

## **EJERCICIO 28 TipadOOs Resolución:**

### **1) Códigos validos:**

```
A objeto = new C();  
objeto.mensajeInterface();
```

```
A objeto = new E();  
objeto.mensajeInterface();
```

#### **Justificación:**

A es una interfaz sólo puede referenciar objetos de clases que la implementen (C y E).  
El método disponible es mensajeInterface().

### **2) Códigos validos:**

```
B objeto = new C();  
objeto.mensajeAbstracto();
```

```
B objeto = new D();  
objeto.mensajeAbstracto();
```

#### **Justificación:**

B es una clase abstracta, no instanciable.  
Las clases que la extienden (C y D) implementan mensajeAbstracto().

### **3) Códigos validos:**

```
D objeto = new D();  
objeto.mensajeAbstracto();
```

```
D objeto = new D();  
objeto.mensajeParticular();
```

```
D objeto = new E();  
objeto.mensajeAbstracto();
```

```
D objeto = new E();  
objeto.mensajeParticular();
```

#### **Justificación:**

D es concreta y E hereda de ella, por eso ambos son válidos.  
D define mensajeParticular() e implementa mensajeAbstracto().

#### **4) Códigos validos:**

```
C objeto = new C();  
objeto.mensajeAbstracto();
```

```
C objeto = new C();  
objeto.mensajeInterface();
```

#### **Justificación:**

C implementa la interfaz A y extiende de B, por lo tanto posee ambos métodos (mensajeAbstracto() y mensajeInterface())

#### **5) Códigos validos:**

```
B objeto = new C();  
objeto.mensajeAbstracto();
```

```
C objeto = new C();  
objeto.mensajeAbstracto();
```

#### **Justificación:**

Ambos tipos (B y C) conocen mensajeAbstracto().

C hereda de B, así que la referencia puede ser de tipo B o C.

#### **6) Códigos validos:**

```
A objeto = new C();  
objeto.mensajeInterface();
```

```
C objeto = new C();  
objeto.mensajeInterface();
```

#### **Justificación:**

mensajeInterface() proviene de la interfaz A, que C implementa, por lo tanto ambas combinaciones son correctas.