Navegación por el cuestionario

i 1 2 3 4 5 6 7 8 9 10

Finalizar revisión

Comenzado el	lunes, 1 de abril de 2019, 17:28
Estado	Finalizado
Finalizado en	martes, 2 de abril de 2019, 16:08
Tiempo empleado	22 horas 40 minutos
Calificación	**3,17** de 10,00 (**32**%)

Información

Marcar pregunta

En la plataforma no se pueden ejecutar aplicaciones con interfaz gráfica.
No os guiéis por las correcciones porque lo único que hacen es comparar vuestro código con una posible solución.
Desarrollar y ejecutad el programa en jGrasp y comparad la solución con la planteada.
Para cualquier duda ya sabéis donde estoy.

Pregunta **1**

Correcta

Puntúa 1,00 sobre 1,00

Marcar pregunta

Indica los métodos que pertenecen a la interfaz MouseListener:

Seleccione una o más de una:

- [x] a. mouseEntered
- [x] b. mouseePressed
- [ ] c. ActionPerformed
- [ ] d. keyPressed

Respuesta correcta

La respuesta correcta es: mousePressed, mouseEntered

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta **2**

Correcta

Puntúa 0,67 sobre 1,00

Marcar pregunta

Indica las afirmaciones que describen los eventos en Java

Seleccione una o más de una:

- [x] a. Cuando se interactúa con una interfaz gráfica se generan diferentes tipos de eventos
- [x] b. Cuando se produce un evento Java crea un objeto de la clase correspondiente a ese tipo de evento
- [ ] c. Cuando se produce un evento, Java crea un objeto Event. El programador tendrá que analizar la acción que lo ha provocado.
- [ ] d. Cuando se genera un evento se abre una ventana avisando de ello.

Respuesta correcta

La respuesta correcta es: Cuando se interactúa con una interfaz gráfica se generan diferentes tipos de eventos, Cuando se produce un evento Java crea un objeto de la clase correspondiente a ese tipo de evento

Correcta

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

Pregunta **3**

Parcialmente correcta

Puntúa 0,50 sobre 1,00

Marcar pregunta

Una lista desplegable…

Seleccione una o más de una:

- [ ] a. Está disponible en AWT, pero no en Swing.
- [ ] b. Todas son correctas.
- [ ] c. Es una mezcla entre un campo de texto editable y una lista.
- [x] d. Se representa en Java con el componente Swing `JComboBox`.

La respuesta correcta es: Es una mezcla entre un campo de texto editable y una lista., Se representa en Java con el componente Swing `JComboBox`.

Parcialmente correcta

Puntos para este envío: 0,50/1,00.

Pregunta **4**

Correcta

Puntúa 1,00 sobre 1,00

Marcar pregunta

El componente `JScrollPanel` permite que aparezcan barras de desplazamiento.

Seleccione una:

- (•) Verdadero
- ( ) Falso

La respuesta correcta es 'Verdadero'

Correcta

Puntos para este envío: 1,00/1,00.

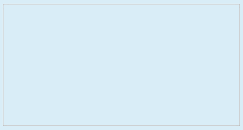
Pregunta **5**

Incorrecta

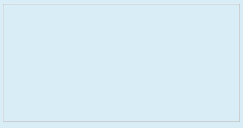
Puntúa 0,00 sobre 1,00

Marcar pregunta

Corrige el código que se plantea para que la ventana que se muestra sea interactiva:



Cuando se el ratón entre en una caja de texto su color cambiará por el indicado. Cuando salga, volverá a ponerse en blanco:



Se usará la interfaz MouseListener. Solo habrá que desarrollar los mouseEntered y mouseExited.

El color de las cajas de texto se cambiarán con el método setBackground.

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

Reiniciar respuesta

```
1  import java.awt.*;          // Componentes de dibujo
2  import javax.swing.*;       // Componentes de una interfaz gráfica
3  import java.util.*;
4  import java.io.*;
5  import java.awt.event.*;
6
7  public class ClaseColores implements MouseListener {
8
9     // Se puede incluir el main en la clase
10    public static void main(String[] args) throws FileNotFoundException {
```

```
11          Scanner leerFichero = new Scanner(new File("colores.txt"));
12
13          ClaseColores gui = new ClaseColores(leerFichero);
14       }
15
16       // Constructor
17       public ClaseColores(Scanner leerDatos) {
18
19          // Crea y configura la ventana principal
20          JFrame ventana = new JFrame();
```

	Expected	Got

✓	`import java.awt.*;  // Componentes de dibujo`
	`import javax.swing.*;  // Componentes de una interfaz gráfica`
	`import java.awt.event.*;  // Eventos`
	`import java.util.*;`
	`import java.io.*;`
	
	`public class ClaseGuiEventos implements MouseListener {`
	
	`  // Se puede incluir el main en la clase`
	`  public static void main(String[] args) throws FileNotFoundException {`
	`  Scanner leerFichero = new Scanner(new File("colores.txt"));`
	
	`  ClaseGuiEventos gui = new ClaseGuiEventos(leerFichero);`
	`  }`
	
	`  // Constructor`
	`  public ClaseGuiEventos(Scanner leerDatos) {`
	
	`  // Crea y configura la ventana principal`
	`  JFrame ventana = new JFrame();`
	`  ventana.setTitle("EJERCICIO");`
	`  ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
	`  ventana.setSize(300, 150);`
	`  ventana.setLocation(600, 200);`
	`  ventana.setLayout(new GridLayout(3, 1));`
	
	`  // Crea y configura los componentes`
	`  while (leerDatos.hasNext()) {`
	`  String texto = leerDatos.next();`
	`  JTextField cajaTexto = new JTextField(texto);`
	`  ventana.add(cajaTexto);`
	
	`  // REGISTRAR UN OBJETO DE LA CLASE OYENTE EN LAS CAJAS DE TEXTO`
	`  cajaTexto.addMouseListener(this);`
	`  }`
	
	`  // Al final. Cuando se haya creado todo se visualiza`
	`  ventana.setVisible(true);`
	`  }`
	
