



## PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

### Pregunta de análisis

Basándose en el código, responda:

#### 1. Indique por qué funcionan o no los métodos sumar:

```
ob1.sumar()  
ob2.sumar()  
ob3.sumar()  
ob4.sumar()
```

Ob1: Porque, tanto la clase Ejecucion1 y Números (clase superior) no definen el método sumar ()

Ob2: Aunque Operadores1 define el método sumar(), Ejecucion2, Operadores1 u Operadores2 no tienen los atributos a y b

Ob3: Porque Ejecucion3 tiene como padres las clases Números y Operadores1

Ob4: Porque Ejecucion4 tiene como padres las clases Números y Operadores3

#### 2. Indique por qué funcionan o no los métodos restar:

```
ob1.restar()  
ob2.restar()  
ob3.restar()  
ob4.restar()
```

ob1: Porque, tanto la clase Ejecución 1 y Números (Su clase padre) no definen el método restar().

ob2: Aunque Operadores2 define el método restar(), ni Ejecución2, ni Operadores1 u Operadores2 (tienen los atributos a y b.

ob3: Aunque Ejecución3 tiene como padre la clase Números, ni la clase Ejecución3, Números u Operadores1 no definen el método restar().

ob4: Porque Ejecución4 tiene como padres las clases Números y Operadores3

#### 3. Indique por qué funcionan o no los métodos multiplicar:

```
ob1.multiplicar()  
ob2.multiplicar()  
ob3.multiplicar()  
ob4.multiplicar()
```



## PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

ob1: Porque, tanto la clase Ejecución 1 y Números (Su clase padre) no definen el método multiplicar().

ob2: Aunque Operadores1 define el método multiplicar(), ni Ejecución2, ni Operadores1 u Operadores2 (Sus clases padres) tienen los atributos a y b.

ob3: Porque Ejecución3 tiene como padres las clases Números y Operadores1 (Por ende, el método, multiplicar()).

ob4: Porque Ejecución4 tiene como padres las clases Números y Operadores3

### 4. Indique por qué funcionan o no los métodos dividir:

```
ob1.dividir()  
ob2.dividir()  
ob3.dividir()  
ob4.dividir()
```

ob1: Porque, tanto la clase Ejecución 1 y Números no definen el método dividir().

ob2: Aunque Operadores2 define el método dividir(), ni Ejecución2, ni Operadores1 u Operadores2 (Sus clases padres) tienen los atributos a y b.

ob3: Aunque Ejecución3 tiene como padre la clase Números, ni la clase Ejecución3, Números u Operadores1 no definen el método dividir().

ob4: Porque Ejecución4 tiene como padres las clases Números y Operadores3

### 5. ¿Qué clase puede existir sin necesidad de que se herede en otra clase?

La respuesta puede variar según la connotación de la pregunta. Si se refiere a que cualquier objeto de esta clase no debe derivar de otra clase (es decir, la clase original no tiene clases padre), entonces serían las clases Números, Operaciones1 y Operaciones2. Si se refiere a que no pueden existir objetos creados con clases distintas a la mencionada (es decir, no hay otras clases que usen esta clase como su padre), entonces serían Ejecución1, Ejecución2, Ejecución3 y Ejecución4. Si se refiere a que no puede haber polimorfismo con un objeto creado a partir de esta clase, entonces no habría ninguna.



## **PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2**