

Navegación por el cuestionario

1	2	3	4	5	6	7	8	9	10
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
11	12	13	14	15					
✓	✓	✓	✓	✓					

Finalizar revisión

Comenzado el	miércoles, 23 de enero de 2019, 17:42
Estado	Finalizado
Finalizado en	jueves, 24 de enero de 2019, 14:17
Tiempo empleado	20 horas 34 minutos
Puntos	13,03/15,00
Calificación	8,69 de 10,00 (87%)

Pregunta 1

Correcta

Puntuá 1,00 sobre 1,00

∨ Marcar pregunta

Si dispongo de una clase "c" con un método "m", y el objeto "obj" es una instancia de la clase "c", ¿cómo podría acceder al método "m" del objeto "obj"?

Seleccione una:

a. Mediante el operador flecha: obj->m().

b. Mediante el operador corchete: obj[m()].

c. Mediante el operador punto: obj.m(). ✓

d. Mediante el operador mayor: obj>m().

La respuesta correcta es: Mediante el operador punto: obj.m().

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 2

Correcta

Puntuá 1,00 sobre 1,00

∨ Marcar pregunta

¿Qué modificadores habría que añadir a un método que se desea que sea accesible desde fuera de la clase y que no haga falta que existan objetos de la clase para poder ser utilizado?

Seleccione una:

a. `protected final`.

b. `private static`.

c. `public static`. ✓

d. `protected`.

La respuesta correcta es: `public static`.

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 3

Correcta

Puntuá 1,00 sobre 1,00

∨ Marcar pregunta

¿Qué tipo es devuelto por un constructor?

Seleccione una:

a. Depende de si el constructor está sobrecargado o no.

b. `void`.

c. No devuelve ningún tipo (ni siquiera `void`). ✓

d. El mismo tipo que el atributo principal de la clase.

La respuesta correcta es: No devuelve ningún tipo (ni siquiera `void`).

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 4

Correcta

Puntuá 1,00 sobre 1,00

∨ Marcar pregunta

¿Cuál de los siguientes modificadores no es aplicable a un atributo?

Seleccione una:

a. `public`.

b. `private`.

c. `extern`. ✓

d. `protected`.

La respuesta correcta es: `extern`.

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 5

Correcta

Puntuá 1,00 sobre 1,00

∨ Marcar pregunta

¿Qué afirmaciones se corresponden con características de los métodos consultores (get) y modificadores (set)?

Seleccione una o más de una:

a. Todos los objetos tienen un método set y otro get.

b. Los métodos get siempre acabarán con la sentencia return. ✓

c. Los métodos consultores normalmente empiezan por set y los modificadores por get.

d. Los métodos modificadores siempre acabarán con la sentencia return.

e. Los métodos set permiten modificar el valor de los atributos de un objeto. ✓

f. Los métodos consultores permiten acceder al valor de los atributos de un objeto. ✓

Respuesta correcta

La respuesta correcta es: Los métodos set permiten modificar el valor de los atributos de un objeto., Los métodos consultores permiten acceder al valor de los atributos de un objeto., Los métodos get siempre acabarán con la sentencia return.

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 6

Correcta

Puntuá 0,67 sobre 1,00

∨ Marcar pregunta

Indica las afirmaciones que caracterizan a los constructores de las clases:

Seleccione una o más de una:

a. Los constructores consumen mucha memoria y se deben usar con cuidado

b. Los constructores son métodos especiales que crean objetos e inicializan su estado ✓

c. El constructor es el método que usamos con la sentencia new ✓

d. Los constructores se consideran una mala práctica en programación y deben evitarse

e. Un constructor se crea siempre sin sentencia return ✓

f. Las clase solo pueden tener un único constructor

Respuesta correcta

La respuesta correcta es: Los constructores son métodos especiales que crean objetos e inicializan su estado, Un constructor se crea siempre sin sentencia return, El constructor es el método que usamos con la sentencia new

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,67/1,00.

Pregunta 7

Correcta

¿Por qué se recomienda dejar el código de entrada y salida por consola fuera de las clases?

Puntúa 1,00 sobre  
1,00

▼ Marcar  
pregunta

Seleccione una:

- a. Porque es más eficiente agrupar todas las operaciones de E/S en una misma clase
- b. Para evitar las excepciones dentro de las clases
- c. Porque dentro de las clases no se pueden realizar operaciones de E/S
- d. Para mejorar su reusabilidad ya que los mensajes para pedir y mostrar datos dependerán de la aplicación deseada ✓

Respuesta correcta

La respuesta correcta es: Para mejorar su reusabilidad ya que los mensajes para pedir y mostrar datos dependerán de la aplicación deseada

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 8

Correcta

Puntúa 0,33 sobre  
1,00

▼ Marcar  
pregunta

Si creamos un objeto de la clase String ¿cuál será su estado y cuál su comportamiento?

