

10.3 Ejercicios



Ejercicio 1

Crea un `ArrayList` con los nombres de 6 compañeros de clase. A continuación, muestra esos nombres por pantalla. Utiliza para ello un bucle `for` que recorra todo el `ArrayList` sin usar ningún índice.



Ejercicio 2

Realiza un programa que introduzca valores aleatorios (entre 0 y 100) en un `ArrayList` y que luego calcule la suma, la media, el máximo y el mínimo de esos números. El tamaño de la lista también será aleatorio y podrá oscilar entre 10 y 20 elementos ambos inclusive.



Ejercicio 3

Escribe un programa que ordene 10 números enteros introducidos por teclado y almacenados en un objeto de la clase `ArrayList`.



Ejercicio 4

Realiza un programa equivalente al anterior pero en esta ocasión, el programa debe ordenar palabras en lugar de números.



Ejercicio 5

Realiza de nuevo el ejercicio de la colección de discos pero utilizando esta vez una lista para almacenar la información sobre los discos en lugar de un `array` convencional. Comprobarás que el código se simplifica notablemente ¿Cuánto ocupa el programa original hecho con un `array`? ¿Cuánto ocupa este nuevo programa hecho con una lista?



Ejercicio 6

Implementa el control de acceso al área restringida de un programa. Se debe pedir un nombre de usuario y una contraseña. Si el usuario introduce los datos correctamente, el programa dirá “Ha accedido al área restringida”. El usuario tendrá un máximo de 3 oportunidades. Si se agotan las oportunidades el programa dirá “Lo siento, no tiene acceso al área restringida”. Los nombres de usuario con sus correspondientes contraseñas deben estar almacenados en una estructura de la clase `HashMap`.



Ejercicio 7

La máquina *Eurocoin* genera una moneda de curso legal cada vez que se pulsa un botón siguiendo la siguiente pauta: o bien coincide el valor con la moneda anteriormente generada - 1 céntimo, 2 céntimos, 5 céntimos, 10 céntimos, 25 céntimos, 50 céntimos, 1 euro o 2 euros - o bien coincide la posición - cara o cruz. Simula, mediante un programa, la generación de 6 monedas aleatorias siguiendo la pauta correcta. Cada moneda generada debe ser una instancia de la clase `Moneda` y la secuencia se debe ir almacenando en una lista.

Ejemplo:

```
2 céntimos - cara
2 céntimos - cruz
50 céntimos - cruz
1 euro - cruz
1 euro - cara
10 céntimos - cara
```



Ejercicio 8

Realiza un programa que escoja al azar 10 cartas de la baraja española (10 objetos de la clase `Carta`). Emplea un objeto de la clase `ArrayList` para almacenarlas y asegúrate de que no se repite ninguna.



Ejercicio 9

Modifica el programa anterior de tal forma que las cartas se muestren ordenadas. Primero se ordenarán por palo: bastos, copas, espadas, oros. Cuando coincida el palo, se ordenará por número: as, 2, 3, 4, 5, 6, 7, sota, caballo, rey.



Ejercicio 10

Crea un mini-diccionario español-inglés que contenga, al menos, 20 palabras (con su correspondiente traducción). Utiliza un objeto de la clase `HashMap` para almacenar las parejas de palabras. El programa pedirá una palabra en español y dará la correspondiente traducción en inglés.



Ejercicio 11

Realiza un programa que escoja al azar 5 palabras en español del mini-diccionario del ejercicio anterior. El programa irá pidiendo que el usuario teclee la traducción al inglés de cada una de las palabras y comprobará si son correctas. Al final, el programa deberá mostrar cuántas respuestas son válidas y cuántas erróneas.



Ejercicio 12

Escribe un programa que genere una secuencia de 5 cartas de la baraja española y que sume los puntos según el juego de la brisca. El valor de las cartas se debe guardar en una estructura `HashMap` que debe contener parejas (figura, valor), por ejemplo ("caballo", 3). La secuencia de cartas debe ser una estructura de la clase `ArrayList` que contiene objetos de la clase `Carta`. El valor de las cartas es el siguiente: as → 11, tres → 10, sota → 2, caballo → 3, rey → 4; el resto de cartas no vale nada.

Ejemplo:

as de oros

cinco de bastos

caballo de espadas

sota de copas

tres de oros

Tienes 26 puntos



Ejercicio 13

Modifica el programa **Gestisimal** realizado anteriormente añadiendo las siguientes mejoras:

- Utiliza una lista en lugar de un *array* para el almacenamiento de los datos.
- Comprueba la existencia del código en el alta, la baja y la modificación de artículos para evitar errores.
- Cambia la opción "Salida de stock" por "Venta". Esta nueva opción permitirá hacer una venta de varios artículos y emitir la factura correspondiente. Se debe preguntar por los códigos y las cantidades de cada artículo que se quiere comprar. Aplica un 21% de IVA.



