

## 7.F. Polimorfismo.

### 2. Interfaces y polimorfismo.

Es posible también llevar a cabo el **polimorfismo** mediante el uso de **interfaces**. Un objeto puede tener una referencia cuyo tipo sea una **interfaz**, pero para que el compilador te lo permita, la clase cuyo **constructor** se utilice para crear el objeto deberá implementar esa **interfaz** (bien por sí misma o bien porque la implemente alguna **superclase**). Un objeto cuya referencia sea de tipo **interfaz** sólo puede utilizar aquellos métodos definidos en la **interfaz**, es decir, que no podrán utilizarse los atributos y métodos específicos de su clase, tan solo los de la **interfaz**.

Las referencias de tipo **interfaz** permiten unificar de una manera bastante estricta la forma de utilizarse de objetos que pertenezcan a clases muy diferentes (pero que todas ellas implementan la misma **interfaz**). De este modo podrías hacer referencia a diferentes objetos que no tienen ninguna relación jerárquica entre sí utilizando la misma variable (referencia a la **interfaz**). Lo único que los distintos objetos tendrían en común es que implementan la misma **interfaz**. En este caso sólo podrás llamar a los métodos de la **interfaz** y no a los específicos de las clases.

Por ejemplo, si tenías una variable de tipo referencia a la interfaz **Arrancable**, podrías instanciar objetos de tipo **Coche** o **Motosierra** y asignarlos a esa referencia (teniendo en cuenta que ambas clases no tienen una relación de herencia). Sin embargo, tan solo podrás usar en ambos casos los métodos y los atributos de la interfaz **Arrancable** (por ejemplo **arrancar**) y no los de **Coche** o los de **Motosierra** (sólo los genéricos, nunca los específicos).

En el caso de las clases **Persona**, **Alumno** y **Profesor**, podrías declarar, por ejemplo, variables del tipo **Imprimible**:

```
Imprimible obj; // Imprimible es una interfaz y no una clase
```

Con este tipo de referencia podrías luego apuntar a objetos tanto de tipo **Profesor** como de tipo **Alumno**, pues ambos implementan la interfaz **Imprimible**:

```
// En algunas circunstancias podría suceder esto:
```

```
obj= new Alumno (nombre, apellidos, fecha, grupo, nota); // Polimorfismo con interfaces
```

```
...
```

```
// En otras circunstancias podría suceder esto:
```

```
obj= new Profesor (nombre, apellidos, fecha, especialidad, salario); // Polimorfismo con interfaces
```

```
...
```

Y más adelante hacer uso de la **ligadura dinámica**:

```
// Llamadas sólo a métodos de la interfaz
```

```
String contenido;
```

```
contenido= obj.devolverContenidoString(); // Ligadura dinámica con interfaces
```

#### Autoevaluación

El polimorfismo puede hacerse con referencias de superclases abstractas, superclases no abstractas o con interfaces.

¿Verdadero o Falso?

☐ Verdadero

☐ Falso