

[Página Principal](#) / [Mis cursos](#) / [131_15021482_ZSIFC02_MP0485_B](#) / [2. Fundamentos de la programación orientada a objetos](#)
/ [Preguntas CodeRunner para probar clases java](#)

Comenzado el	jueves, 27 de octubre de 2022, 19:27
Estado	Finalizado
Finalizado en	miércoles, 2 de noviembre de 2022, 15:00
Tiempo empleado	5 días 20 horas
Puntos	4,00/4,00
Calificación	10,00 de 10,00 (100%)

Pregunta **1**

Correcta

Se puntúa 1,00
sobre 1,00

Escribe una clase que encapsule las típicas coordenadas x y en un objeto. Las coordenadas siempre son enteras.

Además, la clase tiene los siguientes métodos:

- `sumarCoordenadas()` que devuelve la suma de las coordenadas x e y
- `dividirCoordenadas()` que devuelve la división de las coordenadas x e y

Por ejemplo:

Test	Resultado
<pre>Coordenadas punto= new Coordenadas(); punto.x=3; punto.y=8; System.out.println("(" + punto.x + ", " + punto.y + ")");</pre>	(3,8)
<pre>Coordenadas punto= new Coordenadas(); punto.x=5; punto.y=5; System.out.println(punto.sumarCoordenadas());</pre>	10
<pre>Coordenadas punto= new Coordenadas(); punto.x=5; punto.y=2; System.out.println(punto.sumarCoordenadas());</pre>	7
<pre>Coordenadas punto= new Coordenadas(); punto.x=10; punto.y=8; System.out.println(punto.dividirCoordenadas());</pre>	1.25

Respuesta: (sistema de penalización: 0 %)

El editor Ace no está listo. Tal vez funcione si vuelve a cargar la página.

Falling back to raw text area.

```
class Coordenadas{
    // ATTRIBUTES
    int x;
    int y;

    // METHODS
    public int sumarCoordenadas(){
        return x+y;
    }
    public double dividirCoordenadas(){
        return (double)x / y;
    }
}
```

	Test	Esperado	Se obtuvo	
✓	Coordenadas punto= new Coordenadas(); punto.x=3; punto.y=8; System.out.println("(" + punto.x + ", " + punto.y + ")");	(3,8)	(3,8)	✓
✓	Coordenadas punto= new Coordenadas(); punto.x=5; punto.y=5; System.out.println(punto.sumarCoordenadas());	10	10	✓

	Test	Esperado	Se obtuvo	
✓	Coordenadas punto= new Coordenadas(); punto.x=5; punto.y=2; System.out.println(punto.sumarCoordenadas());	7	7	✓
✓	Coordenadas punto= new Coordenadas(); punto.x=10; punto.y=8; System.out.println(punto.dividirCoordenadas());	1.25	1.25	✓

Todas las pruebas superadas. ✓

Question author's solution (Java):

El editor Ace no está listo. Tal vez funcione si vuelve a cargar la página.

Falling back to raw text area.

```
class Coordenadas{
    int x;
    int y;
    int sumarCoordenadas() {
        return x+y;
    }
    double dividirCoordenadas() {
        return (double)x/y;
    }
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta **2**

Correcta

Se puntúa 1,00
sobre 1,00

Escribe una clase Persona con dos atributos:

- nombre de tipo String
- edad de tipo int

A menudo el enunciado es más extenso y descriptivo, en este caso, simplemente, los constructores y métodos necesarios debes de deducirlos de los casos de prueba ejemplo

Por ejemplo:

Test	Resultado
<pre>Persona p1 = new Persona("Yo", 55); System.out.println("p1 es " + p1.nombre + " y su edad es " + p1.edad);</pre>	p1 es Yo y su edad es 55
<pre>Persona miAmigo = new Persona("kosky", 47); System.out.println(miAmigo.getNombre() + ", el año que viene tendrás " + (miAmigo.getEdad()+1));</pre>	kosky, el año que viene tendrás 48
<pre>Persona miAmigo = new Persona("kosky", 47); Persona copiaDeMiAmigo = new Persona(miAmigo); System.out.println(copiaDeMiAmigo.getNombre()+", "+copiaDeMiAmigo.getEdad());</pre>	kosky, 47
<pre>Persona miAmigo = new Persona("chischas", 47); System.out.println(miAmigo);</pre>	chischas 47

Respuesta: (sistema de penalización: 0 %)

El editor Ace no está listo. Tal vez funcione si vuelve a cargar la página.

Falling back to raw text area.