Seleccione una:

- a. El estado son los métodos que devuelven algún valor como length o charAt y el comportamiento los que modifican el valor del String como toUpperCase.
- b. El String en sí sería el estado y lo que haces con él, por ejemplo guardarlo en un array, sería su comportamiento.
- c. El estado sería la memoria que ocupa el objeto y el comportamiento lo que haces con esa memoria, por ejemplo almacenar caracteres.
- d. El estado sería la cadena de caracteres que contiene y el comportamiento serían los métodos que puedes utilizar como indexOf. ✓

Respuesta correcta

La respuesta correcta es: El estado sería la cadena de caracteres que contiene y el comportamiento serían los métodos que puedes utilizar como indexOf.

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,33/1,00.

Pregunta 9

Correcta

Puntúa 0,67 sobre  
1,00

▼ Marcar  
pregunta

¿Cuál es la diferencia entre los objetos y las clases?

Seleccione una:

- a. Un objeto es un tipo de clase sin comportamiento, sin métodos.
- b. Un objeto es una entidad que encapsula datos y comportamientos, mientras que una clase es la plantilla para un tipo de objetos. ✓
- c. Una clase es una instancia de un objeto. Un objeto se puede usar para crear muchas clases.
- d. Los objetos se utilizan en la programación orientada a objetos mientras que las clases se usan en la programación orientada a clases.

Respuesta correcta

La respuesta correcta es: Un objeto es una entidad que encapsula datos y comportamientos, mientras que una clase es la plantilla para un tipo de objetos.

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,67/1,00.

Pregunta 10

Correcta

Puntúa 0,67 sobre  
1,00

▼ Marcar  
pregunta

Indica las afirmaciones que se refieren a la palabra reservada "this".

Seleccione una o más de una:

- a. Permite llamar a un constructor desde otro constructor de la misma clase ✓
- b. Se utiliza para acceder a los atributos y métodos de un objeto desde otro ✓
- c. Es imprescindible cuando tenemos más de un constructor
- d. Se debe usar siempre para inicializar los atributos de una clase
- e. Permite referenciar a los métodos miembro dentro de una clase ✓
- f. Permite referenciar a las variables miembro dentro de una clase ✓

Respuesta correcta

La respuesta correcta es: Permite referenciar a las variables miembro dentro de una clase, Permite referenciar a los métodos miembro dentro de una clase, Permite llamar a un constructor desde otro constructor de la misma clase

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,67/1,00.

Pregunta 11

Correcta

Puntúa 0,80 sobre  
1,00

▼ Marcar  
pregunta

Completa la clase Triangulo. Los datos que tiene que guardar son:

- Base
- Altura

Sus atributos NO serán accesibles desde otros programas y tendrá tantos métodos consultores y modificadores como sea necesario para manejar todos sus valores. Se identificarán con get y set más el atributo correspondiente.

Respuesta: (penalty regime: 10, 20, ... %)

Reiniciar respuesta

```
1 | public class Triangulo {  
2 |     private double base;  
3 |     private double altura;  
4 |  
5 |     public void setBase(double base) {  
6 |         this.base = base;  
7 |     }  
8 |  
9 |     public void setAltura(double altura) {  
10 |         this.altura = altura;  
11 |     }  
12 |  
13 |     public double getBase() {  
14 |         return base;  
15 |     }  
16 |  
17 |     public double getAltura() {  
18 |         return altura;  
19 |     }  
20 | }
```

Test	Expected	Got
✓ Triangulo miTriangulo = new Triangulo(); miTriangulo.setBase(5.2); miTriangulo.setAltura(7.8); System.out.println(miTriangulo.getBase() + " " + miTriangulo.getAltura());	5.2 7.8	5.2 7.8 ✓
✓ Triangulo miTriangulo = new Triangulo(); miTriangulo.setBase(2.5); miTriangulo.setAltura(8.7); System.out.println(miTriangulo.getBase() + " " + miTriangulo.getAltura());	2.5 8.7	2.5 8.7 ✓

Todas las pruebas superadas. ✓

Question author's solution:

```
public class Triangulo {  
    private double base;  
    private double altura;  
  
    public void setBase(double base) {  
        this.base = base;  
    }  
  
    public void setAltura(double altura) {  
        this.altura = altura;  
    }  
  
    public double getBase() {  
        return base;  
    }  
  
    public double getAltura() {  
        return altura;  
    }  
}
```

```

    }
    public double getAltura() {
        return altura;
    }
}

```

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,80/1,00.

