# **Evaluación Teórica – Clases y objetos**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Realice esta evaluación y logre obtener puntos de **¡Experiencia, Competencia y Habilidad!** ¡Adelante!

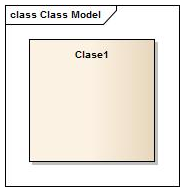
Este desafío busca evaluar la comprensión e interpretación que usted ha logrado de los contenidos del módulo 1.

Para poder responder estas preguntas, idealmente debe revisar los contenidos teóricos y prácticos presentados en las primeras secciones en donde se explican los conceptos básicos de clases y objetos. La estructura de esta evaluación es de selección múltiple, en la que usted deberá:

* Analizar varias implementaciones de clases y objetos en sintaxis Java simple.
* Responder preguntas de análisis basadas en el código presentado y en diagramas UML.

El desafío consta de 13 preguntas. Cada una tiene varias opciones de respuesta pero sólo una es correcta. Al final recibirá la retroalimentación de su desempeño.

Considere el siguiente modelo:



La forma correcta de declarar la clase Clase1 es:

1 punto

public class Clase1{ }

class public Clase1 ( ) {}

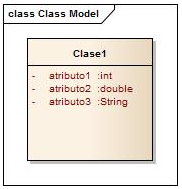
public class Clase1{ };

class public Clase1{}

### 2.

Pregunta 2

Considere el siguiente modelo:



La forma correcta de declarar el atributo atributo1 es:

1 punto

int private atributo1;

private atributo1 int

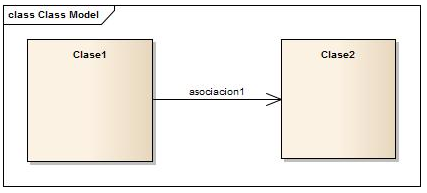
private int atributo1;

private int atributo1

### 3.

Pregunta 3

Considere el siguiente modelo:



¿En qué clase o clases debe ser declarada la asociación llamada asociacion1?

1 punto

En ambas Clase1 y Clase2

En cualquiera Clase1 y Clase2

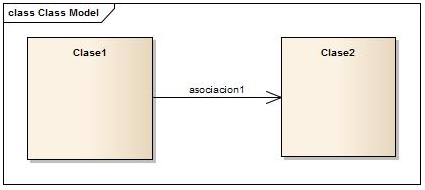
Clase1

**Clase2**

### 4.

Pregunta 4

Considere el siguiente modelo:



¿Cuál es el tipo de la asociación llamada asociacion1?

1 punto

**Clase1**

Class

Clase2

int

### 5.

Pregunta 5

Considere el siguiente código:

public \_\_\_\_\_\_\_\_\_\_ metodo1( double x )

{

int z = 3;

double y = 5.0;

y += x;

return z\*2;

}

¿Cuál es el tipo de retorno del método metodo1?

1 punto

double

int

String

void

### 6.

Pregunta 6

Considere el siguiente código:

public double metodo2()

{

       double x = 5.0;

       x = 7.3;

       return x;

}

En el cuerpo del método metodo2, las instrucciones realizadas, en orden, son:

1 punto

* Creación de variable
* Retorno
* Asignación
* Creación de variable
* Asignación
* Retorno
* Asignación
* Creación de variable
* Retorno
* Retorno
* Creación de variable
* Asignación

### 7.

Pregunta 7

Considere el siguiente código en el lenguaje Java:

public class Clase {

public int metodo1( int a ) {

             a = a \* 3;

             return a;

      }

      public int metodo2( )

      {

             int a = 5;

             a = a / 2;

             return a;

      }

      public int metodo3( )

      {

             int a = 4;

             metodo2( );

             metodo1( a );

}

}

¿Qué valor devolverá el método metodo3?

1 punto

12 6 2 4

### 8.

Pregunta 8

Considere el siguiente código:

public static String metodo1( )

{

       int a = 7;

       String b = "cadena concatenada + a";

       return b;

}

¿Qué valor devolverá el método metodo1?

1 punto

“cadena concatenada a”

“cadena concatenada + a”

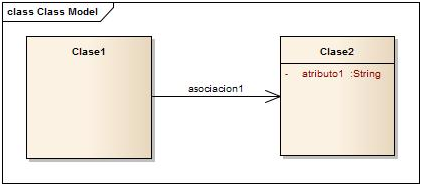
“cadena concatenada 7”

“cadena concatenada + 7”

### 9.

Pregunta 9

Considere el siguiente modelo:



La declaración correcta de las clases Clase1 y Clase2 es:

1 punto

public class Clase1{ private Clase2 asociacion1; }

public class Clase2{ private String atributo1; }

**public class Clase1 { }**

**public class Clase2 { private Clase1 asociacion1; private String atributo1; }**

public class Clase1{private String atributo1;}

public class Clase2{private Clase1 asociacion1;}

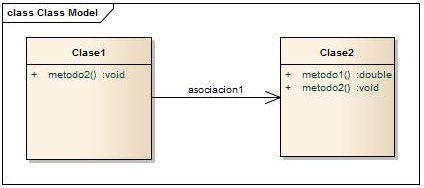
public class Clase1{}

public class Clase2{}

### 10.

Pregunta 10

Considere el siguiente modelo:



