oftalmólogo y Neurólogo.
- Un Oftalmólogo puede graduar la vista de una Persona.

Con la definición anterior realizar los siguientes ejercicios:

1. Si en una clase cualquiera existe un método `m1(Medico m)`, ¿se puede pasar una instancia de Oftalmólogo?

```
Oftalmologo o = new Oftalmologo();  
m1(o);
```

2. ¿Se puede pasar una instancia de Medico?

```
Medico m = new Medico();  
m1(m);
```

3. ¿Se puede pasar una instancia de Curandero?

```
Curandero c = new Curandero();  
m1(c);
```

4. En el método `m1`, ¿se puede invocar el método `curar` sobre `m`?

```
public void m1(Medico m){  
    m.curar();  
}
```

5. En el método `m1`, ¿se puede invocar el método `recetar` sobre `m`?

```
public void m1(Medico m){  
    m.recetar();  
}
```

6. En el método `m1`, ¿se puede invocar el método `graduar` sobre `m`?

```
public void m1(Medico m){  
    m.graduar();  
}
```

7. Si al método m1 se pasa una instancia de Oftalmólogo, ¿se puede invocar el método curar sobre m?

```
public void m1(Medico m){  
    m.curar();  
}  
  
Oftalmologo o = new Oftalmologo();  
m1(o);
```

8. Si al método m1 se pasa una instancia de Oftalmólogo, ¿se puede invocar el método graduar sobre m de alguna forma?

```
public void m1(Medico m){  
    m.graduar();  
}  
  
Oftalmologo o = new Oftalmologo();  
m1(o);
```

9. ¿Se puede asignar una instancia de Oftalmólogo a una variable Médico directamente e invocar a graduar?

```
Medico m = new Oftalmologo();  
m.graduar();
```

10. ¿Se puede asignar una instancia de Médico a una variable Curandero directamente e invocar a curar?

```
Curandero c = new Medico();  
c.curar();
```

11. Imaginando que el método curar es abstracto en la clase Medico. En el método m1, ¿se puede invocar el método curar sobre m?

```
public void m1(Medico m){  
    m.curar();  
}
```

12. Imaginando que el método curar es abstracto en la clase Medico. En el método m1, ¿se puede pasar una instancia de Medico?

```
public void m1(Medico m){  
    m.curar();  
}  
  
Medico m = new Medico();  
m1(m);
```

13. Imaginando que el método curar está definido en un Interfaz implementado por la clase Medico. En el método m1, ¿se puede invocar el método curar sobre m?

```
public void m1(Medico m){  
    m.curar();  
}
```

14. Imaginando que el método curar está definido en un interfaz Curador implementado por Medico y Curandero. ¿Se puede pasar una instancia de Curandero?

```
Public void m1(Medico m){  
    m.curar();  
}  
  
Curandero c = new Curandero();  
m1(c);
```

15. Imaginando que el método curar está definido en un interfaz Curador implementado por Medico y Curandero. En el método m2(Curador c), ¿se puede pasar una instancia de Curandero?

```
public void m2(Curador c){  
    c.curar();  
}  
  
Curandero c = new Curandero();  
m2(c);
```

16. Imaginando que el método curar está definido en un interfaz Curador implementado por Medico y Curandero. En el método m2(Curador c), ¿se puede pasar una instancia de Oftalmologo?

```
public void m2(Curador c){  
    c.curar();  
}  
  
Oftalmologo o = new Oftalmologo();  
m2(o);
```

17. Imaginando que el método curar está definido en un interfaz Curador implementado por Medico y Curandero. En el método m3(Curador c), ¿se puede pasar una instancia de Oftalmologo e invocar a graduar haciendo un casting?

```
public void m3(Curador c){  
    c.curar();  
    Oftalmologo o = (Oftalmologo) c;  
    o.graduar();  
}  
  
Oftalmologo o = new Oftalmologo();  
m3(o);
```