Ejercicio 14

Un supermercado de productos ecológicos nos ha pedido hacer un programa para vender su mercancía. En esta primera versión del programa se tendrán en cuenta los productos que se indican en la tabla junto con su precio. Los productos se venden en bote, brick, etc. Cuando se realiza la compra, hay que indicar el producto y el número de unidades que se compran, por ejemplo **“guisantes”** si se quiere comprar un bote de guisantes y la cantidad, por ejemplo **“3”** si se quieren comprar 3 botes. La compra se termina con la palabra **“fin”**. Suponemos que el usuario no va a intentar comprar un producto que no existe. Utiliza un diccionario para almacenar los nombres y precios de los productos y una o varias listas para almacenar la compra que realiza el usuario.

A continuación se muestra una tabla con los productos disponibles y sus respectivos precios:

avena	garbanzos	tomate	jengibre	quinoa	guisantes
2,21	2,39	1,59	3,13	4,50	1,60

Ejemplo:

```
Producto: tomate
Cantidad: 1
Producto: quinoa
Cantidad: 2
Producto: avena
Cantidad: 1
Producto: tomate
Cantidad: 2
Producto: fin
```

```
Producto Precio Cantidad Subtotal
-----
tomate    1,59      1        1,59
quinoa    4,50      2        9,00
avena     2,21      1        2,21
tomate    1,59      2        3,18
-----
TOTAL: 15,98
```



Ejercicio 15

Realiza una nueva versión del ejercicio anterior con las siguientes mejoras: Si algún producto se repite en diferentes líneas, se deben agrupar en una sola. Por ejemplo, si se pide primero 1 bote de tomate y luego 3 botes de tomate, en el extracto se debe mostrar que se han pedido 4 botes de tomate. Después de teclear “fin”, el programa pide un código de descuento. Si el usuario introduce el código “ECODTO”, se aplica un 10% de descuento en la compra.

Ejemplo:

```
Producto: tomate
Cantidad: 1
Producto: quinoa
Cantidad: 2
Producto: avena
Cantidad: 1
Producto: quinoa
Cantidad: 2
Producto: tomate
Cantidad: 2
Producto: fin
Introduzca código de descuento (INTRO si no tiene ninguno): ECODTO
```

```
Producto Precio Cantidad Subtotal
```

```
-----
tomate    1,59      3      4,77
quinoa    4,50      4     18,00
avena     2,21      1      2,21
-----
```

```
Descuento: 2,50
-----
```

```
TOTAL: 22,48
```



Ejercicio 16

Realiza un programa que sepa decir la capital de un país (en caso de conocer la respuesta) y que, además, sea capaz de aprender nuevas capitales. En principio, el programa solo conoce las capitales de España, Portugal y Francia. Estos datos deberán estar almacenados en un diccionario. Los datos sobre capitales que vaya aprendiendo el programa se deben almacenar en el mismo diccionario. El usuario sale del programa escribiendo la palabra "salir".

Ejemplo:

```
Escribe el nombre de un país y te diré su capital: España
La capital de España es Madrid
Escribe el nombre de un país y te diré su capital: Alemania
No conozco la respuesta ¿cuál es la capital de Alemania?: Berlín
Gracias por enseñarme nuevas capitales
Escribe el nombre de un país y te diré su capital: Portugal
La capital de Portugal es Lisboa
Escribe el nombre de un país y te diré su capital: Alemania
La capital de Alemania es Berlín
Escribe el nombre de un país y te diré su capital: Francia
La capital de Francia es París
Escribe el nombre de un país y te diré su capital: salir
```



Ejercicio 17

Una empresa de venta por internet de productos electrónicos nos ha encargado implementar un carrito de la compra. Crea la clase **Carrito**. Al carrito se le pueden ir agregando elementos que se guardarán en una lista, por tanto, deberás crear la clase **Elemento**. Cada elemento del carrito deberá contener el nombre del **producto**, su **precio** y la **cantidad** (número de unidades de dicho producto). A continuación se muestra tanto el contenido del programa principal como la salida que debe mostrar el programa. Los métodos a implementar se pueden deducir del main.