Pregunta 12

Correcta

Puntuá 1,00 sobre  
1,00

▼ Marcar  
pregunta

Tenemos el siguiente código:

```

Coche miCoche = new Coche("Ford", "Fiesta", 2010);
System.out.println(miCoche);

```

que mostrará por consola lo siguiente:

Ford Fiesta (2010)

Completa la clase Coche con estos 3 atributos privados:

- Marca
- Modelo
- Año

también el método constructor y el método toString necesarios para que el código anterior se ejecute correctamente.

Respuesta: (penalty regime: 10, 20, ... %)

[Reiniciar respuesta](#)

```

1 | public class Coche {
2 |     private String marca;
3 |     private String modelo;
4 |     private int year;
5 |
6 |     Coche(String marca, String modelo, int year) {
7 |         this.marca = marca;
8 |         this.modelo = modelo;
9 |         this.year = year;
10 |
11 |
12 |     public String toString() {
13 |         return marca + " " + modelo + " (" + year + ")";
14 |     }
15 |

```

Test	Expected	Got	
✓ Coche miCoche = new Coche("Ford", "Fiesta", 2010); System.out.println(miCoche);	Ford Fiesta (2010)	Ford Fiesta (2010)	✓
✓ Coche miCoche = new Coche("Opel", "Corsa", 2014); System.out.println(miCoche);	Opel Corsa (2014)	Opel Corsa (2014)	✓

Todas las pruebas superadas. ✓

Question author's solution:

```

public class Coche {
    private String marca;
    private String modelo;
    private int year;

    public Coche(String marca, String modelo, int year) {
        this.marca = marca;
        this.modelo = modelo;
        this.year = year;
    }

    public String toString() {
        return marca + " " + modelo + " (" + year + ")";
    }
}

```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 13

Correcta

Puntuá 0,90 sobre  
1,00

▼ Marcar  
pregunta

Tenemos la siguiente clase:

```

public class Piso {
    public String id;
    public int superficie;
    public double precioMetro;
}

```

Completa el programa para que muestre lo siguiente:

```

A25C(15 euros/m^2): 55.5 metros cuadrados
Es muy caro

```

Respuesta: (penalty regime: 10, 20, ... %)

[Reiniciar respuesta](#)

```

1 | public class PisoMain {
2 |     public static void main(String[] args) {
3 |         Piso piso1 = new Piso();
4 |         String id = "A25C";
5 |         double superficie = 55.5;
6 |         int precioMetro = 15;
7 |         System.out.println(id + "(" + precioMetro + " euros/m^2): " + superficie + " metros cuadrados");
8 |         double precioTotal = superficie * precioMetro;
9 |         if (precioTotal > 600) {
10 |             System.out.println("Es muy caro");
11 |         }
12 |     }
13 |

```

Expected	Got	
✓ A25C(15 euros/m^2): 55.5 metros cuadrados Es muy caro	A25C(15 euros/m^2): 55.5 metros cuadrados Es muy caro	✓

Todas las pruebas superadas. ✓

Question author's solution:

```

public class PisoMain {
    public static void main(String[] args) {
        Piso piso1 = new Piso();
        piso1.id = "A25C";
    }
}

```

```

piso1.superficie = 55.5;
piso1.precioMetro = 15;
System.out.println(piso1.id + "(" + piso1.precioMetro + " euros/m^2): " + piso1.superficie + " metros cuadrados");
if (precioTotal > 600) {
    System.out.println("Es muy caro");
}
}

```

**Correcta**

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,90/1,00.

**Pregunta 14**

Correcta

Puntúa 1,00 sobre  
1,00

▼ Marcar  
pregunta

Completa la clase Rectangulo para que funcione correctamente el siguiente código:

```

Rectangulo miRectangulo = new Rectangulo(5.0, 2.0);
System.out.println(miRectangulo);

```

Deberá mostrar:

```
Rectangulo: 5.0 x 2.0
```

**Respuesta:** (penalty regime: 10, 20, ... %)

**Reiniciar respuesta**

```

1 public class Rectangulo {
2     public double base;
3     public double altura;
4
5     public Rectangulo(double base, double altura) {
6         this.base = base;
7         this.altura = altura;
8     }
9
10    public String toString () {
11        return "Rectangulo: " + base + " x " + altura;
12    }
13 }

```

Todas las pruebas superadas. ✓

Question author's solution:

```

public class Rectangulo {
    private double base;
    private double altura;

    public Rectangulo(double unaBase, double unaAltura) {
        base = unaBase;
        altura = unaAltura;
    }

    public String toString () {
        return "Rectangulo: " + base + " x " + altura;
    }
}

```

**Correcta**

Puntos para este envío: 1,00/1,00.

