

Navegación por el cuestionario

1	2	3	4	5	6	7	8	9	10
✓	✓	✓	✓	✓	✓	✓	✓	✓	✗

Finalizar revisión

Comenzado el	martes, 29 de enero de 2019, 18:27
Estado	Finalizado
Finalizado en	miércoles, 30 de enero de 2019, 10:31
Tiempo empleado	16 horas 3 minutos
Calificación	4,83 de 10,00 (48%)

Pregunta 1

Correcta
Puntuía 0,88 sobre 1,00
 Marcar pregunta

Tenemos el siguiente código:

```
Cuenta miCuenta = new Cuenta(1234, 1000.55, "Pedro Aristi");
Cuenta[] listaCuentas = new Cuenta[8];
Cuenta tuCuenta;
listaCuentas[7] = new Cuenta();
listaCuentas[6] = miCuenta;
```

Si la clase Cuenta se define:

```
public class Cuenta {
    private int numero;
    private double saldo;
    private String titular;

    public Cuenta() {
        this(0, 0.0, "");
    }

    public Cuenta(int numero, double saldo, String titular) {
        this.numero = numero;
        this.saldo = saldo;
        this.titular = titular;
    }
    ...
}
```

¿Qué valor tendrán las siguientes expresiones? Recuerda escribir los valores de acuerdo al tipo que les corresponda: 7.0 en vez de 7 para los double, comillas dobles para las cadenas de caracteres...:

Expresión	Valor	Expresión	Valor
miCuenta.numero	1234 ✓	listaCuenta[7].numero	0 ✓
miCuenta.saldo	1000.55 ✓	listaCuenta[7].titular	- ✓
miCuenta.titular	"Pedro Aristi" ✓	listaCuenta[6].saldo	1000.55 ✓
tuCuenta	null ✓	listaCuenta[0]	null ✓

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,88/1,00.

Pregunta 2

Correcta
Puntuía 0,88 sobre 1,00
 Marcar pregunta

Tenemos el siguiente código:

```
Circulo miCirculo = new Circulo();
Circulo[] listaCirculos = new Circulo[5];
Circulo tuCirculo = new Circulo(5.5, "roja");
listaCirculos[0] = tuCirculo;
listaCirculos[1] = new Circulo(miCirculo);
```

Si la clase Circulo se define:

```
public class Circulo {
    private double radio;
    private String color;

    public Circulo() {
        this(0.0, "");
    }

    public Circulo(double radio, String color) {
        this.radio = radio;
        this.color = color;
    }

    public Circulo(Circulo otro) {
        radio = otro.radio;
        color = otro.color;
    }
    ...
}
```

¿Qué valor tendrán las siguientes expresiones? Recuerda escribir los valores de acuerdo al tipo que les corresponda: 7.0 en vez de 7 para los double, comillas dobles para las cadenas de caracteres...:

Expresión	Valor	Expresión	Valor
miCirculo.radio	0.0 ✓	listaCirculos[0].radio	5.5 ✓
miCirculo.color	"" ✓	listaCirculos[1].radio	0.0 ✓
tuCirculo.radio	5.5 ✓	listaCirculos[3]	null ✓
tuCirculo.color	"rojo" ✓	listaCirculos[0].color	"rojo" ✓

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,88/1,00.

Pregunta 3

Correcta
Puntuía 1,00 sobre 1,00
 Marcar pregunta

Hemos creado el objeto miCasa de la clase Casa con los siguientes atributos:

```
private String direccion;
private double superficie;
private int edad;
```

Dentro de la clase Casa se ha declarado el método `toString` así como los métodos modificadores y consultores.

Indica todas las alternativas válidas para mostrar el objeto miCasa por consola.

Seleccione una o más de una:

- a. `System.out.println(miCasa);` ✓
- b. `System.out.println(miCasa.direccion + " - " + miCasa.superficie + " - " + miCasa.edad);`
- c. `System.out.println(miCasa.toString());`

d. System.out.println(miCasa.getDireccion() + " - " + miCasa.getSuperficie() + " - " + miCasa.getEdad()); ✓

Respuesta correcta

La respuesta correcta es: System.out.println(miCasa);, System.out.println(miCasa.getDireccion() + " - " + miCasa.getSuperficie() + " - " + miCasa.getEdad());

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 4

Correcta

Puntúa 1,00 sobre

1,00

▼ Marcar pregunta

Indica la afirmación que describe el paso de objetos como parámetros a un método.