```
class Persona{
    // ATTRIBUTES
    public String nombre;
    public int edad;

    // GETTER / SETTER
    public String getNombre(){
        return this.nombre;
    }
    public void setNombre(String nombre){
        this.nombre = nombre;
    }
    public int getEdad(){
        return this.edad;
    }
    public void setEdad(int edad){
        this.edad = edad;
    }
}
```

	Test	Esperado	Se obtuvo	
✓	Persona p1 = new Persona("Yo", 55); System.out.println("p1 es " + p1.nombre + " y su edad es " + p1.edad);	p1 es Yo y su edad es 55	p1 es Yo y su edad es 55	✓
✓	Persona miAmigo = new Persona("kosky", 47); System.out.println(miAmigo.getNombre() + ", el año que viene tendrás " + (miAmigo.getEdad()+1));	kosky, el año que viene tendrás 48	kosky, el año que viene tendrás 48	✓

	Test	Esperado	Se obtuvo	
✓	<pre>Persona miAmigo = new Persona("kosky", 47); Persona copiaDeMiAmigo = new Persona(miAmigo); System.out.println(copiaDeMiAmigo.getNombre()+", "+copiaDeMiAmigo.getEdad());</pre>	kosky, 47	kosky, 47	✓
✓	<pre>Persona miAmigo = new Persona("chischas", 47); System.out.println(miAmigo);</pre>	chischas 47	chischas 47	✓

Todas las pruebas superadas. ✓

Question author's solution (Java):

El editor Ace no está listo. Tal vez funcione si vuelve a cargar la página.

Falling back to raw text area.

```
class Persona {
    String nombre;
    int edad;

    Persona(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
    Persona(Persona p) {
        this.nombre = p.nombre;
        this.edad = p.edad;
    }

    String getNombre() {
        return nombre;
    }
    int getEdad() {
        return edad;
    }
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta **3**

Correcta

Se puntúa 1,00
sobre 1,00

LA CLASE CUENTA

Crea una clase Cuenta con los siguientes atributos privados:

- String numeroCuenta;
- String titular;
- double saldo;

Observa casos de prueba ejemplo para deducir constructores y métodos

Por ejemplo:

Test	Resultado
<pre>Cuenta c1 = new Cuenta("111-222", "Epi", 50.0); System.out.println("el saldo inicial de Epi es: " + c1.getSaldo());</pre>	el saldo inicial de Epi es: 50.0
<pre>Cuenta c2 = new Cuenta("999-888", "Blas", 100.0); System.out.println("Datos de la cuenta c2: " + c2);</pre>	Datos de la cuenta c2: (999-888, Blas, 100.0)
<pre>Cuenta c1 = new Cuenta("111-222", "Epi", 50.0); c1.setSaldo(250.0); System.out.println("el nuevo saldo de Epi es: " + c1.getSaldo());</pre>	el nuevo saldo de Epi es: 250.0
<pre>Cuenta c3 = new Cuenta(); System.out.println("datos de cuenta creada sin parámetros: " + c3);</pre>	datos de cuenta creada sin parámetros: (sin numero, sin titular, 0.0)

Respuesta: (sistema de penalización: 0 %)

El editor Ace no está listo. Tal vez funcione si vuelve a cargar la página.