	`  // Desarrolla los métodos de la interfaz MouseListener`
	`  public void mousePressed(MouseEvent evento) {`
	`  }`
	
	`  public void mouseReleased(MouseEvent evento) {`
	
	`  }`
	
	`  // Cuando se entre en una caja de texto, se identificará la caja y se le cambiará el color`
	`  public void mouseEntered(MouseEvent evento) {`
	`  Object objetoMouse = evento.getSource();`
	`  JTextField cajaTextoMouse = (JTextField) objetoMouse;`
	`  String texto = cajaTextoMouse.getText();`
	`  if (texto.equals("ROJO")) {`
	`  cajaTextoMouse.setBackground(Color.RED);`
	`  return;`
	`  }`
	`  if (texto.equals("VERDE")) {`
	`  cajaTextoMouse.setBackground(Color.GREEN);`
	`  return;`
	`  }`
	`  if (texto.equals("AZUL")) {`
	`  cajaTextoMouse.setBackground(Color.BLUE);`
	`  return;`
	`  }`
	
	`  }`
	
	`  // Cuando se salga de una caja de texto, se identificará la caja y se le cambiará el color`
	`  public void mouseExited(MouseEvent evento) {`
	`  Object objetoMouse = evento.getSource();`
	`  JTextField cajaTextoMouse = (JTextField) objetoMouse;`
	`  cajaTextoMouse.setBackground(Color.WHITE);`
	`  }`
	
	`  public void mouseClicked(MouseEvent evento) {`
	`  }`
	`}`
		`import java.awt.*;  // Componentes de dib`
		`import javax.swing.*;  // Componentes de una`
		`import java.util.*;`
		`import java.io.*;`
		`import java.awt.event.*;`
		
		`public class ClaseColores implements MouseListen`
		
		`  // Se puede incluir el main en la clase`
		`  public static void main(String[] args) throws`
		`  Scanner leerFichero = new Scanner(new File`
		
		`  ClaseColores gui = new ClaseColores(leerFi`
		`  }`
		
		`  // Constructor`
		`  public ClaseColores(Scanner leerDatos) {`
		
		`  // Crea y configura la ventana principal`
		`  JFrame ventana = new JFrame();`
		`  ventana.setTitle("EJERCICIO");`
		`  ventana.setDefaultCloseOperation(JFrame.EX`
		`  ventana.setSize(300, 150);`
		`  ventana.setLocation(600, 200);`
		`  ventana.setLayout(new GridLayout(3, 1));`
		
		`  // Crea y configura los componentes`
		`  while (leerDatos.hasNext()) {`
		`  String texto = leerDatos.next();`
		`  JTextField cajaTexto = new JTextField(t`
		`  ventana.add(cajaTexto);`
		
		`  // REGISTRAR UN OBJETO DE LA CLASE OYEN`
		`  cajaTexto.addMouseListener(this);`
		`  }`
		
		`  // Al final. Cuando se haya creado todo se`
		`  ventana.setVisible(true);`
		`  }`
		
		`  // Desarrolla los métodos de la interfaz Mous`
		`  public void mousePressed(MouseEvent evento) {`
		`  }`
		
		`  public void mouseReleased(MouseEvent evento)`
		
		`  }`
		
		`  // Cuando se entre en una caja de texto, se i`
		`  public void mouseEntered(MouseEvent evento) {`
		`  Object objetoMouse = evento.getSource();`
		`  JTextField cajaTextoMouse = (JTextField) o`
		`  String texto = cajaTextoMouse.getText();`
		`  if (texto.equals("ROJO")) {`
		`  cajaTextoMouse.setBackground(Color.RED)`
		`  return;`
		`  }`
		`  if (texto.equals("VERDE")) {`
		`  cajaTextoMouse.setBackground(Color.GREE`
		`  return;`
		`  }`
		`  if (texto.equals("AZUL")) {`
		`  cajaTextoMouse.setBackground(Color.BLUE`
		`  return;`
		`  }`
		
		`  }`
		
		`  // Cuando se salga de una caja de texto, se i`
		`  public void mouseExited(MouseEvent evento) {`
		`  Object objetoMouse = evento.getSource();`
		`  JTextField cajaTextoMouse = (JTextField) o`
		`  cajaTextoMouse.setBackground(Color.WHITE);`
		`  }`
		
		`  public void mouseClicked(MouseEvent evento) {`
		`  }`
		`}`

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

Question author's solution:

```
import java.awt.*;         // Componentes de dibujo
import javax.swing.*;      // Componentes de una interfaz gráfica
import java.awt.event.*;   // Eventos
import java.util.*;
import java.io.*;

public class ClaseGuiEventos implements MouseListener {

   // Se puede incluir el main en la clase
   public static void main(String[] args) throws FileNotFoundException {
      Scanner leerFichero = new Scanner(new File("colores.txt"));

      ClaseGuiEventos gui = new ClaseGuiEventos(leerFichero);
   }

   // Constructor
   public ClaseGuiEventos(Scanner leerDatos) {

      // Crea y configura la ventana principal
      JFrame ventana = new JFrame();
      ventana.setTitle("EJERCICIO");
      ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      ventana.setSize(300, 150);
      ventana.setLocation(600, 200);
      ventana.setLayout(new GridLayout(3, 1));

      // Crea y configura los componentes
      while (leerDatos.hasNext()) {
         String texto = leerDatos.next();
         JTextField cajaTexto = new JTextField(texto);
         ventana.add(cajaTexto);

         // REGISTRAR UN OBJETO DE LA CLASE OYENTE EN LAS CAJAS DE TEXTO
         cajaTexto.addMouseListener(this);
      }

      // Al final. Cuando se haya creado todo se visualiza
```

```
      ventana.setVisible(true);
   }

   // Desarrolla los métodos de la interfaz MouseListener
   public void mousePressed(MouseEvent evento) {
   }

   public void mouseReleased(MouseEvent evento) {

   }

   // Cuando se entre en una caja de texto, se identificará la caja y se le cambiará el color
   public void mouseEntered(MouseEvent evento) {
      Object objetoMouse = evento.getSource();
      JTextField cajaTextoMouse = (JTextField) objetoMouse;
      String texto = cajaTextoMouse.getText();
      if (texto.equals("ROJO")) {
         cajaTextoMouse.setBackground(Color.RED);
         return;
      }
      if (texto.equals("VERDE")) {
         cajaTextoMouse.setBackground(Color.GREEN);
         return;
      }
      if (texto.equals("AZUL")) {
         cajaTextoMouse.setBackground(Color.BLUE);
         return;
      }

   }

   // Cuando se salga de una caja de texto, se identificará la caja y se le cambiará el color
   public void mouseExited(MouseEvent evento) {
      Object objetoMouse = evento.getSource();
      JTextField cajaTextoMouse = (JTextField) objetoMouse;
      cajaTextoMouse.setBackground(Color.WHITE);
   }

   public void mouseClicked(MouseEvent evento) {
   }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

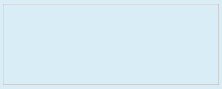
Pregunta 6

Incorrecta

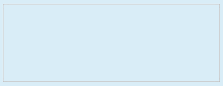
Puntúa 0,00 sobre 1,00

Marcar pregunta

Corrige el código que se plantea para que la ventana que se muestra sea interactiva:



Una vez introducido un texto, cuando se pulse el botón, se obtendrá el número de letras y se mostrará de la siguiente manera:



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

Reiniciar respuesta