Seleccione una:

- a. Los objetos se pasan siempre por valor. Los cambios que se producen dentro del método NO se reflejan fuera de él.
- b. Los objetos se pasan siempre por referencia. Los cambios que se producen dentro del método NO se reflejan fuera de él.
- c. Los objetos se pasan siempre por referencia. Los cambios que se producen dentro del método se reflejan fuera de él. ✓
- d. Los objetos se pasan siempre por valor. Los cambios que se producen dentro del método se reflejan fuera de él.

Pregunta 5

Correcta

Puntúa 0,33 sobre

1,00

▼ Marcar pregunta

Indica que fichero necesitaríamos para utilizar la clase Casa.

Seleccione una:

- a. El fichero Casa.class ✓
- b. El fichero Casa.java
- c. Tanto el fichero Casa.class como Casa.java

Pregunta 6

Incorrecta

Puntúa 0,00 sobre

1,00

▼ Marcar pregunta

Tenemos la clase Circulo con las siguientes características:

```
// Atributos
private double radio;
private String color;

// Constructores
public Circulo()
public Circulo(double radio, String color)
public Circulo(Circulo otro)

// Otros métodos
public void setRadio(double radio)
public void setColor(String color)
public double getRadio()
public String getColor()
public String toString() // Formatea el objeto de la siguiente manera: Circulo color de radio radio
public double calculaArea()
```

y queremos crear la clase ArrayCirculos.

Corrige el método contarColor para que reciba un color y devuelva el número de Circulos de ese color que tiene un objeto ArrayCirculos.

Por ejemplo, si ejecutamos el siguiente código:

```
ArrayCirculos nuevaLista = new ArrayCirculos(5, "rojo", 3);
System.out.println(nuevaLista);
System.out.println("verde: " + nuevaLista.contarColor("verde"));
System.out.println("rojo: " + lista.contarColor("rojo"));
```

Por consola se mostrará :

```
1: Circulo rojo de radio 5.5
2: Circulo rojo de radio 5.5
3: Circulo rojo de radio 5.5
verde: 0
rojo: 3
```

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

Reiniciar respuesta

```
1 | public class ArrayCirculos {
2 |     // Atributos
3 |     Circulo[] lista;
4 |
5 |     // Constructor
6 |     public ArrayCirculos(double radio, String color, int num) {
7 |         lista = new Circulo[num];
8 |         for (int i = 0; i < lista.length; i++) {
9 |
10 |             // Crear un nuevo objeto de la clase Circulo
11 |             Circulo nuevoCirculo = new Circulo(radio, color);
12 |
13 |             // Lo asigna al elemento actual del array
14 |             lista[i] = nuevoCirculo;
15 |         }
16 |     }
17 |
18 |     // Métodos
19 |     public void setCirculo(int num, double radio, String color) {
20 |         lista[num - 1].setColor(color);
| }
```

Syntax Error(s)

```
_Tester_.java:56: error: method contarColor in class ArrayCirculos cannot be applied to given types;
System.out.println("verde: " + lista.contarColor("verde"));
^
 required: String,Circulo[]
 found: String
 reason: actual and formal argument lists differ in length
 _Tester_.java:57: error: method contarColor in class ArrayCirculos cannot be applied to given types;
System.out.println("blanco: " + lista.contarColor("blanco"));
^
 required: String,Circulo[]
 found: String
 reason: actual and formal argument lists differ in length
 _Tester_.java:58: error: method contarColor in class ArrayCirculos cannot be applied to given types;
System.out.println("negro: " + lista.contarColor("negro"));
^
 required: String,Circulo[]
 found: String
 reason: actual and formal argument lists differ in length
3 errors
```

Question author's solution:

```
public class ArrayCirculos {
    // Atributos
```

```

Circulo[] lista;

// Constructores
public ArrayCirculos(double radio, String color, int num) {
    lista = new Circulo[num];
    for (int i = 0; i < lista.length; i++) {

        // Crear un nuevo objeto de la clase Circulo
        Circulo nuevoCirculo = new Circulo(radio, color);

        // Lo asigna al elemento actual del array
        lista[i] = nuevoCirculo;
    }
}

// Metodos
public void setCirculo(int num, double radio, String color) {
    lista[num - 1].setColor(color);
    lista[num - 1].setRadio(radio);
}

public String toString() {
    String formato = "";
    for (int i = 0; i < lista.length; i++) {
        formato += (i + 1) + ": " + lista[i] + "\n";
    }
    return formato;
}

// Corrige este metodo
public int contarColor(String color) {
    int cont = 0;
    for (int i = 0; i < lista.length; i++) {
        if (lista[i].getColor().equals(color)) {
            cont++;
        }
    }
    return cont;
}
}

```

Incorrecta

Puntos para este envío: 0,00/1,00.

