



Navegación por el cuestionario



Finalizar revisión

Comenzado el	domingo, 17 de febrero de 2019, 07:15
Estado	Finalizado
Finalizado en	domingo, 17 de febrero de 2019, 20:38
Tiempo empleado	13 horas 22 minutos
Calificación	7,07 de 10,00 (71%)

Pregunta 1

Correcta

Puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

¿Cuáles de los siguientes métodos nos permiten insertar elementos de un `TreeSet` ?

Seleccione una:

- ☐ a. `offer()`
- ☐ b. `append()`
- ☒ c. `add()` ✓
- ☐ d. `insert()`

La respuesta correcta es: `add()`

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 2

Correcta

Puntúa 0,00 sobre 1,00

🚩 Marcar pregunta

Para expresar que una clase es genérica, hay que indicar el parámetro formal que podrá ser de cualquier tipo de objeto entre `"<"` . ¿Verdadero o falso?

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

La respuesta correcta es 'Verdadero'

Correcta

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

Pregunta 3

Correcta

Puntúa 0,40 sobre 1,00

🚩 Marcar pregunta

Tenemos el siguiente ArrayList:

```
lista = ["ROJO", "BLANCO", "AZUL", "NEGRO"]
```

Indica los valores de las siguientes expresiones:

Expresión	Valor
<code>lista.size()</code>	4 ✓
<code>lista.indexOf("AZUL")</code>	2 ✓
<code>lista.indexOf("azul")</code>	-1 ✓
<code>lista.get(1);</code>	BLANCO ✓
<code>lista.get(lista.size());</code>	NEGRO ✓

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,40/1,00.

Pregunta 4

Incorrecta

Puntúa 0,00 sobre 1,00

🚩 Marcar pregunta

Tenemos el siguiente ArrayList:

```
ArrayList<String> lista = new ArrayList<String>();
```

Indica el contenido del ArrayList despues de las siguientes expresiones:

Expresión	System.out.println(lista);
<code>lista.add("azul");</code>	[azul] ✗
<code>lista.add(0, "blanco");</code>	["blanco", "azul"] ✗
<code>lista.add("rojo");</code>	["blanco", "azul", "rojo"] ✗
<code>lista.remove(1);</code>	["blanco", "rojo"] ✗
<code>lista.add(1, "verde");</code>	["blanco", "verde", "rojo"] ✗

Incorrecta

Puntos para este envío: 0,00/1,00.

Pregunta 5

Correcta

Puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

Indica la manera correcta de crear un ArrayList que almacene cadenas de caracteres:

Seleccione una:

- ☐ a. `ArrayList lista = new ArrayList();`
- ☒ b. `ArrayList<String> lista = new ArrayList<String>();` ✓
- ☐ c. `ArrayList[String] lista = new ArrayList[String]();`
- ☐ d. `ArrayList<String> = new ArrayList<String>();`
- ☐ e. `ArrayList<String> lista = new ArrayList();`

Respuesta correcta

La respuesta correcta es: `ArrayList<String> lista = new ArrayList<String>();`

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 6

Correcta

Puntúa 1,00 sobre 1,00

🚩 Marcar

Corrige el método `buscarPalabra` para que reciba un ArrayList de String y una palabra, busque la palabra en la lista y devuelva el índice de esa palabra. Si no encuentra la palabra devolverá -1.

Cuando el método funcione el programa mostrará:

```
[rosa, verde, rojo, negro]  
-1
```

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

Reiniciar respuesta

```

8      lista.add("rojo");
9      lista.add("negro");
10
11     System.out.println(lista);
12     System.out.println(buscarPalabra(lista, "azul"));
13     System.out.println(buscarPalabra(lista, "negro"));
14     System.out.println(buscarPalabra(lista, "verde"));
15 }
16
17 // Metodo
18 public static int buscarPalabra(ArrayList<String> lista, String palabra) {
19     for (String elemento : lista) {
20
21         if (elemento.equals(palabra)) {
22             return lista.indexOf(palabra);
23         }
24     }
25     return -1;
26 }
27 }