```
1  import java.awt.*;            // Componentes de dibujo
2  import javax.swing.*;         // Componentes de una interfaz gráfica
3  import java.awt.event.*;
4
5  public class ClaseGuiEventos06 implements ActionListener {
6
7     // Se puede incluir el main en la clase
8     public static void main(String[] args) {
9        ClaseGuiEventos06 gui = new ClaseGuiEventos06();
10    }
11
12    // Atributos
13    private JFrame ventana;
14    private JButton boton;
15    private JLabel resultado;
16    private JTextField texto;
17
18    // Constructor
19    public ClaseGuiEventos06() {
20
```

	Expected	Got

✗	`import java.awt.*;  // Componentes de dibujo`
	`import javax.swing.*;  // Componentes de una interfaz gráfica`
	`import java.awt.event.*;  // Eventos`
	
	`public class ClaseGuiEventos implements ActionListener {`
	
	`  // Se puede incluir el main en la clase`
	`  public static void main(String[] args) {`
	`  ClaseGuiEventos gui = new ClaseGuiEventos();`
	`  }`
	
	`  // Atributos`
	`  private JFrame ventana;`
	`  private JButton boton;`
	`  private JLabel resultado;`
	`  private JTextField texto;`
	
	`  // Constructor`
	`  public ClaseGuiEventos() {`
	
	`  // Crea y configura la ventana principal`
	`  ventana = new JFrame();`
	`  ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
	`  ventana.setLocation(500, 500);`
	`  ventana.setTitle("EJERCICIO 2");`
	`  ventana.setLayout(new BorderLayout());`
	
	`  // Crea y configura los componentes`
	`  ventana.add(new JLabel("Escribe un texto"), BorderLayout.WEST);`
	`  texto = new JTextField(15);`
	`  ventana.add(texto, BorderLayout.CENTER);`
	
	`  boton = new JButton();`
	`  boton.setText("CONTAR");`
	`  ventana.add(boton, BorderLayout.NORTH);`
	
	`  // REGISTRAR UN OBJETO DE LA CLASE OYENTE EN EL BOTON`
	`  boton.addActionListener(this);`
	
	`  resultado = new JLabel("Número de letras del texto");`
	`  ventana.add(resultado, BorderLayout.SOUTH);`
	
	`  // Al final. Cuando se haya creado todo se visualiza`
	`  ventana.pack();`
	`  ventana.setVisible(true);`
	`  }`
	
	`  // Implementamos la interfaz ActionListener. Sobreescribimos el método actionPerformed`
		`***Error***`
		`Traceback (most recent call last):`
		`  File "prog.python3", line 123, in <module>`
		`  comment = "Expected {1} lines, got {2}".format(1`
		`IndexError: tuple index out of range`

Expected	Got

// Leemos el texto introducido en la caja de texto, obtenemos el número de letras
// y lo mostramos en la etiqueta resultado
@Override
public void actionPerformed(ActionEvent event) {
String textoIntroducido = texto.getText();
int numeroLetras = textoIntroducido.length();
resultado.setText("Número de letras: " + numeroLetras);
}
}	

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

Question author's solution:

```
import java.awt.*;            // Componentes de dibujo
import javax.swing.*;         // Componentes de una interfaz gráfica
import java.awt.event.*;      // Eventos

public class ClaseGuiEventos implements ActionListener {

   // Se puede incluir el main en la clase
   public static void main(String[] args) {
      ClaseGuiEventos gui = new ClaseGuiEventos();
   }

   // Atributos
   private JFrame ventana;
   private JButton boton;
   private JLabel resultado;
   private JTextField texto;

   // Constructor
   public ClaseGuiEventos() {

      // Crea y configura la ventana principal
      ventana = new JFrame();
      ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      ventana.setLocation(500, 500);
      ventana.setTitle("EJERCICIO 2");
      ventana.setLayout(new BorderLayout());

      // Crea y configura los componentes
      ventana.add(new JLabel("Escribe un texto"), BorderLayout.WEST);
      texto = new JTextField(15);
      ventana.add(texto, BorderLayout.CENTER);

      boton = new JButton();
      boton.setText("CONTAR");
      ventana.add(boton, BorderLayout.NORTH);

      // REGISTRAR UN OBJETO DE LA CLASE OYENTE EN EL BOTON
      boton.addActionListener(this);

      resultado = new JLabel("Número de letras del texto");
      ventana.add(resultado, BorderLayout.SOUTH);

      // Al final. Cuando se haya creado todo se visualiza
      ventana.pack();
      ventana.setVisible(true);
   }

   // Implementamos la interfaz ActionListener. Sobreescribimos el método actionPerformed
   // Leemos el texto introducido en la caja de texto, obtenemos el número de letras
   // y lo mostramos en la etiqueta resultado
   @Override
   public void actionPerformed(ActionEvent event) {
      String textoIntroducido = texto.getText();
      int numeroLetras = textoIntroducido.length();
      resultado.setText("Número de letras: " + numeroLetras);
   }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