Pregunta 7

Correcta

Puntúa 0,75 sobre
1,00

Marcar
pregunta

Tenemos la clase Circulo con las siguientes características:

```

// Atributos
private double radio;
private String color;

// Constructores
public Circulo()
public Circulo(double radio, String color)
public Circulo(Circulo otro)

// Otros métodos
public void setRadio(double radio)
public void setColor(String color)
public double getRadio()
public String getColor()
public String toString()           // Formatea el objeto de la siguiente manera: Circulo color de radio radio
public double calculaArea()

```

Completa el método modificar para que reciba un array de Circulos y un Scanner para leer por teclado, pida el Circulo que se quiere cambiar así como los nuevos valores y los modifique.

```

public static void modificar(Circulo[] lista, Scanner leerDatos) {
    System.out.print("Círculo a cambiar: ");
    int num = leerDatos.nextInt();
    System.out.print("Nuevo radio: ");
    double radio = leerDatos.nextDouble();
    System.out.print("Nuevo color: ");
    String color = leerDatos.next();
    lista[num - 1].setRadio(radio);
    lista[num - 1].setColor(color);
    System.out.print("El círculo" + num + " se ha cambiado");
    Nada
}

```

Si hacemos una llamada al método veremos algo como:

```

Círculo a cambiar: 2
Nuevo radio: 22
Nuevo color: rojo
El círculo se ha cambiado

```

y se modificarán los valores del elemento con índice 1.

lista[num].setColor(color);	Círculo[]
return lista;	[num]

Respuesta correcta

La respuesta correcta es:

Tenemos la clase Circulo con las siguientes características:

```

// Atributos
private double radio;
private String color;

// Constructores
public Circulo()
public Circulo(double radio, String color)
public Circulo(Circulo otro)

// Otros métodos
public void setRadio(double radio)
public void setColor(String color)
public double getRadio()
public String getColor()
public String toString()           // Formatea el objeto de la siguiente manera: Circulo color de radio radio
public double calculaArea()

```

Completa el método modificar para que reciba un array de Circulos y un Scanner para leer por teclado, pida el Circulo que se quiere cambiar así como los nuevos valores y los modifique.

```

public static void modificar([Círculo[]] lista, Scanner leerDatos) {
    System.out.print("Círculo a cambiar: ");
    [int num = leerDatos.nextInt()];
    System.out.print("Nuevo radio: ");
    [double radio = leerDatos.nextDouble()];
    System.out.print("Nuevo color: ");
    [String color = leerDatos.next()];
    lista[num - 1].setRadio(radio);
    [lista[num - 1].setColor(color)];
    System.out.print("El círculo" + num + " se ha cambiado");
    [Nada]
}

```

Si hacemos una llamada al método veremos algo como:

```
Círculo a cambiar: 2
Nuevo radio: 22
Nuevo color: rojo
El círculo se ha cambiado
```

y se modificarán los valores del elemento con índice 1.

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,75/1,00.

Pregunta 8

Incorrecta

Puntuó 0,00 sobre

1,00

∨ Marcar pregunta

Tenemos la clase ArrayCírculos con las siguientes características:

```
public class ArrayCírculos {
    // Atributos
    private Círculo[] lista;

    // Constructor
    public ArrayCírculos(double radio, String color, int num)

    // Métodos
    public void setCírculo(int num, double radio, String color)
    public String toString()
    public int contarColor(String color)
    public String colorMayor()
}
```

Corrige el programa para que cree un objeto de la clase ArrayCírculos, modifique su valores por otros introducidos por teclado y nos diga cuantos Círculos son de color azul y que color tiene el Círculo mayor de la siguiente manera:

```
Número de Círculos que va a tener: 2
CÍRCULO 1:
Color: azul
Radio: 5,5
CÍRCULO 2:
Color: rojo
Radio: 2,2
1: Círculo azul de radio 5.5
2: Círculo rojo de radio 2.2
Azul: 1
Círculo mayor: azul
```

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

Reiniciar respuesta

```
1 import java.util.*;
2
3 public class Programa {
4     public static void main(String[] args) {
5         Scanner teclado = new Scanner(System.in);
6         System.out.print("Número de Círculos que va a tener: ");
7         int num = teclado.nextInt();
8         ArrayCírculos lista = new ArrayCírculos(0.0, "", num);
9         for (int i = 0; i < num; i++) {
10             System.out.println("CÍRCULO " + (i+1) + ": ");
11             System.out.print("Color: ");
12             String color = teclado.nextLine();
13             System.out.print("Radio: ");
14             double radio = teclado.nextInt();
15         }
16         System.out.print(lista);
17         System.out.println("Azul: " + lista.contarColor("rojo"));
18         System.out.println("Círculo mayor: " + lista.colorMayor());
19     }
20 }
```