```

	Expected	Got	
✓	[rosa, verde, rojo, negro]	[rosa, verde, rojo, negro]	✓
	-1	-1	
	3	3	
	1	1	

Todas las pruebas superadas. ✓

Question author's solution:

```

import java.util.*;

public class ArrayLists {
    public static void main(String[] args) {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("rosa");
        lista.add("verde");
        lista.add("rojo");
        lista.add("negro");

        System.out.println(lista);
        System.out.println(buscarPalabra(lista, "azul"));
        System.out.println(buscarPalabra(lista, "negro"));
        System.out.println(buscarPalabra(lista, "verde"));
    }

    // Metodo
    public static int buscarPalabra(ArrayList<String> lista, String palabra) {
        for (String elemento : lista) {
            if (elemento.equals(palabra)) {
                return lista.indexOf(palabra);
            }
        }
        return -1;
    }
}

```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 7

Correcta

Puntúa 0,67 sobre 1,00

⚑ Marcar pregunta

Ordena el código del método para que recorra un ArrayList y devuelva el número de letras de la palabra más corta que encuentre utilizando el bucle for-each:

```

public static int metodo(ArrayList<String> palabras) {
    String primera = palabras.get(0);
    int min = primera.length();
    for(String palabra : palabras) {
        int largo = palabra.length();
        min = Math.min(largo, min);
    }
    return min;
}

```

Respuesta correcta

La respuesta correcta es:

Ordena el código del método para que recorra un ArrayList y devuelva el número de letras de la palabra más corta que encuentre utilizando el bucle for-each:

```

public static int metodo(ArrayList<String> palabras) {
    [String primera = palabras.get(0);]
    [int min = primera.length();]
    [for(String palabra : palabras) {}]
    [int largo = palabra.length();]
    [min = Math.min(largo, min);]
    }
    return min;
}

```

Correcta

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

Pregunta 8

Correcta

Puntúa 1,00 sobre 1,00

⚑ Marcar pregunta

Ordena el código del método para que recorra un ArrayList y devuelva la palabra más larga que encuentre:

```

public static String metodo(ArrayList<String> palabras) {
    String palabraMax = palabras.get(0);
    int max = palabraMax.length();
    for(int i = 1; i < palabras.size(); i++) {
        String palabra = palabras.get(i);
        int largo = palabra.length();
        if (largo > max) {
            max = largo;
            palabraMax = palabra;
        }
    }
    return palabraMax;
}

```

return max;

for(int i = 0; i < palabras.size(); i++) {

Respuesta correcta

La respuesta correcta es:
Ordena el código del método para que recorra un ArrayList y devuelva la palabra más larga que encuentre:

```
public static String metodo(ArrayList<String> palabras) {  
    [String palabraMax = palabras.get(0);]  
    [int max = palabraMax.length();]  
    [for(int i = 1; i < palabras.size(); i++) {}]  
    [String palabra = palabras.get(i);]  
    [int largo = palabra.length();]  
    [if (largo > max) {  
        max = largo;  
        palabraMax = palabra;  
    }]  
    [return palabraMax;]  
}
```

Correcta
Puntos para este envío: 1,00/1,00.

Pregunta 9
Correcta
Puntúa 1,00 sobre 1,00
Marcar pregunta

Crea el método intercambiar para que reciba un ArrayList de String e intercambie su contenido por parejas: la primera con la segunda, la tercera con la cuarta... Si la lista tiene un número impar de elementos, el último se quedará igual.
Cuando el método funcione, el programa mostrará:

[verde, rosa, negro, rojo]
[rojo, verde, negro]

PISTA: Comprueba cuántos elementos hay que intercambiar antes de empezar a recorrer la lista. Te dirá cuando debes acabar de recorrerla.