Contenido del main:

```
Carrito miCarrito = new Carrito();
miCarrito.agrega(new Elemento("Tarjeta SD 64Gb", 19.95, 2));
miCarrito.agrega(new Elemento("Canon EOS 2000D", 449, 1));
System.out.println(miCarrito);
System.out.print("Hay " + miCarrito.numeroDeElementos());
System.out.println(" productos en la cesta.");
System.out.println("El total asciende a "
    + String.format("%.2f", miCarrito.importeTotal()) + " euros");
```

```

System.out.println("\nContinúa la compra...\n");
miCarrito.agrega(new Elemento("Samsung Galaxy Tab", 199, 3));
miCarrito.agrega(new Elemento("Tarjeta SD 64Gb", 19.95, 1));
System.out.println(miCarrito);
System.out.print("Ahora hay " + miCarrito.numeroDeElementos());
System.out.println(" productos en la cesta.");
System.out.println("El total asciende a "
    + String.format("%.2f", miCarrito.importeTotal()) + " euros");

```

Salida:

```

Contenido del carrito
=====
Tarjeta SD 64Gb PVP: 19,95 Unidades: 2 Subtotal: 39,90
Canon EOS 2000D PVP: 449,00 Unidades: 1 Subtotal: 449,00

Hay 2 productos en la cesta.
El total asciende a 488,90 euros

Continúa la compra...

```

```

Contenido del carrito
=====
Tarjeta SD 64Gb PVP: 19,95 Unidades: 2 Subtotal: 39,90
Canon EOS 2000D PVP: 449,00 Unidades: 1 Subtotal: 449,00
Samsung Galaxy Tab PVP: 199,00 Unidades: 3 Subtotal: 597,00
Tarjeta SD 64Gb PVP: 19,95 Unidades: 1 Subtotal: 19,95

Ahora hay 4 productos en la cesta.
El total asciende a 1105,85 euros

```



Ejercicio 18

Mejora el programa anterior (en otro proyecto diferente) de tal forma que al intentar agregar un elemento al carrito, se compruebe si ya existe el producto y, en tal caso, se incremente el número de unidades sin añadir un nuevo elemento. Observa que en el programa anterior, se repetía el producto “Tarjeta SD 64Gb” dos veces en el carrito. En esta nueva versión ya no sucede esto, si no que se incrementa el número de unidades del producto que se agrega.

El contenido del main es idéntico al ejercicio anterior.

Salida:

Contenido del carrito

=====

Tarjeta SD 64Gb PVP: 19,95 Unidades: 2 Subtotal: 39,90

Canon EOS 2000D PVP: 449,00 Unidades: 1 Subtotal: 449,00

Hay 2 productos en la cesta.

El total asciende a 488,90 euros

Continúa la compra...

Contenido del carrito

=====

Tarjeta SD 64Gb PVP: 19,95 Unidades: 3 Subtotal: 59,85

Canon EOS 2000D PVP: 449,00 Unidades: 1 Subtotal: 449,00

Samsung Galaxy Tab PVP: 199,00 Unidades: 3 Subtotal: 597,00

Ahora hay 3 productos en la cesta.

El total asciende a 1105,85 euros



Ejercicio 19

Realiza un buscador de sinónimos. Utiliza el diccionario español-inglés que se proporciona a continuación. El programa preguntará una palabra y dará una lista de sinónimos (palabras que tienen el mismo significado). Por ejemplo, si se introduce la palabra “caliente”, el programa dará como resultado: ardiente, candente, abrasador. ¿Cómo sabe el programa cuáles son los sinónimos de “caliente”? Muy fácil, en el diccionario debe existir la entrada (“caliente”, “hot”), por tanto solo tendrá que buscar las palabras en español que también signifiquen “hot”; esta información estará en las entradas (“ardiente”, “hot”) y (“abrasador”, “hot”). Cuando una palabra existe en el diccionario pero no tiene sinónimos, debe mostrar el mensaje “No conozco sinónimos de esa palabra”. Si una palabra no está en el diccionario se mostrará el mensaje “No conozco esa palabra”. El usuario sale del programa escribiendo la palabra “salir”.

Español	caliente	rojo	ardiente	verde	agujetas	abrasador	hierro	grande
Inglés	hot	red	hot	green	stiff	hot	iron	big

Ejemplo:

Introduzca una palabra y le daré los sinónimos: caliente

Sinónimos de caliente: ardiente, abrasador

Introduzca una palabra y le daré los sinónimos: rojo

No conozco sinónimos de esa palabra

Introduzca una palabra y le daré los sinónimos: blanco


```
No conozco esa palabra
Introduzca una palabra y le daré los sinónimos: grande
No conozco sinónimos de esa palabra
Introduzca una palabra y le daré los sinónimos: salir
```



Ejercicio 20

Amplía el programa anterior de tal forma que sea capaz de aprender palabras y sinónimos. Cuando una palabra no tiene sinónimos, es decir, cuando aparece la palabra en español con su traducción y esa traducción no la tiene ninguna otra palabra española, se le preguntará al usuario si quiere añadir uno (un sinónimo) y, en caso afirmativo, se pedirá la palabra y se añadirá al diccionario. Se puede dar la circunstancia de que el usuario introduzca una palabra en español que no está en el diccionario; en tal caso, se mostrará el consiguiente mensaje y se dará la posibilidad al usuario de añadir la entrada correspondiente en el diccionario pidiendo, claro está, la palabra en inglés.

