# Fundamentos de Programación (ST0242)

## Taller número 10

*En el taller 10, usted resolverá varios problemas de Arreglos en Codingbat. Su cuenta de estudiante será* <codigoDeEstudiante>s20182g037t10@hotmail.com (no es necesario que la cuenta exista)*. Debe agregar al profesor para que se pueda calificar el taller. La cuenta del profesor es:* [edwin.duque@outlook.com](mailto:edwin.duque@outlook.com)*. Para los puntos que se resuelven en Java, usted debe incluir todos los archivos del proyecto en un archivo .zip. El último punto se consigna en un archivo pdf. Entregar por interactivaVirtual.*

1. **Array-3**:

* maxSpan
* fix34
* canBalance
* squareUp
* maxMirror

1. Cree una clase Persona con los siguientes atributos: nombre, género, edad, peso. Escriba otra clase llamada ColeccionPersonas con los siguientes métodos:
   1. popularPersonas(): Crea instancias de personas de manera aleatoria y los agrega a un ArrayList de referencias a Personas.
   2. ordenar(): ordena las referencias de manera ascendente por edad, si hay empate por edad, se ordenan por peso.
   3. Imprimir(): imprime el ArrayList.

Para probar los métodos, cree un método main() que invoque primero el método popularPersonas(), luego imprima la colección, luego la ordene y luego la imprima ordenada.

1. La editorial “No lea todo lo que cree” quiere guardar información referente a los libros que imprime. Para cada libro se guarda el nombre del autor, el nombre del libro y el número de páginas que éste tiene. Para esto se utiliza la clase Libro. También se tiene una clase ColeccionLibros, con referencias a los libros. En la clase ColeccionLibros se implementa la siguiente funcionalidad:
   1. popularLibros(): recibe la información de un libro y lo agrega a la colección. Asuma que el usuario no introduce libros repetidos.
   2. escritorMasProlijo(): devuelve el nombre del autor que más páginas ha escrito.
   3. escritorConMasArticulos(): devuelve el nombre del autor que más libros ha escrito.
   4. tituloMasLargo(): retorna el nombre del libro más largo, así como su autor.

Para probar lo anterior, cree un método main() en la clase ColeccionLibros.

1. Cree una clase Carro, que tendrá la placa de un carro y su dueño. Ahora cree una clase ColeccionCarros (para almacenar el inventario de carros) que tendrá un ArrayList de referencias objetos del tipo Carri. Dentro de la clase ColeccionCarros, usted va a implementar la funcionalidad CRUD (Create, Read, Update, Delete) de la siguiente manera:
   1. Cree un método crear(), que reciba una referencia a un carro nuevo, y si no está en el inventario la agregue al mismo.
   2. Cree un método leer(), que reciba la placa de un carro. Si el carro está en el inventario, retorna una referencia al mismo. De lo contrario, retorna null.
   3. Cree un método actualizar(), que recibe una referencia de un carro actualizado. Si el carro existe, se actualiza el nombre del dueño. Si el carro no existe, no se hace nada.
   4. Cree un método borrar(), que reciba la placa de un carro y lo retire del inventario. Si el carro no estaba en el inventario, no se hace nada.

Pruebe la funcionalidad anterior desde un método main().

1. Escriba un método que reciba un arreglo de caracteres (por ejemplo {‘a’, ‘b’, ‘c’}) y escriba todas las palabras de 2 letras que se pueden escribir con esas letras. En el ejemplo: “aa”, “ab”, “ac”, “ba”, bb”, “bc”, “ca”, “cb”, “cc”. Una forma sencilla es con un ciclo dentro de otro ciclo.
2. Escriba un método que reciba un arreglo de caracteres (por ejemplo {‘a’, ‘b’, ‘c’}) y escriba todas las palabras de 3 letras que se pueden escribir con esas letras.
3. Escriba un programa Java que cree un ArrayList de Objetos de tipo Moto. El programa pide por teclado los datos de las motos y los guarda en el ArrayList. Escriba los siguientes métodos:
   1. listarTodas(): Muestra todas las motos que se han introducido.
   2. listarMarca(): Muestra las motos de una marca que se recibe como parámetro
   3. listarPorKilometros(): Muestra las motos que tienen menos de un número de kilómetros que se recibe como parámetro
   4. listarMasTrajinada(): Muestra la moto con mayor número de kilómetros.
   5. listarOrdenadas(): Muestra las motos ordenadas de manera ascendente por número de kilómetros.
4. Desarrolle un programa que lea una serie de valores numéricos enteros desde el teclado y los guarde en un ArrayList de tipo Integer. La lectura de números termina cuando se introduzca el valor -99. Este valor no se guarda en el ArrayList. A continuación, el programa mostrará por pantalla el número de valores que se han leído, su suma, su media y su moda. Por último, se mostrarán todos los valores leídos, indicando cuántos de ellos son mayores que la media.
5. Consulte en qué consiste un arreglo bidimensional (o matriz) y cómo se maneja en Java. Escriba un programa sencillo que utilice una matriz y explique lo que hace.