CURSO DE PROGRAMACIÓN FULL STACK

COLECCIONES

PARADIGMA ORIENTADO A OBJETOS

GUÍA DE COLECCIONES

COLECCIONES

Una colección representa un grupo de objetos. Esto objetos son conocidos como elementos. Cuando queremos trabajar con un conjunto de elementos, necesitamos un almacén donde poder guardarlos. En las colecciones se puede almacenar cualquier tipo de objeto y se puede utilizar una serie de métodos comunes, como por ejemplo: agregar, eliminar, obtener el tamaño de la colección. Las colecciones son una especie de arrays de tamaño dinámico. Java Collections Framework es el marco de trabajo que permite trabajar con objetos contenedores de objetos tales como listas, conjuntos y mapas.

Las operaciones básicas de una collection son:

- add(T): Añade un elemento.
- iterator(): Obtiene un "iterador" que permite recorrer la colección visitando cada elemento.
- size(): Obtiene la cantidad de elementos que esta colección almacena.
- contains(t): Pregunta si el elemento t ya está dentro de la colección.

LISTAS

Las listas modelan una colección de objetos ordenados por posición. La principal diferencia entre las listas y los arreglos tradicionales, es que la lista crece de manera dinámica a medida que se van agregando objetos. No necesitamos saber de antemano la cantidad de elementos con la que vamos a trabajar. El framework trae varias implementaciones de distintos tipos de listas tales como ArrayList, LinkedList y Vector.

- ArrayList: se implementa como un arreglo de tamaño variable. A medida que se agregan más elementos, su tamaño aumenta dinámicamente.
- LinkedList: se implementa como una lista de doble enlace. Su rendimiento al agregar y quitar es mejor que Arraylist, pero peor en los métodos set y get.
- Vector: es similar a un ArrayList, pero está sincronizado.

CONJUNTOS

Los conjuntos modelan una colección de objetos de una misma clase donde cada elemento aparece sólo una vez, a diferencia de una lista donde los elementos podían repetirse. El framework trae varias implementaciones de distintos tipos de conjuntos:

- HashSet, se implementa utilizando una tabla hash. Los elementos no están ordenados. Los métodos de agregar, eliminar y contiene tienen una complejidad de tiempo constante por lo que tienen mejor performance que el Treeset.
- TreeSet se implementa utilizando una estructura de árbol (árbol rojo-negro en el libro de algoritmos). Los elementos de un conjunto están ordenados, pero los métodos de agregar, eliminar y contiene tienen una complejidad peor a la de HashSet. Ofrece varios métodos para tratar con el conjunto ordenado como primero(), último(), headSet(), tailSet(), etc.

- LinkedHashSet está entre HashSet y TreeSet. Se implementa como una tabla hash con una lista vinculada que se ejecuta a través de ella, por lo que proporciona el orden de inserción.

MAPAS

Los mapas son una de las estructuras de datos importantes del Framework de Collections. Las implementaciones de mapas son HashMap, TreeMAp, LinkedHashMap y HashTable.

- HashMap es un mapa implementado a través de una tabla hash, las llaves se almacenan utilizando un algoritmo de hash.
- TreeMap es un mapa implementado a través de un árbol, es decir que las llaves se almacenan en una estructura de árbol, y por lo tanto, se almacenan ordenadas.
- LinkedHashMap es un HashMap que conserva el orden de inserción.
- Hashtable está sincronizado, en contraste con HashMap. Tiene una sobrecarga para la sincronización.

PREGUNTAS DE APRENDIZAJE

1) Cual de estos paquetes es el contenedor de las colecciones:
   a) java.lang
   b) java.util
   c) java.net
   d) java.awt

2) Las colecciones en Java son:
   a) Un grupo de objetos.
   b) Un grupo de clases.
   c) Un grupo de interfaces.
   d) Ninguna de las anteriores.

3) En el framework de colecciones de Java un Set es
   a) Una colección que no puede contener elementos duplicados
   b) Una colección ordenada que puede contener elementos duplicados
   c) Un objeto que mapea conjuntos de clave valor y no puede contener valores duplicados
   d) Ninguna de las anteriores

4) Cual de estos métodos, no sirve para todos los elementos de una colección de manera aleatoria:
   a) .rand();
   b) .shuffle();
   c) .randomize;
   d) .ambiguous();

5) Cual de estos métodos, borra todos los elementos de una colección:
   a) .clear();
   b) .delete();
   c) .remove();
   d) .reset();

6) Para conocer el numero de elementos en una lista se usa la función:
   a) lista.lenght();
   b) lista.size();
   c) lista.contains();
   d) lista.iterator();

7) En Java un Iterator es:
   a) Una interface que proporciona los métodos para borrar elementos de una colección
   b) Una interface que proporciona los métodos para recorrer los elementos de una colección y posibilita el borrado de elementos
   c) Una interface que proporciona los métodos para ordenar los elementos de la colección.
   d) Ninguna de las anteriores

EJERCICIOS DE APRENDIZAJE

En este módulo vamos a continuar modelando los objetos del mundo real con el lenguaje de programación Java, pero ahora vamos a utilizar las colecciones para poder manejarlas de manera más sencilla y ordenada.