**Pregunta 15**

Correcta

Puntúa 1,00 sobre  
1,00

▼ Marcar  
pregunta

Tenemos la clase Cuenta con los siguientes métodos:

```

// Constructores
public Cuenta()
public Cuenta(int numero, double saldo, String titular)

// Modificadores y consultores
public void setNumero(int numero)
public void setSaldo(double saldo)
public void setTitular(String titular)
public int getNumero()
public double getSaldo()
public String getTitular()

// Devuelve un String con el siguiente formato: titular (numero): saldo euros
public String toString()

// Incrementará el saldo en la cantidad indicada y devolverá el total obtenido
public double ingresar(double cantidad)

// Si hay saldo suficiente reducirá el saldo en la cantidad indicada y devolverá true
// En caso contrario devolverá false.
public boolean retirar(double cantidad)

```

Corrige el siguiente programa para que muestre lo siguiente:

```

(0): 0.0 euros
1234 Pedro Arana 1000.67
Ana Albizu(22): 500.5 euros
Ingresar: 1000.5 Saldo: 1000.5
Retirar: true Saldo: 500.5
Retirar: false Saldo: 500.5
Ana Albizu(22): 500.5 euros

```

**Respuesta:** (penalty regime: 10, 20, ... %)

**Reiniciar respuesta**

```

1 public class CuentaMain {
2     public static void main(String[] args) {
3         //int numero;
4         //double saldo;
5         //String titular;
6         // Crea una cuenta vacía y la muestra
7         Cuenta miCuenta = new Cuenta();
8         System.out.println(miCuenta.toString());
9
10        // Modifica los valores de miCuenta
11        miCuenta.setTitular("Pedro Arana");
12        miCuenta.setSaldo(1000.67);
13        miCuenta.setNumero(1234);
14
15        // Muestra los valores de miCuenta
16        System.out.println(miCuenta.getNumero() + " " + miCuenta.getTitular() + " " + miCuenta.getSaldo());
17
18        // Crea y muestra otra cuenta con los siguientes valores 22, 500.5, "Ana Albizu"
19        Cuenta otraCuenta = new Cuenta(22, 500.5, "Ana Albizu");

```

Expected	Got
✓ (0): 0.0 euros 1234 Pedro Arana 1000.67 Ana Albizu(22): 500.5 euros Ingresa: 1000.5 Saldo: 1000.5 Retira: true Saldo: 500.5 Retira: false Saldo: 500.5 Ana Albizu(22): 500.5 euros	(0): 0.0 euros 1234 Pedro Arana 1000.67 Ana Albizu(22): 500.5 euros Ingresa: 1000.5 Saldo: 1000.5 Retira: true Saldo: 500.5 Retira: false Saldo: 500.5 Ana Albizu(22): 500.5 euros

Todas las pruebas superadas. ✓

Question author's solution:

```

public class CuentaMain {
    public static void main(String[] args) {
        // Crea una cuenta vacia y la muestra
        Cuenta miCuenta = new Cuenta();
        System.out.println(miCuenta);

        // Modifica los valores de miCuenta
        miCuenta.setTitular("Pedro Arana");
        miCuenta.setSaldo(1000.67);
        miCuenta.setNumero(1234);

        // Muestra los valores de miCuenta
        System.out.println(miCuenta.getNumero() + " " + miCuenta.getTitular() + " " + miCuenta.getSaldo());

        // Crea y muestra otra cuenta con los siguientes valores 22, 500.5, "Ana Albizu"
        Cuenta otraCuenta = new Cuenta(22, 500.5, "Ana Albizu");
        System.out.println(otraCuenta);

        // Ingresa 500 euros en otraCuenta
        System.out.println("Ingresa: " + otraCuenta.ingresar(500) + " Saldo: " + otraCuenta.getSaldo());

        // Retira 500 euros de otraCuenta
        System.out.println("Retira: " + otraCuenta.retirar(500) + " Saldo: " + otraCuenta.getSaldo());

        // Retira 2000 euros otraCuenta
        System.out.println("Retira: " + otraCuenta.retirar(2000) + " Saldo: " + otraCuenta.getSaldo());

        // Muestra otraCuenta
        System.out.println(otraCuenta);
    }
}

```

**Correcta**

Puntos para este envío: 1,00/1,00.

**Finalizar revisión**

### Contacta con nosotros

- 📍 Dirección: Calle Álava 41, interior - Vitoria-Gasteiz
- 📞 Teléfono: 945 567 953
- ✉️ E-mail: [ulhi@ulhi.net](mailto:ulhi@ulhi.net)
- 🐦 Twitter: [@UrrutikoLH](https://twitter.com/UrrutikoLH)