Pregunta **7**

Incorrecta

Puntúa 0,00 sobre 1,00

Marcar pregunta

Corrige el código que se plantea para que la ventana que se muestra sea interactiva:

Cuando se pulse el botón "MAYUSCULAS" el texto pasará a muyúsculas y cuando se pulse "MINUSCULAS" a minúsculas:

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

Reiniciar respuesta

```
1  import java.awt.*;          // Componentes de dibujo
2  import javax.swing.*;       // Componentes de una interfaz gráfica
3  import java.awt.event.*;
4
5  public class ClaseGuiEventos07 implements ActionListener {
6
7     // Se puede incluir el main en la clase
8     public static void main(String[] args) {
9
10       ClaseGuiEventos07 gui = new ClaseGuiEventos07();
11    }
12
13    // Constantes
14    String OPCION1 = "MAYUSCULAS";
15    String OPCION2 = "MINUSCULAS";
16
17    // Atributos
18    JTextArea cajaTexto;
19
20    // Constructor
```

Expected	Got

import java.awt.*; // Componentes de dibujo
import javax.swing.*; // Componentes de una interfaz gráfica
import java.awt.event.*; // Eventos
	Error
	Traceback (most recent call last):
	File "prog.python3", line 157, in
	comment = "Expected {1} lines, g

	Expected	Got

	public class ClaseGuiEventos implements ActionListener {
	
	  // Se puede incluir el main en la clase
	  public static void main(String[] args) {
	
	  ClaseGuiEventos gui = new ClaseGuiEventos();
	  }
	
	  // Constantes
	  String OPCION1 = "MAYUSCULAS";
	  String OPCION2 = "MINUSCULAS";
	
	  // Atributos
	  JTextArea cajaTexto;
	
	  // Constructor
	  public ClaseGuiEventos() {
	
	  // Crea y configura la ventana principal
	  JFrame ventana = new JFrame();
	  ventana.setTitle("EJERCICIO");
	  ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	  ventana.setLayout(new FlowLayout());
	
	  // Crea y configura los componentes
	  JLabel etiqueta = new JLabel("Escribe lo que quieras...");
	  ventana.add(etiqueta, BorderLayout.NORTH);
	
	  cajaTexto = new JTextArea(10, 50);
	  ventana.add(cajaTexto, BorderLayout.CENTER);
	
	  JButton boton1 = new JButton(OPCION1);
	  ventana.add(boton1);
	
	  // SE REGISTRA UN OBJETO DE LA CLASE OYENTE EN EL BOTON 1
	  boton1.addActionListener(this);
	
	  JButton boton2 = new JButton(OPCION2);
	  ventana.add(boton2);
	
	  // SE REGISTRA UN OBJETO DE LA CLASE OYENTE EN EL BOTON 2
	  Oyente pulsarBoton2 = new Oyente();
	  boton2.addActionListener(pulsarBoton2);
	
	  // Al final. Cuando se haya creado todo se visualiza
	  ventana.pack();
	  ventana.setVisible(true);
	  }
	
	  // Implementamos la interfaz ActionListener para el botón 1. Sobreescribimos el método actionPerformed
	  // Cuando se pulse el boton se pasará el texto escrito a mayúsculas
	  @Override
	  public void actionPerformed(ActionEvent evento) {
	  String texto = cajaTexto.getText();
	  texto = texto.toUpperCase();
	  cajaTexto.setText(texto);
	  }
	
	  // Crear la clase Oyente para el botón 2
	  // Se crea una clase que tiene el comportamiento del interfaz ActionListener
	  // Cuando se pulse el boton se pasará el texto escrito a minúsculas
	  private class Oyente implements ActionListener {
	  @Override
	  public void actionPerformed(ActionEvent event) {
	  String texto = cajaTexto.getText();
	  texto = texto.toLowerCase();
	  cajaTexto.setText(texto);
	  }
	  }
	}
		IndexError: tuple index out of range

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

Question author's solution:

```
import java.awt.*;         // Componentes de dibujo
import javax.swing.*;      // Componentes de una interfaz gráfica
import java.awt.event.*;   // Eventos

public class ClaseGuiEventos implements ActionListener {

   // Se puede incluir el main en la clase
   public static void main(String[] args) {

      ClaseGuiEventos gui = new ClaseGuiEventos();
   }

   // Constantes
   String OPCION1 = "MAYUSCULAS";
   String OPCION2 = "MINUSCULAS";

   // Atributos
   JTextArea cajaTexto;

   // Constructor
   public ClaseGuiEventos() {

      // Crea y configura la ventana principal
      JFrame ventana = new JFrame();
      ventana.setTitle("EJERCICIO");
      ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      ventana.setLayout(new FlowLayout());

      // Crea y configura los componentes
      JLabel etiqueta = new JLabel("Escribe lo que quieras...");
      ventana.add(etiqueta, BorderLayout.NORTH);

      cajaTexto = new JTextArea(10, 50);
      ventana.add(cajaTexto, BorderLayout.CENTER);

      JButton boton1 = new JButton(OPCION1);
      ventana.add(boton1);

      // SE REGISTRA UN OBJETO DE LA CLASE OYENTE EN EL BOTON 1
      boton1.addActionListener(this);

      JButton boton2 = new JButton(OPCION2);
      ventana.add(boton2);

      // SE REGISTRA UN OBJETO DE LA CLASE OYENTE EN EL BOTON 2
      Oyente pulsarBoton2 = new Oyente();
      boton2.addActionListener(pulsarBoton2);

      // Al final. Cuando se haya creado todo se visualiza
      ventana.pack();
      ventana.setVisible(true);
   }

   // Implementamos la interfaz ActionListener para el botón 1. Sobreescribimos el método actionPerformed
   // Cuando se pulse el boton se pasará el texto escrito a mayúsculas
   @Override
   public void actionPerformed(ActionEvent evento) {
      String texto = cajaTexto.getText();
```

```
    texto = texto.toUpperCase();
    cajaTexto.setText(texto);
  }

  // Crear la clase Oyente para el botón 2
  // Se crea una clase que tiene el comportamiento del interfaz ActionListener
  // Cuando se pulse el boton se pasará el texto escrito a minúsculas
  private class Oyente implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent event) {
      String texto = cajaTexto.getText();
      texto = texto.toLowerCase();
      cajaTexto.setText(texto);
    }
  }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

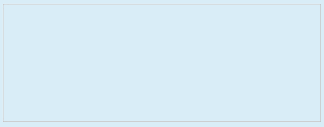
Pregunta **8**

Incorrecta

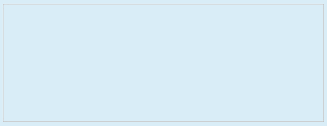
Puntúa 0,00 sobre 1,00

Marcar pregunta

Corrige el código que se plantea para que la ventana que se muestra sea interactiva:



Cuando se pulse el botón "BUSCAR" identificará el color elegido y buscará el primer circulo de ese color en un array. Mostrará el radio de ese circulo. En este caso, será 6.00:



Se usará la clase Circulo y al constructor de la interfaz gráfica se le pasará el array de Circulos donde realizar la busqueda.

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

Reiniciar respuesta

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class CirculoGUI08 implements ActionListener {

  // Programa principal
  public static void main(String[] args) {

    // Creamos un array con 4 Circulos cada uno de un color
    Circulo[] circulos = new Circulo[4];
    for (int i = 0; i < 4; i++) {
      Circulo otroCirculo = new Circulo((i + 1) * 2, colores[i]);
      circulos[i] = otroCirculo;
    }

    CirculoGUI08 gui = new CirculoGUI08(circulos);
  }

  // Atributos
```

