

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

obj1.sumar(): No funciona, porque obj1 es una instancia de Ejecucion1 y hereda únicamente Números, que no posee el método sumar().

obj2.sumar(): No funciona, porque obj2 es una instancia de Ejecucion2 y hereda Operadores1 y Operadores2, entonces si posee el método sumar(), pero no es subclase de números, entonces no posee los atributos a y b para ejecutar los métodos heredados.

obj3.sumar(): Funciona, porque obj3 es una instancia de Ejecucion3 y hereda Operadores1 y Números, entonces si posee el método sumar() y los atributos a y b para ejecutar el metodo.

obj4.sumar(): Funciona, porque obj3 es una instancia de Ejecucion4 y hereda Operadores3 y Números, adicionalmente Operadores3 hereda Operadores1 y Operadores2 por tanto también sus métodos, en este caso sumar(), de Números hereda los atributos a y b para ejecutar el metodo.

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

obj1.restar(): No funciona, porque obj1 es una instancia de Ejecucion1 y hereda únicamente Números, que no posee el método restar().

obj2.restar(): No funciona, porque obj2 es una instancia de Ejecucion2 y hereda Operadores1 y Operadores2, entonces si posee el método restar(), pero no es subclase de números, entonces no posee los atributos a y b para ejecutar los métodos heredados.

obj3.restar(): No funciona, por que obj3 es una instancia de Ejecucion3 y hereda Operadores1 y Números, y Operadores1 no posee el método restar().

obj4.restar(): Funciona, porque obj3 es una instancia de Ejecucion4 y hereda Operadores3 y Números, adicionalmente Operadores3 hereda Operadores1 y Operadores2 por tanto también sus métodos, en este caso restar(), de Números hereda los atributos a y b para ejecutar el metodo.

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

obj1.multiplicar(): No funciona, porque obj1 es una instancia de Ejecucion1 y hereda únicamente Números, que no posee el método multiplicar().

obj2.multiplicar(): No funciona, porque obj2 es una instancia de Ejecucion2 y hereda Operadores1 y Operadores2, entonces si posee el método multiplicar(), pero no es subclase de números, entonces no posee los atributos a y b para ejecutar los métodos heredados.

obj3.multiplicar(): Funciona, porque obj3 es una instancia de Ejecucion3 y hereda Operadores1 y Números, entonces si posee el método multiplicar() y los atributos a y b para ejecutar el metodo.

obj4.multiplicar(): Funciona, porque obj3 es una instancia de Ejecucion4 y hereda Operadores3 y Números, adicionalmente Operadores3 hereda Operadores1 y Operadores2 por tanto también sus métodos, en este caso multiplicar(), de Números hereda los atributos a y b para ejecutar el metodo.

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

obj1.dividir(): No funciona, porque obj1 es una instancia de Ejecucion1 y hereda únicamente Números, que no posee el método dividir().

obj2.dividir(): No funciona, porque obj2 es una instancia de Ejecucion2 y hereda Operadores1 y Operadores2, entonces si posee el método dividir(), pero no es subclase de números, entonces no posee los atributos a y b para ejecutar los métodos heredados.

obj3.dividir(): No funciona, por que obj3 es una instancia de Ejecucion3 y hereda Operadores1 y Números, y Operadores1 no posee el método dividir().

obj4.dividir(): Funciona, porque obj3 es una instancia de Ejecucion4 y hereda Operadores3 y Números, adicionalmente Operadores3 hereda Operadores1 y Operadores2 por tanto también sus métodos, en este caso dividir(), de Números hereda los atributos a y b para ejecutar el metodo.

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

La clase Numeros puede existir sin ser heredada por otra clase porque es completamente funcional por sí sola. Las demás clases dependen de heredar o combinarse con otras clases para funcionar correctamente.

