



Navegación por el cuestionario



Finalizar revisión

| | |
|-----------------|---|
| Comenzado el | lunes, 18 de febrero de 2019, 11:12 |
| Estado | Finalizado |
| Finalizado en | miércoles, 20 de febrero de 2019, 13:09 |
| Tiempo empleado | 2 días 1 hora |
| Calificación | 8,15 de 10,00 (82%) |

Pregunta 1

Correcta

Puntuá 1,00 sobre 1,00

⚑ Marcar pregunta

En la clase **Double** disponemos de un método que nos permite pasar una cadena, que contiene un número, a un tipo de dato numérico con el que podemos hacer operaciones, ¿cuál de las siguientes líneas sería una conversión válida?

Seleccione una:

- ☐ a. `double d = Double.dobleValue("33.4");`
- ☐ b. `double d = Double.fromString("33.4");`
- ☐ c. `double d = Double.valueOf("33.4").toDouble();`
- ☒ d. `double d = Double.parseDouble("33.4");` ✓

La respuesta correcta es: `double d = Double.parseDouble("33.4");`

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 2

Correcta

Puntuá 0,90 sobre 1,00

⚑ Marcar pregunta

Dada una lista t de datos tipo Integer, ¿cómo podríamos mostrar todos los elementos de la lista?

Seleccione una:

- ☐ a. No se puede tener una lista de datos tipo **Integer** dado que son objetos inmutables.

☐ b.

```
for (t : Integer j) {  
    System.out.println(j);  
}
```

☒ c.

```
Iterator it = t.iterator();  
while(it.hasNext()) {  
    System.out.println(it.next());  
}
```

☐ d.

```
for (Integer j : t) {  
    System.out.println(t);  
}
```

La respuesta correcta es:

```
Iterator it = t.iterator();  
while(it.hasNext()) {  
    System.out.println(it.next());  
}
```

Correcta

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

Pregunta 3

Correcta

Puntuá 0,75 sobre 1,00

⚑ Marcar pregunta

Tenemos el siguiente código:

```
public static void metodo(ArrayList<Integer> numeros) {  
    for (int i = numeros.size() - 1; i > 0; i--) {  
        int numero = numeros.get(i);  
        if (numero % 2 == 1) {  
            numeros.add(numero);  
        } else {  
            numeros.add(0, numero);  
        }  
    }  
}
```

¿Qué se verá por consola en los siguientes casos?

| Lista inicial | Lista final |
|--------------------|------------------------------------|
| [2, 5, 1, 8] | [2, 8, 2, 5, 1, 8, 5] ✓ |
| [15, 10, 5, 30] | [30, 10, 30, 15, 10, 5, 30] ✓ |
| [-3, 13, 5, 7, 13] | [-3, 13, 5, 7, 13, 13, 7, 5, 13] ✓ |
| [12, 0, 2, 12, -2] | [2, 12, 2, -2, 12, 0, 2, 12, -2] ✓ |

Correcta

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

Pregunta 4

Correcta

Puntuá 0,70 sobre 1,00

⚑ Marcar pregunta

¿Cuál de las siguientes expresiones encajan con la expresión regular **"[A-Z0-9]+0-9"** ?

Seleccione una:

- ☒ a. `"AZ090-9"` ✓
- ☐ b. `"AAA09"`
- ☐ c. `"ABC9+0"`
- ☐ d. `"0AZZ-9"`

La respuesta correcta es: `"AZ090-9"`

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,70/1,00.

Pregunta 5

Correcta

Puntuá 1,00 sobre 1,00

⚑ Marcar pregunta

¿Cuáles de las siguientes cadenas encajan con la expresión regular **"(\\|+([a-z]))*\\|++"** ?

Seleccione una:

- ☐ a. `"a*b+"`
- ☐ b. `"++++"`
- ☐ c. `"*a"`
- ☒ d. `"a*b*+"` ✓

La respuesta correcta es: "++"

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 6

Correcta

Puntúa 1,00 sobre 1,00

⚑ Marcar pregunta

Ordena el código del método para que recorra un ArrayList de enteros y elimine los números que sean divisibles por el valor pasado como parámetros:

```
public static void eliminarDivisible(ArrayList<Integer> numeros, int divisor) {  
    Iterator<Integer> iterador = numeros.iterator();  
    while (iterador.hasNext()) {  
        int numero = iterador.next();  
        if (numero % divisor == 0) {  
            iterador.remove();  
        }  
    }  
}
```

Respuesta correcta

La respuesta correcta es:

Ordena el código del método para que recorra un ArrayList de enteros y elimine los números que sean divisibles por el valor pasado como parámetros:

```
public static void eliminarDivisible(ArrayList<Integer> numeros, int divisor) {  
    [Iterator<Integer> iterador = numeros.iterator();]  
    [while (iterador.hasNext()) {}]  
    [int numero = iterador.next();]  
    [if (numero % divisor == 0) {}]  
    [iterador.remove();]  
    }  
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 7

Correcta

Puntúa 0,80 sobre 1,00

⚑ Marcar pregunta

Corrige el método minPrimer para que obtenga el valor menor del ArrayList pasado como parámetros y lo inserte al comienzo de la lista. El resultado del programa será:

Lista inicial: [3, -5, 18, 10]
Lista final: [-5, 3, 18, 10]

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

Reiniciar respuesta

```
1 import java.util.*;  
2 public class Programa {  
3     public static void main(String[] args) {  
4         ArrayList<Integer> lista = new ArrayList<Integer>();  
5         lista.add(3);  
6         lista.add(-5);  
7         lista.add(18);  
8         lista.add(10);  
9  
10        System.out.println("Lista inicial: " + lista);  
11        minPrimer(lista);  
12        System.out.println("Lista final: " + lista);  
13    }  
14    public static void minPrimer(ArrayList<Integer> lista) {  
15        int min = lista.get(0);  
16        int indice = 0;  
17        for (int i = 1; i < lista.size(); i++) {  
18            int num = lista.get(i);  
19            if (num < min) {  
20                min = num;  
                ..  
            }  
        }  
    }
```

| | Expected | Got | |
|---|--|--|---|
| ✓ | Lista inicial: [3, -5, 18, 10] Lista final: [-5, 3, 18, 10] | Lista inicial: [3, -5, 18, 10] Lista final: [-5, 3, 18, 10] | ✓ |

Todas las pruebas superadas. ✓

Question author's solution:

```
import java.util.*;  
  
public class Programa {  
    public static void main(String[] args) {  
        ArrayList<Integer> lista = new ArrayList<Integer>();  
        lista.add(3);  
        lista.add(-5);  
        lista.add(18);  
        lista.add(10);  
  
        System.out.println("Lista inicial: " + lista);  
        minPrimer(lista);  
        System.out.println("Lista final: " + lista);  
    }  
  
    public static void minPrimer(ArrayList<Integer> lista) {  
        int min = lista.get(0);  
        int indice = 0;  
        for (int i = 1; i < lista.size(); i++) {  
            int num = lista.get(i);  
            if (num < min) {  
                min = num;  
                indice = i;  
            }  
        }  
        lista.remove(indice);  
        lista.add(0, min);  
    }  
}
```

Correcta

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

Pregunta 8

Correcta

Puntúa 1,00 sobre 1,00

⚑ Marcar pregunta

Tenemos el siguiente código:

```
public static ArrayList<String> crearLista() throws FileNotFoundException {  
    ArrayList<String> lista = new ArrayList<String>();  
    Scanner leerDatos = new Scanner(new File("fechas.txt"));  
    Pattern formatoFecha = Pattern.compile("\\d{1,2}-\\d{1,2}-\\d{2}");  
    while (leerDatos.hasNext()) {  
        String fecha = leerDatos.next();  
        Matcher comparaFormato = formatoFecha.matcher(fecha);  
        if (comparaFormato.matches()) {  
            lista.add(fecha);  
        }  
    }  
    return lista;  
}
```

```
}
```

indica qué afirmaciones son válidas.

Seleccione una o más de una:

☒ a. Lee fechas del fichero "fechas.txt" y las que cumplen con el formato dd-mm-aa o d-m-aa las guarda en una lista de String

☐ b. La fecha 12-5-1992 se guardará en la lista

☒ c. La fecha 12-5-92 se guardará en la lista

☐ d. La fecha 2-5-92 NO se guardará en la lista

Respuesta correcta

La respuesta correcta es: Lee fechas del fichero "fechas.txt" y las que cumplen con el formato dd-mm-aa o d-m-aa las guarda en una lista de String, La fecha 12-5-92 se guardará en la lista

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 9

Correcta

Puntúa 1,00 sobre 1,00

Marcar pregunta

Escribe el código para el método contarTiradas. Recibirá como parámetro un ArrayList de enteros entre 0 y MAX y devolverá un array que cuente cuantas veces se repite cada número.

Quando el método funcione, el programa mostrará lo siguiente:

```
[3, 0, 3, 0, 4, 4, 1, 4, 3, 4, 1, 5, 3, 0, 3, 0, 4, 1, 1, 4, 5, 1, 3, 2, 5]
[4, 5, 1, 6, 6, 3]
```

4 indica que el 0 se repite 4 veces, 5 que el 1 aparece 5 veces...

Prueba a utilizar un bucle for-each.

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

Reiniciar respuesta