VER VIDEO: Lista

1. Crear una clase llamada Palabra que mantenga información sobre diferentes palabras con un atributo ArrayList de tipo String, que se le van a ir agregando palabras por medio del método add(String). Al final, el programa debe permitirnos conocer el conjunto de palabras de una determinada longitud ingresada por el usuario. Este conjunto deberá estar ordenado alfabéticamente. Crear una aplicación que muestre toda la información que disponga la clase Palabra.

   NOTA: Si necesitas saber más sobre ordenamiento en java consulta los siguientes links. Algoritmos de ordenamiento. / Implementar Comparator

2. Diseñar 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 y su media (promedio). Por último se mostrarán todos los valores leídos, indicando cuántos de ellos son mayores que la media. Utilizar los siguientes 3 métodos para resolver el ejercicio:

   - Método leerValores(): pide por teclado los números y los almacena en el ArrayList. La lectura finaliza cuando se introduce el valor -99. El método devuelve mediante return el ArrayList con los valores introducidos.

   - Método calcularSuma(): Recibe como parámetro el ArrayList con los valores numéricos y calcula y devuelve su suma. En este método se utiliza un Iterator para recorrer el ArrayList.

   - Método mostrarResultados(): Recibe como parámetro el ArrayList, la suma y la media aritmética. Este método muestra por pantalla todos los valores, su suma y su media y calcula y muestra cuantos números son superiores a la media.

3. Crear una clase llamada CantanteFamoso, esta clase guardará cantantes famosos y tendrá como atributos el nombre y discoConMasVentas, y los métodos getters y setters.

   Se debe además crear una clase Test con el método main que inicialice un arrayList listaCantantesFamosos y agregue manualmente 5 objetos de tipo CantanteFamoso a la la lista. Luego, se debe mostrar los nombres de cada cantante y su disco con más ventas por pantalla.

Además, se debe pedir al usuario un nombre y disco con más ventas de otro cantante famoso, y una vez introducidos los datos mostrar la lista actualizada. Una vez mostrada la lista actualizada, se debe dar opción a elegir entre volver a introducir los datos de otro cantante, editar un cantante, eliminar un cantante o salir del programa. Se podrán introducir tantos datos de cantantes como se desee.

VER VIDEO: Conjunto

4. Se necesita implementar un sistema en el que se puedan cargar alumnos, a los cuales los caracterizan el nombre y apellido, el número de legajo, el sexo, condición (regular o condicional) y la nota final. Estos alumnos se deben cargar en una asignatura, llamada Curso Programación Egg. Implemente las clases y métodos necesarios para esta situación, teniendo en cuenta lo que se pide a continuación:

- Mostrar en pantalla todos los alumnos que se encuentren en la asignatura.
- Mostrar en pantalla los alumnos que se encuentren como condicional y su cantidad.
- Ordenar los alumnos de acuerdo a su nota (de mayor a menor) y mostrarlo en pantalla.
- Ordenar los alumnos de acuerdo a su nota (de menor a mayor) y mostrarlo en pantalla.
- Ordenar los alumnos por nombre y apellido y mostrarlo en pantalla

Nota: para los ordenamientos utilizar las facilidades provistas por la plataforma Java.

5. Implemente un programa para una Librería haciendo uso de conjuntos para evitar datos repetidos. Para ello, se debe crear una clase llamada Libro que guarde la información de cada uno de los libros de una Biblioteca. La clase Libro debe guardar el título del libro, autor, número de ejemplares del libro y número de ejemplares prestados.

La clase Librería contendrá además los siguientes métodos:

- Constructor por defecto.
- Constructor con parámetros.
- Métodos Setters/getters
- Método préstamo que incrementa el atributo correspondiente cada vez que se realice un préstamo del libro. No se podrán prestar libros de los que no queden ejemplares disponibles para prestar. Devuelve true si se ha podido realizar la operación y false en caso contrario.

- Método devolución que decremente el atributo correspondiente cuando se produzca la devolución de un libro. No se podrán devolver libros que no se hayan prestado. Devuelve true si se ha podido realizar la operación y false en caso contrario.

- Método toString para mostrar los datos de los libros.

VER VIDEO: Mapa

6. Se necesita una aplicación para una tienda en la cual queremos almacenar los distintos productos que venderemos y el precio que tendrán. Además, se necesita que la aplicación cuente con las funciones básicas, como introducir un elemento, modificar su precio, eliminar un producto y mostrar los productos que tenemos con su precio (Utilizar Hashmap).

7. Almacena en un HashMap los códigos postales de 10 ciudades a elección de esta página: https://mapanet.eu/index.htm. Nota: Poner el código postal sin la letra, solo el numero.

- Pedirle al usuario que ingrese 10 códigos postales y sus ciudades.
- Muestra por pantalla los datos introducidos
- Pide un código postal y muestra la ciudad asociada si existe sino avisa al usuario.
- Después editar el código postal que buscó el usuario.
- Muestra por pantalla los datos
- Agregar una ciudad con su código postal correspondiente más al HashMap.
- Elimina 3 ciudades existentes dentro del HashMap, que pida el usuario.
- Muestra por pantalla los datos