Falling back to raw text area.

```
class Cuenta{
    // ATTRIBUTES
    private String numeroCuenta;
    private String titular;
    private double saldo;

    // GETTER / SETTER
    public String getNumeroCuenta(){
        return this.numeroCuenta;
    }
    public void setNumeroCuenta(String numeroCuenta){
        this.numeroCuenta = numeroCuenta;
    }
    public String getTitular(){
        return this.titular;
    }
    public void setTitular(String titular){
        this.titular = titular;
    }
}
```

	Test	Esperado	Se obtuvo	
--	-------------	-----------------	------------------	--

	Test	Esperado	Se obtuvo	
✓	<pre>try { Field f = Cuenta.class.getDeclaredField("numeroCuenta"); System.out.println(Modifier.isPrivate(f.getModifiers())?"numeroCuenta definido como private OK":"ERROR atributo numeroCuenta no es private"); } catch(NoSuchFieldException e) { System.out.println("no existe atributo numeroCuenta"); }</pre>	numeroCuenta definido como private OK	numeroCuenta definido como private OK	✓
✓	<pre>try { Field f = Cuenta.class.getDeclaredField("titular"); System.out.println(Modifier.isPrivate(f.getModifiers())?"titular definido como private OK":"ERROR atributo titular no es private"); } catch(NoSuchFieldException e) { System.out.println("no existe atributo titular"); }</pre>	titular definido como private OK	titular definido como private OK	✓
✓	<pre>try { Field f = Cuenta.class.getDeclaredField("saldo"); System.out.println(Modifier.isPrivate(f.getModifiers())?"saldo definido como private OK":"ERROR atributo saldo no es private"); } catch(NoSuchFieldException e) { System.out.println("no existe atributo saldo"); }</pre>	saldo definido como private OK	saldo definido como private OK	✓
✓	<pre>Cuenta c1 = new Cuenta("111-222", "Epi", 50.0); System.out.println("el saldo inicial de Epi es: " + c1.getSaldo());</pre>	el saldo inicial de Epi es: 50.0	el saldo inicial de Epi es: 50.0	✓

	Test	Esperado	Se obtuvo	
✓	Cuenta c2 = new Cuenta("999-888", "Blas", 100.0); System.out.println("Datos de la cuenta c2: " + c2);	Datos de la cuenta c2: (999-888, Blas, 100.0)	Datos de la cuenta c2: (999-888, Blas, 100.0)	✓
✓	Cuenta c1 = new Cuenta("111-222", "Epi", 50.0); c1.setSaldo(250.0); System.out.println("el nuevo saldo de Epi es: " + c1.getSaldo());	el nuevo saldo de Epi es: 250.0	el nuevo saldo de Epi es: 250.0	✓
✓	Cuenta c3 = new Cuenta(); System.out.println("datos de cuenta creada sin parámetros: " + c3);	datos de cuenta creada sin parámetros: (sin numero, sin titular, 0.0)	datos de cuenta creada sin parámetros: (sin numero, sin titular, 0.0)	✓