Si en el método metodo2 de la clase Clase1, se quiere usar el método metodo1 de la Clase2, se debe usar la siguiente instrucción:

1 punto

**Clase2.metodo1( );**

metodo1( );

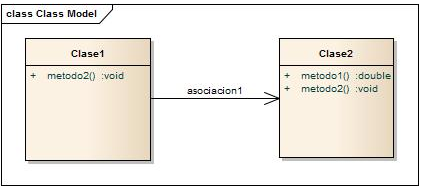
**Clase1.metodo1( );**

asociacion1.metodo1( );

### 11.

Pregunta 11

Considere el siguiente modelo:



Si el método metodo2 de la clase Clase2, quiere usar el método metodo1 de la Clase2, ¿Cómo lo debe llamar?

1 punto

Clase2.metodo1( );

metodo1( );

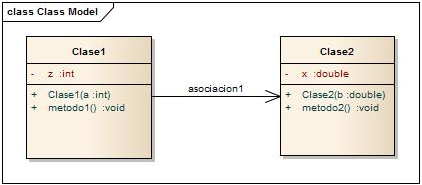
asociacion1.metodo1( );

Clase1.metodo1( );

### 12.

Pregunta 12

Considere el siguiente modelo, y el siguiente fragmento de declaración de las clases del modelo:



public class Clase1{

       private int z;

       public Clase1( int a ) {

              z = a;

        }

}

public class Clase2 {

        private double x;

       public Clase2( double b ) {

             x = b;

       }

}

Si el método metodo1 de la clase Clase1 quiere crear un objeto de la clase Clase2, llamado asociación1, e inicializar su atributo x con un valor de 6.3. La instrucción que debería usar es:

1 punto

Clase2 asociacion1 = new asociacion1.Clase2( 6.3 );

asociacion1 = new Clase2( 6.3 );

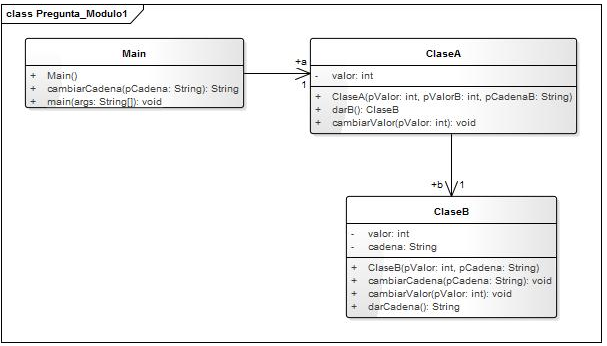
Clase2 asociacion1 = new Clase2( 6.3 );

asociacion = new asociacion1.Clase2( 6.3 );

### 13.

Pregunta 13

Considere el siguiente modelo del mundo:



Se necesita completar el siguiente código con el objetivo de implementar un método que cambie la cadena del objeto de la ClaseB desde la clase Main y devuelva su valor. Para esto:

* Revise la implementación de las clases Main, ClaseA y ClaseB que se encuentra en el siguiente recuadro.
* Piense cómo completar el método cambiar Cadena de la clase Main.

// Clase Main

public class Main

{

private ClaseA a;

public Main()

{

a = new ClaseA( 1, 2, "Hola Mundo!!" );

//La siguiente instrucción imprime en la consola de eclipse

//la cadena que se pasa como parámetro a

//System.out.println

System.out.println( cambiarCadena( "La nueva cadena" ) );

}

public String cambiarCadena( String pCadena )

{

//Aquí van las instrucciones necesarias para completar

//el método cambiarCadena el cual cambia la cadena

//del objeto de la ClaseB y retorna su nuevo valor.

}

}

// Clase ClaseA

public class ClaseA

{

private int valor;

private ClaseB b;

public ClaseA( int pValor, int pValorB, String pCadenaB )

{

valor = pValor;

b = new ClaseB( pValorB, pCadenaB );

}

public ClaseB darB( )

{

return b;

}

public void cambiarValor( int pValor )

{

valor = pValor;

}

}

// Clase ClaseB

public class ClaseB

{

private int valor;

private String cadena;

public ClaseB( int pValor, String pCadena )

{

valor = pValor;

cadena = pCadena;

}

public String darCadena( )

{

return cadena;

}

public void cambiarValor( int pValor )

{

valor = pValor;

}

public void cambiarCadena( String pCadena )

{

cadena = pCadena;

}

}

¿Cuál sería un código adecuado para implementar el método cambiarCadena de la clase Main?

1 punto

a.b.cambiarCadena( pCadena );

return a.b.darCadena();

cambiarCadena( pCadena );

return a.darB().darCadena();

a.darB().cambiarCadena(pCadena);

return “La nueva Cadena”;

a.darB( ).cambiarCadena( pCadena );

return a.darB().darCadena();