```
7 -      for (int i = 0; i < 25; i++) {
8         int num = numAzar.nextInt(MAX);
9         lista.add(num);
10      }
11
12      System.out.println(lista);
13      int[] contador = contarTiradas(lista);
14      System.out.println(Arrays.toString(contador));
15  }
16 -  public static int[] contarTiradas(ArrayList<Integer> listaNum) {
17      int[] tiradas = new int[MAX];
18      for (int i = 0; i < listaNum.size(); i++){
19          int num = listaNum.get(i);
20          tiradas[num]++;
21      }
22
23      return tiradas;
24  }
25
26 }
```

| | Expected | Got |
|--|---|--|
| | <pre>[3, 0, 3, 0, 4, 4, 1, 4, 3, 4, 1, 5, 3, 0, 3, 0, 4, 1, 1, 4, 5, 1, 3, 2, 5] [4, 5, 1, 6, 6, 3]</pre> | <pre>[3, 0, 3, 0, 4, 4, 1, 4, 3, 4, 1, 5, 3, 0, 3, 0, 4, 1, 1, 4, 5, 1, [4, 5, 1, 6, 6, 3]</pre> |

Todas las pruebas superadas.

Question author's solution:

```
import java.util.*;
public class Programa {
    public static final int MAX = 6;
    public static void main(String[] args) {
        Random numAzar = new Random(10);
        ArrayList<Integer> lista = new ArrayList<Integer>();
        for (int i = 0; i < 25; i++) {
            int num = numAzar.nextInt(MAX);
            lista.add(num);
        }

        System.out.println(lista);
        int[] contador = contarTiradas(lista);
        System.out.println(Arrays.toString(contador));
    }

    public static int[] contarTiradas(ArrayList<Integer> numeros) {
        int[] contar = new int[MAX];
        for (int num : numeros) {
            contar[num]++;
        }
        return contar;
    }
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 10

Incorrecta

Puntúa 0,00 sobre 1,00

Marcar pregunta

Escribe el código para un programa que pida 5 codigos con el siguiente formato:

```
3 letras mayúsculas
2 dígitos
1 letra mayúscula
Por ejemplo: FSP56B
```

Si el dato introducido cumple con el formato, lo almacenará en un ArrayList de cadenas de caracteres.

Finalmente muestra el contenido de la lista. Si los códigos introducidos son: ABC12D, PSW12d, ArL23B, GLA81M y 12ADR2

```
[3, 0, 3, 0, 4, 4, 1, 4, 3, 4, 1, 5, 3, 0, 3, 0, 4, 1, 1, 4, 5, 1, 3, 2, 5]
[4, 5, 1, 6, 6, 3]
```

Prueba a utilizar un bucle for-each.

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

```
1 import java.util.regex.*; // Para poder usar patrones
2 import java.util.*;
3 public class Ejercicio10 {
4     public static void main(String[] args) {
5         Scanner teclado = new Scanner(System.in);
6         int contOK = 0;
7         ArrayList<String> codigosOK = new ArrayList<String>();
8         String codigo = "";
9         Pattern formatoString = Pattern.compile("[A-Z]{3}[0-9]{2}[A-Z]{1}");
10        Matcher comparaFormato =
11        while(contOK != 5) {
12            do {
13                codigo = teclado.next();
14                comparaFormato = formatoString.matcher(codigo);
15            }while(!comparaFormato.matches());
16            codigosOK.add(codigo);
17            contOK++;
18        }
19        System.out.print(codigosOK);
20    }
```

| | Input | Expected | Got |
|----|-------|----------|-----|
| .. | | | |

| | Input | Expected | Got | |
|---|--|------------------|---|---|
| ✖ | ABC12D PSW12d ArL23B GLA81M 12ADR2 | [ABC12D, GLA81M] | ***Error*** Exception in thread "main" java.util.NoSuchElementException at java.util.Scanner.throwFor(Scanner.java:862) at java.util.Scanner.next(Scanner.java:1371) at Ejercicio10.main(Ejercicio10.java:13) | ✖ |

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

Mostrar diferencias

Question author's solution:

```
import java.util.*;
import java.util.regex.*;

public class Programa {
    public static void main(String[] args) {
        Scanner leerTeclado = new Scanner(System.in);
        ArrayList<String> lista = new ArrayList<String>();
        Pattern formatoCodigo = Pattern.compile("[A-Z]{3}[0-9]{2}[A-Z]*");
        for (int i = 0; i < 5; i++) {
            String codigo = leerTeclado.next();
            Matcher comparaFormato = formatoCodigo.matcher(codigo);
            if (comparaFormato.matches()) {
                lista.add(codigo);
            }
        }

        System.out.println(lista);
    }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

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 SAIA
Lanbide Heziketako Sailburuordetza
DEPARTAMENTO DE EDUCACIÓN
Ministerio de Formación Profesional