	Expected	Got

✗	`// Importamos las librerias necesarias`
	`import java.awt.*;`
	`import javax.swing.*;`
	`import java.awt.event.*;`
	
	`public class CirculoGUI implements ActionListener {`
	
	`  // Programa principal`
	`  public static void main(String[] args) {`
	
	`    // Creamos un array con 4 Circulos cada uno de un color`
	`    Circulo[] circulos = new Circulo[4];`
	`    for (int i = 0; i < 4; i++) {`
	`      Circulo otroCirculo = new Circulo((i + 1) * 2, colores[i]);`
	`      circulos[i] = otroCirculo;`
	`    }`
	
	`    CirculoGUI gui = new CirculoGUI(circulos);`
	`  }`
	
	`  // Atributos`
	`  public static String[] colores = {"Blanco", "Negro", "azul", "Verde"};`
	`  private JLabel radio;`
	`  private JComboBox<String> color;`
	`  private Circulo[] circulos;`
	
	`  // Constructor`
	`  public CirculoGUI(Circulo[] arrayCirculos) {`
	
	`    // Guardamos el array de circulos en el atributo circulos para poder trabajar con él en la clase`
	`    circulos = arrayCirculos;`
	
	`    // Crea y configura la ventana principal`
	`    JFrame ventana = new JFrame();`
	`    ventana.setTitle("ELEGIR CIRCULO");`
	`    ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
	`    ventana.setSize(400, 150);`
	`    ventana.setLayout(new BorderLayout());`
	
	`    // Crea y configura las etiquetas y el JComboBox para el color.`
	`    JPanel arriba = new JPanel(new FlowLayout());`
	`    radio = new JLabel("Radio: ");`
	`    arriba.add(radio);`
	`    arriba.add(new JLabel("Color: "));`
	`    color = new JComboBox<String>(colores);`
	`    arriba.add(color);`
	`    ventana.add(arriba, BorderLayout.CENTER);`
	
	`    // Crea y configura los botones`
	`    JButton boton = new JButton("BUSCAR");`
	`    ventana.add(boton, BorderLayout.SOUTH);`
	
	`    // Registrar el oyente en el botón`
	`    boton.addActionListener(this);`
	
	`    // Al final. Cuando se haya creado todo se visualiza`
	`    ventana.setVisible(true);`
	`  }`
	
	`  // Implementamos la interfaz ActionListener para el botón. Sobreescribimos el método actionPerformed`
	`  // Cuando se pulse el boton buscará el primer círculo de ese color y mostrará su radio`
	`  @Override`
	`  public void actionPerformed(ActionEvent evento) {`
	
	`    String colorElegido = (String) color.getSelectedItem();`
		`***Error***`
		`Traceback (most recent call last):`
		`  File "prog.python3", line 161, in <m`
		`    comment = "Expected {1} lines, got`
		`IndexError: tuple index out of range`

	Expected	Got

	for (Circulo unCirculo : circulos) {
	String unColor = unCirculo.getColor();
	if (colorElegido.equals(unColor)) {
	double unRadio = unCirculo.getRadio();
	radio.setText(String.format("Radio: %.2f", unRadio));
	return;
	}
	}
	}
	}	

