Ejercicios Colecciones Kotlin

Descargar estos ejercicios

Ejercicio 1

Teniendo en cuenta las colecciones vistas en el tema, vamos a proponer una serie de casos y deberás asociar la colección más adecuada. Además, para cada caso, deberás de crear la variable inicializada a algún valor de ejemplo. Crea la variable con **var o val** dependiendo que su mutabilidad.

1. Colección para almacenar todas las provincias de España, que luego será utilizada en un desplegable para seleccionar una en concreto.
2. Colección para almacenar el sueldo de un empleado para cada uno de los 12 meses del año, este empleado siempre tendrá un total de 12 pagas, pero la paga de cada més se puede cambiar en cualquier momento.
3. Colección para almacenar los cinco nombres, de los grupos musicales que actúan semanalmente en una sala de conciertos, los grupos pueden cambiar cada semana pero siempre son cinco.
4. Colección para almacenar los artículos del carrito de la compra, el cliente puede cambiar de opinión con respecto a alguno de los artículos que ya están en el carrito. Crea un **data class Articulos** sencilla [nombre, cantidad]
5. Colección que almacena los meses del año.
6. Colección para guardar el par login/password correspondiente a los usuarios que se van apuntando a una asociación de baile, los usuarios pueden cambiar el password en cualquier momento, el login será fijo.
7. Colección para guardar las series de la aplicación de administración de la anterior entrega (utiliza la data class anterior).
8. Colección para guardar la relación entre los meses del año y los días de cada més.
9. Colección para almacenar el menú de la cena asociado a cada dia de la semana, en un hospital. Los datos (dia, menu como string) lo guardaremos en un tipo *Pair*.
10. Colección para guardar las palabras clave de un lenguaje de programación, en una versión determinada.

Ejercicio 2

Vamos a crear una de las parte de una aplicación, la que se encarga de la gestión de las contraseñas. Para ello crearemos un tipo **object GestionClaves** (para conseguir la funcionalidad de una clase estática), el tipo tendrá un campo privado de tipo **MutableMap contraseña** con sus dos partes a String (una para el login y la otra para el password), un tipo exception **ClavesException** para los casos inválidos y las funciones necesarias que te permitan añadir, modificar y eliminar el login y password de los usuarios.

Antes de salir de programa se mostrará el contenido del MutableMap, para comprobar que los datos se han gestionado correctamente.

El programa principal podría mostrar un menú como el siguiente:

```
1. Añade login
2. Modifica password
3. Elimina login
4. Salir
```

> **Aviso**
>
> Al añadir y eliminar se deberá controlar que el login no sea vacío, al modificar se deberá controlar que el login existe y se pedirá el password antiguo para comprobar que coincide y permitir el cambio.

Ejercicio 3

Vamos a seguir desarrollando el ejercicio de la gestión de series de la entrega anterior. Para ello deberemos de usar la colección más adecuada posible, para almacenar las series que un supuesto espectador esta visualizando, ya ha visualizado o visualizará en un futuro.
A partir del código anterior, procederás a dar la funcionalidad necesaria para sustituir las salidas por pantalla básicas, que teníamos hasta ahora, por la gestión correcta de las series en la colección elegida.
Debes usar las funciones de búsqueda, selección, etc. del paquete **kotlin.collections** para

conseguir un código más corto y claro.

Puedes usar los siguientes datos para probar el programa ListaSeries