Input	Expected	Got
3 rojo 3.0 azul 2.2 blanco 6.6	Número de Círculos que va a tener: CÍRCULO 1: Color: Radio: CÍRCULO 2: Color: Radio: CÍRCULO 3: Color: Radio: 1: Círculo rojo de radio 3.0 2: Círculo azul de radio 2.2 3: Círculo blanco de radio 6.6 Azul: 1 Círculo mayor: blanco	Número de Círculos que va a tener: CÍRCULO 1: Color: Radio: ***Error*** Exception in thread "main" java.util.InputMismatchException at java.util.Scanner.throwFor(Scanner.java:864) at java.util.Scanner.next(Scanner.java:1485) at java.util.Scanner.nextInt(Scanner.java:2117) at java.util.Scanner.nextInt(Scanner.java:2076) at Programa.main(Programa.java:14)

Prueba cancelada debido a un error.

Your code must pass all tests to earn any marks. Try again.

Mostrar diferencias

Question author's solution:

```
import java.util.*;

public class Programa {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.print("Número de Círculos que va a tener: ");
        int num = teclado.nextInt();
        ArrayCírculos lista = new ArrayCírculos(0, "", num);
        for (int i = 1; i <= num; i++) {
            System.out.println("CÍRCULO " + i + ": ");
            System.out.print("Color: ");
            String color = teclado.next();
            System.out.print("Radio: ");
            double radio = teclado.nextDouble();
            lista.setCírculo(i, radio, color);
        }
        System.out.print(lista);
        System.out.println("Azul: " + lista.contarColor("azul"));
        System.out.println("Círculo mayor: " + lista.colorMayor());
    }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

Pregunta 9

Incorrecta

Puntuó 0,00 sobre

1,00

∨ Marcar pregunta

Tenemos la clase Concesionario y queremos añadir el método **cochesMasRecientes**. Se le pasará como parámetro un año y devolverá un array de booleanos cuyos elementos serán true o false en función del año del objeto Coche con el mismo índice en el array. Si el coche es más reciente el valor será true y en caso contrario false.

Por ejemplo, si el concesionario tiene los siguientes coches:

```
1: Opel Astra (2015)
2: Hyundai Ioniq (2018)
3: Renault Megane (2014)
4: Volkswagen Golf (2013)
5: Ford Focus (2016)
```

Y hacemos la siguiente llamada:

```
boolean[] resultado = unConcesionario.cochesMasRecientes(2015);
```

El contenido de resultado será:

```
[false, true, false, false, true]
```

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

Reiniciar respuesta

```
1 | public class Concesionario {  
2 |     // Atributos  
3 |     private String nombre;  
4 |     private Coche[] lista;  
5 |  
6 |     // Constructor  
7 |     public Concesionario(String concesionario, String marca, int num) {  
8 |         nombre = concesionario;  
9 |         lista = new Coche[num];  
10|         for (int i = 0; i < lista.length; i++) {  
11|             lista[i] = new Coche(marca, "", 0);  
12|         }  
13|     }  
14|  
15|     // Metodos  
16|     public void setCoche(int indice, String marca, String modelo, int year) {  
17|         lista[indice].setMarca(marca);  
18|         lista[indice].setModelo(modelo);  
19|         lista[indice].setYear(year);  
20|     }  
21| }
```

Syntax Error(s)

```
_Tester_.java:55: error: cannot find symbol  
boolean[] resultado = unConcesionario.cochesMasRecientes(2015);  
^  
symbol:   method cochesMasRecientes(int)  
location: variable unConcesionario of type Concesionario  
1 error
```

Question author's solution:

```
public class Concesionario {  
    // Atributos  
    private String nombre;  
    private Coche[] lista;  
  
    // Constructor  
    public Concesionario(String concesionario, String marca, int num) {  
        nombre = concesionario;  
        lista = new Coche[num];  
        for (int i = 0; i < lista.length; i++) {  
            lista[i] = new Coche(marca, "", 0);  
        }  
    }  
  
    // Metodos  
    public void setCoche(int indice, String marca, String modelo, int year) {  
        lista[indice].setMarca(marca);  
        lista[indice].setModelo(modelo);  
        lista[indice].setYear(year);  
    }  
  
    public int getNumCoches() {  
        return lista.length;  
    }  
  
    public String toString() {  
        String formato = nombre.toUpperCase() + "\n";  
        for (int i = 0; i < lista.length; i++) {  
            formato += (i + 1) + ":" + lista[i] + "\n";  
        }  
        return formato;  
    }  
  
    public String mostrarCoche(int indice) {  
        return lista[indice].toString();  
    }  
  