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

Question author's solution:

```
// Importamos las librerias necesarias
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class CirculoGUI implements ActionListener {

   // Programa principal
   public static void main(String[] args) {

      // Creamos un array con 4 Circulos cada uno de un color
      Circulo[] circulos = new Circulo[4];
      for (int i = 0; i < 4; i++) {
         Circulo otroCirculo = new Circulo((i + 1) * 2, colores[i]);
         circulos[i] = otroCirculo;
      }

      CirculoGUI gui = new CirculoGUI(circulos);
   }

   // Atributos
   public static String[] colores = {"Blanco", "Negro", "azul", "Verde"};
   private JLabel radio;
   private JComboBox<String> color;
   private Circulo[] circulos;

   // Constructor
   public CirculoGUI(Circulo[] arrayCirculos) {

      // Guardamos el array de circulos en el atributo circulos para poder trabajar con él en la clase
      circulos = arrayCirculos;

      // Crea y configura la ventana principal
      JFrame ventana = new JFrame();
      ventana.setTitle("ELEGIR CIRCULO");
      ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      ventana.setSize(400, 150);
      ventana.setLayout(new BorderLayout());

      // Crea y configura las etiquetas y el JComboBox para el color.
      JPanel arriba = new JPanel(new FlowLayout());
      radio = new JLabel("Radio: ");
      arriba.add(radio);
      arriba.add(new JLabel("Color: "));
      color = new JComboBox<String>(colores);
      arriba.add(color);
      ventana.add(arriba, BorderLayout.CENTER);

      // Crea y configura los botones
      JButton boton = new JButton("BUSCAR");
      ventana.add(boton, BorderLayout.SOUTH);

      // Registrar el oyente en el botón
      boton.addActionListener(this);

      // Al final. Cuando se haya creado todo se visualiza
      ventana.setVisible(true);
   }


   // Implementamos la interfaz ActionListener para el botón. Sobreescribimos el método actionPerformed
   // Cuando se pulse el boton buscará el primer círculo de ese color y mostrará su radio
   @Override
   public void actionPerformed(ActionEvent evento) {

      String colorElegido = (String) color.getSelectedItem();

      for (Circulo unCirculo : circulos) {
         String unColor = unCirculo.getColor();
         if (colorElegido.equals(unColor)) {
            double unRadio = unCirculo.getRadio();
            radio.setText(String.format("Radio: %.2f", unRadio));
            return;
         }
      }
   }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

Pregunta **9**

Incorrecta

Puntúa 0,00 sobre 1,00

Marcar pregunta

Añade el código necesario para que la ventana que se muestra sea interactiva:

Cuando se pulse el botón, se recogerán los datos introducidos y se creará un objeto de la clase Coche con esos datos:

Finalmente, se mostrará el contenido del ArrayList para comprobar que se ha añadido correctamente:

Recuerda que los ArrayList disponen del método toString().

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

Reiniciar respuesta

```
1  import java.util.*;
2  import java.awt.*;
3  import javax.swing.*;
4  import java.awt.event.*;
5
```

```java
public class CocheGuiEventos {

   // Programa principal
   public static void main(String[] args) {

      // Crea el ArrayList de datos sobre los que vamos a trabajar
      ArrayList<Coche> listaCoches = new ArrayList<Coche>();
      listaCoches.add(new Coche("Opel", "Meriva", 2013));
      listaCoches.add(new Coche("Honda", "Civic", 2016));
      listaCoches.add(new Coche("Renault", "Clio", 2014));
      listaCoches.add(new Coche("Ford", "Fiesta", 2012));

      // Creamos la interfaz gráfica y le pasamos los datos
      CocheGuiEventos gui = new CocheGuiEventos(listaCoches);
   }
```

Expected	Got

`// Importamos las librerias necesarias`
`import java.util.*;`
`import java.awt.*;`
`import javax.swing.*;`
`import java.awt.event.*;`
 	
`public class CocheGuiEventos {`
 	
`   // Programa principal`
`   public static void main(String[] args) {`
 	
`      // Crea el ArrayList de datos sobre los que vamos a trabajar`
`      ArrayList<Coche> listaCoches = new ArrayList<Coche>();`
`      listaCoches.add(new Coche("Opel", "Meriva", 2013));`
`      listaCoches.add(new Coche("Honda", "Civic", 2016));`
`      listaCoches.add(new Coche("Renault", "Clio", 2014));`
`      listaCoches.add(new Coche("Ford", "Fiesta", 2012));`
 	
`      // Creamos la interfaz gráfica y le pasamos los datos`
`      CocheGuiEventos gui = new CocheGuiEventos(listaCoches);`
`   }`
 	
`   // Atributos`
`   JTextField cajaMarca;`
`   JTextField cajaModelo;`
`   JTextField cajaYear;`
`   ArrayList<Coche> coches;         // Datos con los que vamos a trabajar`
 	
`   // Constructor`
`   public CocheGuiEventos(ArrayList<Coche> coches) {`
 	
`      // Guardamos el ArrayList de Coches recibido en el atributo coches`
`      this.coches = coches;`
 	
`      // Crea y configura la ventana principal`
`      JFrame ventana = new JFrame();`
`      ventana.setTitle("AÑADIR COCHE");`
`      ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
`      ventana.setSize(300, 150);`
`      ventana.setLayout(new BorderLayout());`
 	
`      // Crea y configura la etiqueta de la zona norte`
`      JLabel titulo = new JLabel("Vas añadir un nuevo coche");`
`      ventana.add(titulo, BorderLayout.NORTH);`
 	
`      // Crea y configura la zona centro.`
`      // Crea las etiquetas y las cajas de texto y las añade al marco secundario`
`      // El marco secundario lo añade a la zona centro`
`      JPanel centro = new JPanel(new GridLayout(3, 2));`
`      centro.add(new JLabel("Marca: "));`
`      cajaMarca = new JTextField(10);`
`      centro.add(cajaMarca);`
`      centro.add(new JLabel("Modelo: "));`
`      cajaModelo = new JTextField(10);`
`      centro.add(cajaModelo);`
`      centro.add(new JLabel("Año: "));`
`      cajaYear = new JTextField(10);`
`      centro.add(cajaYear);`
`      ventana.add(centro, BorderLayout.CENTER);`
 	
`      // Crea y configura el botón de la zona sur`
`      JButton botonCrear = new JButton("CREAR");`
`      ventana.add(botonCrear, BorderLayout.SOUTH);`
 	
`      // CREAR OYENTES Y REGISTRARLOS`
`      OyenteCrear pulsarCrear = new OyenteCrear();`
`      botonCrear.addActionListener(pulsarCrear);`
 	
`      // Al final. Cuando se haya creado todo se visualiza`
`      ventana.setVisible(true);`
`   }`
 	
`   // CREAR OYENTE ActionListener PARA EL BOTON CREAR`
`   private class OyenteCrear implements ActionListener {`
 	
`      @Override`
`      public void actionPerformed(ActionEvent evento) {`
`         String marca = cajaMarca.getText();`
`         String modelo = cajaModelo.getText();`
`         String yearTexto = cajaYear.getText();`
`         int year = Integer.parseInt(yearTexto);`
`         coches.add(new Coche(marca, modelo, year));`
`         JOptionPane.showMessageDialog(null, coches.toString());`
`      }`
`   }`
`}`
	`***Error***`
	`Traceback (most recent call last):`
	`  File "prog.python3", line 182, in <module>`
	`    comment = "Expected {1} lines, got {2}".format(lines, len`
	`IndexError: tuple index out of range`

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

Question author's solution:

```java
// Importamos las librerias necesarias
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class CocheGuiEventos {

   // Programa principal
   public static void main(String[] args) {

      // Crea el ArrayList de datos sobre los que vamos a trabajar
      ArrayList<Coche> listaCoches = new ArrayList<Coche>();
      listaCoches.add(new Coche("Opel", "Meriva", 2013));
      listaCoches.add(new Coche("Honda", "Civic", 2016));
      listaCoches.add(new Coche("Renault", "Clio", 2014));
      listaCoches.add(new Coche("Ford", "Fiesta", 2012));

      // Creamos la interfaz gráfica y le pasamos los datos
      CocheGuiEventos gui = new CocheGuiEventos(listaCoches);
   }

   // Atributos
   JTextField cajaMarca;
   JTextField cajaModelo;
```

```
   JTextField cajaYear;
   ArrayList<Coche> coches;        // Datos con los que vamos a trabajar

   // Constructor
   public CocheGuiEventos(ArrayList<Coche> coches) {

      // Guardamos el ArrayList de Coches recibido en el atributo coches
      this.coches = coches;

      // Crea y configura la ventana principal
      JFrame ventana = new JFrame();
      ventana.setTitle("AÑADIR COCHE");
      ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      ventana.setSize(300, 150);
      ventana.setLayout(new BorderLayout());

      // Crea y configura la etiqueta de la zona norte
      JLabel titulo = new JLabel("Vas añadir un nuevo coche");
      ventana.add(titulo, BorderLayout.NORTH);

      // Crea y configura la zona centro.
      // Crea las etiquetas y las cajas de texto y las añade al marco secundario
      // El marco secundario lo añade a la zona centro
      JPanel centro = new JPanel(new GridLayout(3, 2));
      centro.add(new JLabel("Marca: "));
      cajaMarca = new JTextField(10);
      centro.add(cajaMarca);
      centro.add(new JLabel("Modelo: "));
      cajaModelo = new JTextField(10);
      centro.add(cajaModelo);
      centro.add(new JLabel("Año: "));
      cajaYear = new JTextField(10);
      centro.add(cajaYear);
      ventana.add(centro, BorderLayout.CENTER);

      // Crea y configura el botón de la zona sur
      JButton botonCrear = new JButton("CREAR");
      ventana.add(botonCrear, BorderLayout.SOUTH);

      // CREAR OYENTES Y REGISTRARLOS
      OyenteCrear pulsarCrear = new OyenteCrear();
      botonCrear.addActionListener(pulsarCrear);

      // Al final. Cuando se haya creado todo se visualiza
      ventana.setVisible(true);
   }

   // CREAR OYENTE ActionListener PARA EL BOTON CREAR
   private class OyenteCrear implements ActionListener {

      @Override
      public void actionPerformed(ActionEvent evento) {
         String marca = cajaMarca.getText();
         String modelo = cajaModelo.getText();
         String yearTexto = cajaYear.getText();
         int year = Integer.parseInt(yearTexto);
         coches.add(new Coche(marca, modelo, year));
         JOptionPane.showMessageDialog(null, coches.toString());
      }
   }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

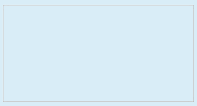
Pregunta **10**

Incorrecta

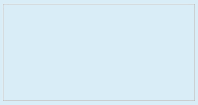
Puntúa 0,00 sobre 1,00

Marcar pregunta

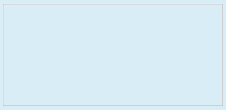
Añade el código necesario para que la ventana que se muestra sea interactiva:



Cuando se pulse el botón, se recogerán los datos introducidos, se creará un objeto de la clase Coche con esos datos y el año 2010 y se buscarán todos los coches que sean de la misma marca y modelo utilizando el método equals de la clase Coche:



Finalmente, se mostrará el resultado obtenido de la siguiente manera:



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

Reiniciar respuesta

```
1  // Importamos las librerias necesarias
2  import java.util.*;
3  import java.awt.*;
4  import javax.swing.*;
5  import java.awt.event.*;
6
7  public class CocheGuiEventos10 {
8
9     // Programa principal
10    public static void main(String[] args) {
11
12       // Crea el ArrayList de datos sobre los que vamos a trabajar
13       ArrayList<Coche> listaCoches = new ArrayList<Coche>();
14       listaCoches.add(new Coche("Opel", "Meriva", 2013));
15       listaCoches.add(new Coche("Ford", "Fiesta", 2015));
16       listaCoches.add(new Coche("Honda", "Civic", 2016));
17       listaCoches.add(new Coche("Renault", "Clio", 2014));
18       listaCoches.add(new Coche("Ford", "Fiesta", 2012));
19
20       // Creamos la interfaz gráfica y le pasamos los datos
```

	Expected	Got

✗	`// Importamos las librerias necesarias`
	`import java.util.*;`
	`import java.awt.*;`
	`import javax.swing.*;`
	`import java.awt.event.*;`
	
	`public class CocheGuiEventos {`
	
	`  // Programa principal`
	`  public static void main(String[] args) {`
	
	`  // Crea el ArrayList de datos sobre los que vamos a trabajar`
	`  ArrayList<Coche> listaCoches = new ArrayList<Coche>();`
	`  listaCoches.add(new Coche("Opel", "Meriva", 2013));`
	`  listaCoches.add(new Coche("Ford", "Fiesta", 2015));`
	`  listaCoches.add(new Coche("Honda", "Civic", 2016));`
	`  listaCoches.add(new Coche("Renault", "Clio", 2014));`
	`  listaCoches.add(new Coche("Ford", "Fiesta", 2012));`
		`***Error***`
		`Traceback (most recent call last):`
		`  File "prog.python3", line 181, in <module>`
		`  comment = "Expected {1} lines, got {2}".format(lines, len`
		`IndexError: tuple index out of range`