Ejemplo:

```
Introduzca una palabra y le daré los sinónimos: caliente
Sinónimos de caliente: ardiente, abrasador
Introduzca una palabra y le daré los sinónimos: rojo
No conozco sinónimos de esa palabra ¿quiere añadir alguno? (s/n): s
Introduzca un sinónimo de rojo: colorado
Gracias por enseñarme nuevos sinónimos.
Introduzca una palabra y le daré los sinónimos: blanco
No conozco esa palabra ¿quiere añadirla al diccionario? (s/n): s
Introduzca la traducción de blanco en inglés: white
Introduzca una palabra y le daré los sinónimos: rojo
Sinónimos de rojo: colorado
Introduzca una palabra y le daré los sinónimos: blanco
No conozco sinónimos de esa palabra ¿quiere añadir alguno? (s/n): s
Introduzca un sinónimo de blanco: albino
Gracias por enseñarme nuevos sinónimos.
Introduzca una palabra y le daré los sinónimos: blanco
Sinónimos de blanco: albino
Introduzca una palabra y le daré los sinónimos: salir
```



Ejercicio 21

La asociación “Amigos de los anfibios” nos ha encargado una aplicación educativa sobre estos animalitos. Crea un programa que pida al usuario el tipo de anfibio y que, a continuación, nos muestre su hábitat y su alimentación. Si el tipo de anfibio introducido no existe, se debe mostrar el mensaje “Ese tipo de anfibio no existe”.

Ejemplo 1:

Introduzca el tipo de anfibio: salamandra

Hábitat: Ecosistemas húmedos.

Alimentación: Pequeños crustáceos e insectos.

Ejemplo 2:

Introduzca el tipo de anfibio: gato

Ese tipo de anfibio no existe.

La información se debe guardar en dos diccionarios (dos HashMap). Uno de ellos tendrá parejas clave-valor del tipo **(tipo de anfibio, hábitat)** y otro **(tipo de anfibio, alimentación)**. A continuación se muestra la información en una tabla:

				
Tipo de anfibio	rana	salamandra	sapo	tritón
Hábitat	En los trópicos y cerca de las zonas húmedas y acuáticas	Ecosistemas húmedos.	En cualquier sitio salvo el desierto y la Antártida.	América y África.
Alimentación	Larvas e insectos.	Pequeños crustáceos e insectos.	Insectos, lombrices y pequeños roedores.	Insectos

**Ejercicio 22**

Amplía el [ejercicio 13 del capítulo anterior](#) que implementaba cuentas corrientes de un banco de tal forma que cada cuenta lleve un registro de todos los movimientos realizados: ingresos, cargos y transferencias (tanto enviadas como recibidas).

Contenido del main

```
CuentaCorriente cuenta1 = new CuentaCorriente();
CuentaCorriente cuenta2 = new CuentaCorriente(1500);
CuentaCorriente cuenta3 = new CuentaCorriente(6000);
cuenta1.ingreso(2000);
cuenta1.cargo(600);
cuenta3.ingreso(75);
cuenta1.cargo(55);
cuenta2.transferencia(cuenta1, 100);
cuenta1.transferencia(cuenta3, 250);
cuenta3.transferencia(cuenta1, 22);
cuenta1.movimientos();
```

Salida

Movimientos de la cuenta 1654432813

Ingreso de 2000 € Saldo: 2000,00 €

Cargo de 600 € Saldo: 1400,00 €

Cargo de 55 € Saldo: 1345,00 €

Transf. recibida de 100 € de la cuenta 1654432813 Saldo 1445,00 €

Transf. emitida de 250 € a la cuenta 6546817008 Saldo 1195,00 €

Transf. recibida de 22 € de la cuenta 1654432813 Saldo 1217,00 €



Ejercicio 23

En ajedrez, el valor de las piezas se mide en peones. Una **dama** vale 9 peones, una **torre** 5 peones, un **alfil** 3, un **caballo** 2 y un **peón** vale, lógicamente, 1 peón. Realiza un programa que genere al azar las capturas que ha hecho un jugador durante una partida. El número de capturas será un valor aleatorio entre 0 y 15. Hay que tener en cuenta que cada jugador tiene la posibilidad de capturar algunas de las siguientes piezas (no más): 1 dama, 2 torres, 2 alfiles, 2 caballos y 8 peones. Al final debe aparecer la puntuación total.

Ejemplo:

Fichas capturadas por el jugador:

Alfil (3 peones)

Caballo (2 peones)

Peón (1 peones)

Torre (5 peones)

Peón (1 peones)

Puntos totales: 12 peones.