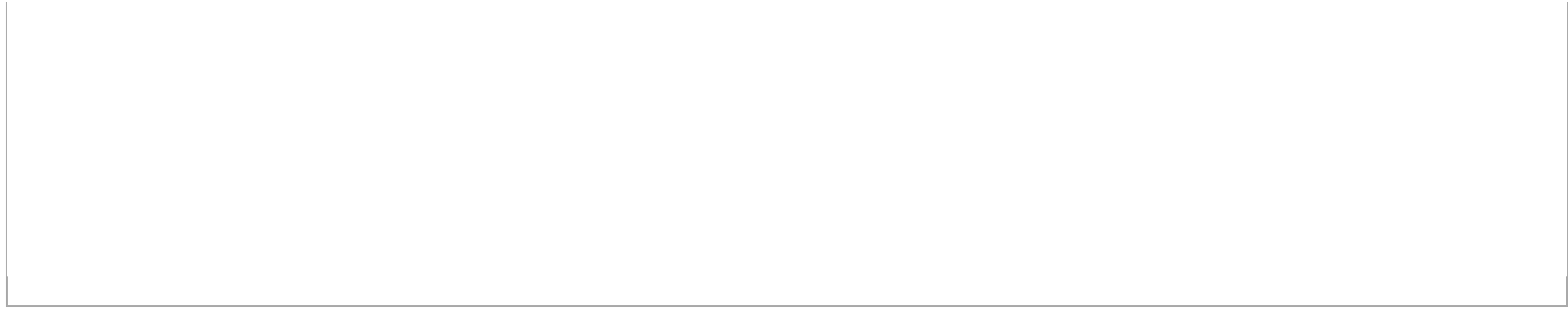
Todas las pruebas superadas. ✓

Question author's solution (Java):

```

1 class Cuenta {
2     private String numeroCuenta;
3     private String titular;
4     private double saldo;
5
6     Cuenta(String numeroCuenta, String titular, double saldo) {
7         this.numeroCuenta = numeroCuenta;
8         this.titular = titular;
9         this.saldo = saldo;
10    }
11
12    Cuenta() {
13        this("sin numero", "sin titular", 0.0);
14    }
15

```



Correcta

Puntos para este envío: 1,00/1,00.

Pregunta **4**

Correcta

Se puntúa 1,00
sobre 1,00

LA CLASE COMPLEJO

Seguimos familiarizándonos con Code Runner. Ahora probamos la clase Complejo para lo que nos remitimos al boletín correspondiente.

un número complejo tiene una parte real a y una imaginaria b . Tiene varias representaciones pero la más sencilla es al "binómica": $a + bi$

Escribe la siguiente clase Complejo



Además a la descripción gráfica de arriba añade las siguientes consideraciones:

- Los atributos son private
- `Complejo()`: Constructor que inicializa los atributos a cero.
- `toString()`: Convierte a String el número complejo, mediante la concatenación de los valores de sus atributos. Ejemplo, si 3 es la parte real y 4 la parte imaginaria. devuelve como resultado la cadena de texto `3 + 4i`. Otro ejemplo, si 3 es la parte real y -4 la parte imaginaria. devuelve como resultado la cadena de texto `3 - 4i`.
- `sumar(Complejo b)`: Devuelve un nuevo número complejo resultado de sumar el número complejo "this" con el número

complejo b. Para esto calcula la parte real del nuevo número sumando la parte real de this y b y calcula la imaginaria sumando la imaginaria de this y b. Ej. $2 + 3i$ sumado con $2 + 1i$ es $4 + 4i$

Por ejemplo:

Test	Resultado
Complejo a= new Complejo(1.1,2.2); Complejo b= new Complejo(3.3,0.0); Complejo c= a.sumar(b); System.out.println(c);	4.4 + 2.2i
Complejo a= new Complejo(8,-9); System.out.println(a);	8.0 -9.0i
Complejo a = new Complejo(8, -9); Complejo b = new Complejo(); Complejo c = b.sumar(a); System.out.println(c);	8.0 -9.0i

Respuesta: (sistema de penalización: 0 %)

```
1 class Complejo {  
2     // ATTRIBUTES  
3     double real;  
4     double imag;  
5  
6     // GETTER / SETTER  
7     public double getReal() {  
8         return this.real;  
9     }  
10  
11     public void setReal(double real) {  
12         this.real = real;  
13     }  
14 }
```

	Test	Esperado	Se obtuvo	
✓	Complejo a= new Complejo(1.1,2.2); Complejo b= new Complejo(3.3,0.0); Complejo c= a.sumar(b); System.out.println(c);	4.4 + 2.2i	4.4 + 2.2i	✓
✓	Complejo a= new Complejo(8,-9); System.out.println(a);	8.0 -9.0i	8.0 -9.0i	✓
✓	Complejo a = new Complejo(8, -9); Complejo b = new Complejo(); Complejo c = b.sumar(a); System.out.println(c);	8.0 -9.0i	8.0 -9.0i	✓

Todas las pruebas superadas. ✓

Question author's solution (Java):

```
1 class Complejo {  
2     private double real;  
3     private double imag;  
4  
5     Complejo() {  
6         this.real = 0;
```




Correcta

Puntos para este envío: 1,00/1,00.

[◀ Soluciones Unidad2](#)

Ir a...

[Tarea 1 Unidad2 ▶](#)