	Expected	Got

	`        // Creamos la interfaz gráfica y le pasamos los datos`
	`        CocheGuiEventos gui = new CocheGuiEventos(listaCoches);`
	`    }`
	
	`    // Atributos`
	`    JTextField cajaMarca;`
	`    JTextField cajaModelo;`
	`    ArrayList<Coche> coches;         // Datos con los que vamos a trabajar`
	
	`    // Constructor`
	`    public CocheGuiEventos(ArrayList<Coche> coches) {`
	
	`        // Guardamos el ArrayList de Coches recibido en el atributo coches`
	`        this.coches = coches;`
	
	`        // Crea y configura la ventana principal`
	`        JFrame ventana = new JFrame();`
	`        ventana.setTitle("AÑADIR COCHE");`
	`        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
	`        ventana.setLayout(new BorderLayout());`
	
	`        // Crea y configura la etiqueta de la zona norte`
	`        JLabel titulo = new JLabel("Vas a buscar un coche");`
	`        ventana.add(titulo, BorderLayout.NORTH);`
	
	`        // Crea y configura la zona centro.`
	`        // Crea las etiquetas y las cajas de texto y las añade al marco secundario`
	`        // El marco secundario lo añade a la zona centro`
	`        JPanel centro = new JPanel(new GridLayout(2, 2));`
	`        centro.add(new JLabel("Marca: "));`
	`        cajaMarca = new JTextField(10);`
	`        centro.add(cajaMarca);`
	`        centro.add(new JLabel("Modelo: "));`
	`        cajaModelo = new JTextField(10);`
	`        centro.add(cajaModelo);`
	`        ventana.add(centro, BorderLayout.CENTER);`
	
	`        // Crea y configura el botón de la zona sur`
	`        JButton botonBuscar = new JButton("BUSCAR");`
	`        ventana.add(botonBuscar, BorderLayout.SOUTH);`
	
	`        // CREAR OYENTES Y REGISTRARLOS`
	`        OyenteBuscar pulsarBuscar = new OyenteBuscar();`
	`        botonBuscar.addActionListener(pulsarBuscar);`
	
	`        // Al final. Cuando se haya creado todo se visualiza`
	`        ventana.pack();`
	`        ventana.setVisible(true);`
	`    }`
	
	`    // CREAR OYENTE ActionListener PARA EL BOTON CREAR`
	`    private class OyenteBuscar implements ActionListener {`
	
	`        @Override`
	`        public void actionPerformed(ActionEvent evento) {`
	`            String marca = cajaMarca.getText();`
	`            String modelo = cajaModelo.getText();`
	`            Coche cocheBuscar = new Coche(marca, modelo, 2010);`
	`            String resultado = "";`
	`            for (Coche unCoche: coches) {`
	`                if (unCoche.equals(cocheBuscar)) {`
	`                    resultado += unCoche + "\n";`
	`                }`
	`            }`
	`            JOptionPane.showMessageDialog(null, resultado);`
	`        }`
	`    }`
	`}`	

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

Question author's solution:

```
// Importamos las librerias necesarias
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class CocheGuiEventos {

   // Programa principal
   public static void main(String[] args) {

      // Crea el ArrayList de datos sobre los que vamos a trabajar
      ArrayList<Coche> listaCoches = new ArrayList<Coche>();
      listaCoches.add(new Coche("Opel", "Meriva", 2013));
      listaCoches.add(new Coche("Ford", "Fiesta", 2015));
      listaCoches.add(new Coche("Honda", "Civic", 2016));
      listaCoches.add(new Coche("Renault", "Clio", 2014));
      listaCoches.add(new Coche("Ford", "Fiesta", 2012));

      // Creamos la interfaz gráfica y le pasamos los datos
      CocheGuiEventos gui = new CocheGuiEventos(listaCoches);
   }

   // Atributos
   JTextField cajaMarca;
   JTextField cajaModelo;
   ArrayList<Coche> coches;         // Datos con los que vamos a trabajar

   // Constructor
   public CocheGuiEventos(ArrayList<Coche> coches) {

      // Guardamos el ArrayList de Coches recibido en el atributo coches
      this.coches = coches;

      // Crea y configura la ventana principal
      JFrame ventana = new JFrame();
      ventana.setTitle("AÑADIR COCHE");
      ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      ventana.setLayout(new BorderLayout());

      // Crea y configura la etiqueta de la zona norte
      JLabel titulo = new JLabel("Vas a buscar un coche");
      ventana.add(titulo, BorderLayout.NORTH);

      // Crea y configura la zona centro.
      // Crea las etiquetas y las cajas de texto y las añade al marco secundario
      // El marco secundario lo añade a la zona centro
      JPanel centro = new JPanel(new GridLayout(2, 2));
      centro.add(new JLabel("Marca: "));
      cajaMarca = new JTextField(10);
      centro.add(cajaMarca);
      centro.add(new JLabel("Modelo: "));
      cajaModelo = new JTextField(10);
      centro.add(cajaModelo);
      ventana.add(centro, BorderLayout.CENTER);

      // Crea y configura el botón de la zona sur
      JButton botonBuscar = new JButton("BUSCAR");
      ventana.add(botonBuscar, BorderLayout.SOUTH);
```

```
    // CREAR OYENTES Y REGISTRARLOS
    OyenteBuscar pulsarBuscar = new OyenteBuscar();
    botonBuscar.addActionListener(pulsarBuscar);

    // Al final. Cuando se haya creado todo se visualiza
    ventana.pack();
    ventana.setVisible(true);
  }

  // CREAR OYENTE ActionListener PARA EL BOTON CREAR
  private class OyenteBuscar implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent evento) {
       String marca = cajaMarca.getText();
       String modelo = cajaModelo.getText();
       Coche cocheBuscar = new Coche(marca, modelo, 2010);
       String resultado = "";
       for (Coche unCoche: coches) {
          if (unCoche.equals(cocheBuscar)) {
             resultado += unCoche + "\n";
          }
       }
       JOptionPane.showMessageDialog(null, resultado);
    }
  }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

Finalizar revisión