

**1) Crea un proyecto que se llame EjerTresSetInt**

Escribe un programa en Java que genere un número aleatorio de datos (entre 1 y 15 números) los valores de dichos números estarán comprendidos entre 1000 y 8000:

- En un objeto de clase **HashSet**.
- En un objeto de clase **LinkedHashSet**.
- En un objeto de clase **TreeSet**.

Una vez hecho hace un recorrido de cada objeto y muestra todos sus elementos.

Muestra en pantalla el resultado de invocar al método toString de cada uno de los objetos.

Ejemplo de ejecución:

```
Se van a generar 6 datos.
```

```
Insertando:
```

```
5187 * 1361 * 1856 * 3621 * 7605 * 7740 *
```

```
Valores de objHashSet:
```

```
1856 1361 5187 3621 7605 7740
```

```
Valores de objLinkHashSet:
```

```
5187 1361 1856 3621 7605 7740
```

```
Valores de objTreeSet:
```

```
1361 1856 3621 5187 7605 7740
```

```
Valores de objHashSet: [1856, 1361, 5187, 3621, 7605, 7740]
```

```
Valores de objLinkHashSet: [5187, 1361, 1856, 3621, 7605, 7740]
```

```
Valores de objTreeSet: [1361, 1856, 3621, 5187, 7605, 7740]
```

**2) Crea un proyecto que se llame EjerTresSetString**

Hacer lo mismo que en el ejercicio anterior pero con conjuntos de String y asignando los elementos al azar a partir de un vector de Strings.

Ejemplo de ejecución:

```
Se van a generar 8 datos.
```

```
Insertando:
```

```
este * montaña * letra * bajo * frio * bajo * bajo * aleatorio *
```

```
Valores de objHashSet (no hay orden):
```

```
este frio aleatorio bajo letra montaña
```

```
Valores de objLinkHashSet (por orden de inserción):
```

```
este montaña letra bajo frio aleatorio
```

```
Valores de objTreeSet (orden alfabético):
```

```
aleatorio bajo este frio letra montaña
```

```
Valores de objHashSet: [este, frio, aleatorio, bajo, letra, montaña]
```

```
Valores de objLinkHashSet: [este, montaña, letra, bajo, frio, aleatorio]
```

```
Valores de objTreeSet: [aleatorio, bajo, este, frio, letra, montaña]
```

**3) Crea un proyecto que se llame EjerDosListInt**

Escribe un programa en Java que genere un número aleatorio de datos que consisten en números al azar con valores entre 1 y 15 y los va insertando:

- En un objeto de clase **ArrayList**.
- En un objeto de clase **LinkedList**.

Una vez hecho hace un recorrido de cada objeto y muestra todos sus elementos.

Muestra en pantalla el resultado de invocar al método toString de cada uno de los objetos.

Ejemplo de ejecución:

Se van a generar 14 datos.

Insertando:

15 \* 4 \* 9 \* 1 \* 6 \* 12 \* 13 \* 15 \* 1 \* 11 \* 5 \* 8 \* 12 \* 8 \*

Valores de objArrayList:

15 4 9 1 6 12 13 15 1 11 5 8 12 8

Valores de objLinkedList:

15 4 9 1 6 12 13 15 1 11 5 8 12 8

Valores de objArrayList: [15, 4, 9, 1, 6, 12, 13, 15, 1, 11, 5, 8, 12, 8]

Valores de objLinkedList: [15, 4, 9, 1, 6, 12, 13, 15, 1, 11, 5, 8, 12, 8]

**4) Crea un proyecto que se llame EjerDosListString**

Hacer lo mismo que en el ejercicio anterior pero con conjuntos de String y asignando los elementos al azar a partir de un vector de Strings.

Ejemplo de ejecución:

Se van a generar 3 datos.

Insertando:

este \* este \* bajo \*

Valores de objArrayList:

este este bajo

Valores de objLinkedList:

este este bajo

Valores de objArrayList: [este, este, bajo]

Valores de objLinkedList: [este, este, bajo]

**5) Crea un proyecto que se llame ListarNumerosEnOrden**

Escribe un programa en Java que pida números al usuario hasta que teclee el -9999. Una vez dado ese valor escribe en orden creciente todos los elementos distintos de la secuencia de números introducida por el usuario.

Para almacenar los números que diga el usuario se creará un objeto TreeSet<Integer>.

**6) Crea un proyecto que se llame `ListarNumPalabrasOrdenadas`**

Escribe un programa en Java que:

- Pida al usuario que introduzca un número indeterminado de palabras. La introducción de datos terminará cuando introduzca un "\*" (que por supuesto no se tendrá en cuenta para hacer lo que se pide en el ejercicio).
- Pida que se diga un número entero positivo num que no puede ser mayor que el número de palabras distintas que se han dado.
- Muestre las num primeras palabras en orden alfabético de la lista de palabras que ha dado el usuario.

**7) Crea un proyecto que se llame `TreemapNotasAlumnosPorApellido`**

Escribe un programa en Java que:

- Pida el apellido (se supone que no se repiten) y la calificación obtenida por un alumno. La entrada de datos terminará cuando se introduzca como apellido "\*".
- Muestre un listado por orden alfabético de los datos de cada alumno.

**8) Crea un proyecto que se llame `TreemapNotasAlumnosPorNota`**

Modifica el programa del ejercicio anterior para que podamos mostrar un listado ordenado por notas, teniendo en cuenta que puede haber repeticiones de notas.