

El1013/MT1013 Estructuras de datos
Primer control parcial
Lunes, 14 de Octubre de 2019

☐0 ☐0 ☐0 ☐0 ☐0 ☐0 ☐0 ☐0 ☐0 ← Marca los dígitos numéricos de tu DNI justificados a la derecha.

☐1 ☐1 ☐1 ☐1 ☐1 ☐1 ☐1 ☐1 ☐1

☐2 ☐2 ☐2 ☐2 ☐2 ☐2 ☐2 ☐2 ☐2

☐3 ☐3 ☐3 ☐3 ☐3 ☐3 ☐3 ☐3 ☐3

☐4 ☐4 ☐4 ☐4 ☐4 ☐4 ☐4 ☐4 ☐4

☐5 ☐5 ☐5 ☐5 ☐5 ☐5 ☐5 ☐5 ☐5

☐6 ☐6 ☐6 ☐6 ☐6 ☐6 ☐6 ☐6 ☐6

☐7 ☐7 ☐7 ☐7 ☐7 ☐7 ☐7 ☐7 ☐7

☐8 ☐8 ☐8 ☐8 ☐8 ☐8 ☐8 ☐8 ☐8

☐9 ☐9 ☐9 ☐9 ☐9 ☐9 ☐9 ☐9 ☐9

Nombre
 Apellidos
 DNI:

IMPORTANTE

- La duración de la prueba es de 1 hora.
- Cada cuestión tiene una única respuesta correcta.
- Indica las respuestas marcando el recuadro correcto con una cruz ☒ o llenándolo ☐.
- Utiliza un bolígrafo negro o azul para marcar las respuestas.
- Si marcas alguna casilla por error indica que es una marca incorrecta escribiendo "no" a su lado.
- Cada respuesta correcta suma 1 punto; una incorrecta resta 0.3 puntos; y las cuestiones en blanco valen 0 puntos.

[C01]. Sea una lista `List<String> l`, ¿cómo crearías un `ListIterator<String>` para recorrerla?

- ☒ `ListIterator<String> iter = l.listIterator();`
- ☐ `ListIterator<String> iter = new ListIterator<String>(l);`
- ☐ `Iterator<String> iter = l.listIterator();`
- ☐ `ListIterator<String> iter = l.iterator(0);`

[C03]. ¿Cuál de estos métodos no pertenece a la interfaz `Iterator<T>` ?

- ```

☐ boolean hasNext()
☐ void remove()
☒ void add(T e)
☐ T next()

```

[C08]. ¿Cuál de las siguientes clases de la *Java Collection Framework* NO ORDENA los elementos ?

- ☐ LinkedList<T>
 ☒ HashSet<T>  
☐ PriorityQueue<T>
 ☐ TreeSet<T>

## CATÁLOGO

[C09]. Dadas dos listas de enteros,  $a = [12, 5, 3, 7, 8, 0, 3]$  y  $b = [10, 2, 3, 0]$ , ¿qué valor contendrá la variable  $a$  tras la sentencia  $x = a.\text{retainAll}(b)$ ?

- ☐ [12, 5, 7, 8]  
☐ true

- ☐ [10, 2, 3, 0]  
☒ [3, 0, 3]

[C02]. Dado el siguiente método

```
public void imprime(PriorityQueue<Integer> q) {
 Iterator<Integer> it = q.iterator();
 while(it.hasNext())
 System.out.println(it.next());
}
```

¿Cuál de las siguientes afirmaciones **ES CORRECTA**?

- ☒ Imprime los elementos de la cola en la consola, siendo el primero el menor de todos ellos y el resto desordenados.  
☐ Imprime los elementos de la cola en la consola ordenados de menor a mayor.  
☐ Imprime los elementos de la cola en la consola ordenados de mayor a menor.  
☐ No es correcto ya que la clase PriorityQueue no dispone del método iterator().

[C04]. Deseamos crear un nuevo `HashSet<String> newS` que sea copia de una ya existente anterior `oldS`, ¿cuál de estos fragmentos de código **NO ES CORRECTO**?

- ☐ `HashSet<String> newS = new HashSet<String>(oldS);`  
☐ `HashSet<String> newS = new HashSet<String>();`  
    `newS.addAll(oldS);`  
☒ `HashSet<String> newS = oldS;`  
☐ `HashSet<String> newS = new HashSet<String>();`  
    for(String data: oldS)  
        `newS.add(data);`

[C05]. Dada una lista de enteros `LinkedList<Integer> l` deseamos borrar todos sus elementos. ¿Cuál de estas formas **NO ES CORRECTA**?

- ☐ `l.clear();`  
☐ `l.retainAll(new LinkedList<Integer>());`  
☐ `l = new LinkedList<Integer>();`  
☒ `l.removeAll(new LinkedList<Integer>());`

## CATÁLOGO

[C06]. Tenemos un diccionario `Map<String, Integer> map` y deseamos mostrar por pantalla todos los valores de los pares almacenados en el diccionario. ¿Cuál de estos fragmentos NO ES CORRECTO?

- ☒

```
Iterator<Integer> iter = map.iterator();
while(iter.hasNext()) {
 int value = iter.next();
 System.out.println(value);
}
```
- ☐

```
Iterator<String> iter = map.keySet().iterator();
while(iter.hasNext()) {
 int value = map.get(iter.next());
 System.out.println(value);
}
```
- ☐

```
Iterator<Integer> iter = map.values().iterator();
while(iter.hasNext()) {
 int value = iter.next();
 System.out.println(value);
}
```
- ☐

```
for (String key: map.keySet())
 System.out.println(map.get(key));
```

[C07]. Dadas las siguientes declaraciones:

```
interface I {
 void metodo1();
}
```

```
public class A implements I {
 void metodo1() {...};
 void metodo3() {...};
}
```

```
interface J extends I {
 void metodo2();
}
```

```
public class B extends A implements J {
 void metodo2() {...};
 void motodo4() {...};
}
```

¿Cuál de las siguientes expresiones NO ES CORRECTA?

- ☐

```
I obj = new B();
obj.metodo1();
```
- ☒

```
J obj = new A();
obj.metodo2();
```
- ☐

```
A obj = new B();
obj.metodo3();
```
- ☐

```
B obj = new B();
obj.metodo4();
```

[C10]. Dado el siguiente fragmento de código:

```
B.clear()
for (Collection<Integer> col: A) {
 ListIterator<String> it = col.listIterator();
 while(it.hasNext()) {
 String s = it.next()
 if (B.containsKey(s))
 B.put(s, B.get(s)+1)
 else
 B.put(s, 1);
 }
}
```

¿Cuáles ha de ser la declaración de las variables A y B para que el código sea correcto?

☒ Set<List<String>> A;  
Map<String, Integer> B;

☐ Set<String> A;  
Map<String, Integer> B;

☐ List<List<String>> A;  
Map<String, String> B;

☐ List<Set<String>> A;  
Map<Integer, String> B;

## CATÁLOGO