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

```
Reiniciar respuesta  
17 System.out.println(lista);  
18 }  
19  
20 public static void intercambiar(ArrayList<String> palabras) {  
21     int tamanholista = palabras.size();  
22     for(int i = 0; i < tamanholista; i++) {  
23         String guardarPalabra = palabras.get(i);  
24         if(tamanholista % 2 == 0) {  
25             palabras.remove(i);  
26             palabras.add(i + 1, guardarPalabra);  
27             i++;  
28         }else{  
29             palabras.remove(i);  
30             palabras.add(i + 1, guardarPalabra);  
31             i += 2;  
32         }  
33     }  
34 }  
35  
36 }
```

	Expected	Got	
✓	[verde, rosa, negro, rojo] [rojo, verde, negro]	[verde, rosa, negro, rojo] [rojo, verde, negro]	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
import java.util.*;  
public class Programa {  
    public static void main(String[] args) {  
        ArrayList<String> lista = new ArrayList<String>();  
        lista.add("rosa");  
        lista.add("verde");  
        lista.add("rojo");  
        lista.add("negro");  
        intercambiar(lista);  
        System.out.println(lista);  
  
        lista.clear();  
        lista.add("verde");  
        lista.add("rojo");  
        lista.add("negro");  
        intercambiar(lista);  
        System.out.println(lista);  
    }  
  
    public static void intercambiar(ArrayList<String> lista) {  
        int largo = lista.size();  
        if (largo % 2 == 1) {  
            largo = largo - 1;  
        }  
        for (int i = 0; i < largo; i = i + 2) {  
            String elemento = lista.get(i);  
            lista.set(i, lista.get(i + 1));  
            lista.set(i+ 1, elemento);  
        }  
    }  
}
```

Correcta
Puntos para este envío: 1,00/1,00.

Pregunta 10
Correcta
Puntúa 1,00 sobre 1,00
Marcar pregunta

Crea el método eliminar para que reciba un ArrayList de String y elimine todas las palabras que tenga un número impar de letras.
Cuando el método funcione, el programa mostrará:

[rosa, rojo]

Prueba a utilizar un iterador para recorrer la lista.

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

```
Reiniciar respuesta  
1 import java.util.*;  
2 public class Programa {  
3     public static void main(String[] args) {  
4         ArrayList<String> lista = new ArrayList<String>();  
5         lista.add("rosa");  
6         lista.add("verde");  
7         lista.add("rojo");  
8         lista.add("negro");  
9         eliminar(lista);  
10        System.out.println(lista);  
11    }  
12  
13    public static void eliminar(ArrayList<String> palabras) {  
14        for(int i = 0; i < palabras.size(); i++) {  
15            String palabra = palabras.get(i);  
16            int countWord = palabra.length();  
17            if(countWord % 2 == 1) {  
18                palabras.remove(i);  
19            }  
20        }  
21    }  
22 }
```

	Expected	Got	

Expected	Got
✓ [rosa, rojo]	[rosa, rojo] ✓

Todas las pruebas superadas. ✓

Question author's solution:

```
import java.util.*;

public class Programa {

    public static void main(String[] args) {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("rosa");
        lista.add("verde");
        lista.add("rojo");
        lista.add("negro");
        eliminar(lista);
        System.out.println(lista);
    }

    public static void eliminar (ArrayList<String> lista) {
        Iterator<String> iterador = lista.iterator();
        while (iterador.hasNext()) {
            String palabra = iterador.next();
            if (palabra.length() % 2 == 1)
                iterador.remove();
        }
    }
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Finalizar revisión



Urrutiko Lanbide Heziketako Institutua
Instituto de Formación Profesional a Distancia

Contacta con nosotros

📍 Dirección: Calle Álava 41, interior - Vitoria-Gasteiz
☎️ Teléfono : 945 567 953
✉️ E-mail: ulhi@ulhi.net
🐦 Twitter: @UrrutikoLH



EUSKO JAURLARITZA
GOBIERNO VASCO

HEZKUNTZA SAILA
Lanbide Heziketa Zehazkuntza
DEPARTAMENTO DE EDUCACIÓN
Viceconsejería de Formación Profesional