    /*  
     * Busca los coches mas reciente al año indicado  
     * Devolverá un array de booleanos que contendrá true si el coche en esa posición es más reciente  
     * false en caso contrario  
     */  
    public boolean[] cochesMasRecientes(int year) {  
        boolean[] masRecientes = new boolean[lista.length];  
        for (int i = 0; i < lista.length; i++) {  
            if (lista[i].getYear() > year) {  
                masRecientes[i] = true;  
            }  
        }  
        return masRecientes;  
    }  
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

Pregunta 10

Incorrecta

Puntúa 0,00 sobre
1,00

▼ Marcar
pregunta

Tenemos la clase Concesionario con las siguientes características:

```
// Constructor  
public Concesionario(String nombreConcesionario, String marca, int numCoches)  
  
// Modifica los datos del coche con el índice indicado en el concesionario. Los índices empezarán en 0.  
public void setCoche(int indice, String marca, String modelo, int year)  
  
// Devuelve el número de coches en el concesionario  
public int getNumCoches()  
  
// Formatea los datos del concesionario de la siguiente manera:  
// NOMBRE DEL CONCESIONARIO  
// 1: Marca Modelo (año)  
// 2: Marca Modelo (año)  
public String toString()  
  
// Muestra los datos del coche en el concesionario con el índice indicado  
public String mostrarCoche(int indice)  
  
// Busca el primer coche que del modelo indicado. Devolverá su índice (empieza en 0) o -1 si no lo encuentra  
public int BuscarModelo(String modelo)  
  
// Busca el coche más viejo del concesionario. Devolverá su índice (empieza en 0) en el concesionario  
public int buscarViejo()
```

Crea el método crearConcesionario que recibirá un objeto de la clase File, conectará con el fichero, leerá el nombre y el número de coches y creará un concesionario. Como marca de los coches utilizará un String vacío. A continuación, modificará las características de los coches del concesionario con los valores del fichero.

El fichero seguirá el formato siguiente:

```
Todo coches  
5  
Opel Astra 2015  
Hyundai Ioniq 2018  
Renault Megane 2014  
Volkswagen Golf 2013  
Ford Focus 2016
```

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

Reiniciar respuesta

```
1 import java.util.*;
2
3 public class ConcesionarioMain {
4     public static void main(String[] args) {
5
6         Scanner leerTeclado = new Scanner(System.in);
7
8         // Leer fichero y crear concesionario
9         System.out.print("Nombre del fichero: ");
10        Concesionario tuConcesionario = crearConcesionario(new File(leerTeclado.next()));
11        System.out.print(tuConcesionario);
12    }
13
14    // CREAR EL METODO
15
16 }
```

Syntax Error(s)

```
ConcesionarioMain.java:10: error: cannot find symbol
    Concesionario tuConcesionario = crearConcesionario(new File(leerTeclado.next()));
                                         ^
      symbol:   class File
      location: class ConcesionarioMain
1 error
```

Question author's solution:

```
import java.util.*;
import java.io.*;

public class ConcesionarioMain {
    public static void main(String[] args) throws FileNotFoundException {
        Scanner leerTeclado = new Scanner(System.in);

        // Leer fichero y crear concesionario
        System.out.print("Nombre del fichero: ");
        Concesionario tuConcesionario = crearConcesionario(new File(leerTeclado.next()));
        System.out.print(tuConcesionario);
    }

    /*
     * Crea y modifica los datos de un concesionario a partir de los valores del fichero
     */
    public static Concesionario crearConcesionario(File fichEntrada) throws FileNotFoundException {

        // Conecta con el fichero
        Scanner leerFichero = new Scanner(fichEntrada);

        // Lee el nombre y el numero de coches y crea el concesionario
        // Como marca utilizara un String vacio
        String nombre = leerFichero.nextLine();
        int num = leerFichero.nextInt();
        Concesionario unConcesionario = new Concesionario(nombre, "", num);

        // Modifica las caracteristicas de los coches con los valores leidos
        for (int i = 0; i < num; i++) {
            String marca = leerFichero.next();
            String modelo = leerFichero.next();
            int year = leerFichero.nextInt();
            unConcesionario.setCoche(i, marca, modelo, year);
        }

        return unConcesionario;
    }
}
```

Incorrecta

Puntos para este envío: 0.00/1.00.

Finalizar revisión

Contacta con nosotros

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