**2014 Beginning Java 8 Language Features**

**Chapter 12: Collections**

**¿Qué es una colección?**

* Una colección es un objeto que contiene un grupo de objetos. Una colección también es conocida como contenedor. Cada objeto en una colección es llamado *elemento* de la colección.
* El concepto de colección en java no es diferente del concepto de colección que usamos en nuestra vida diaria. Observamos diferentes tipos de colecciones cada día. Cada colección contiene un grupo de objetos. ¿Qué diferencia un tipo de colección de otro tipo de colección? Un tipo de colección se distingue de otro tipo de colección basado en la manera en que administran o manejan sus elementos. Tomemos unos ejemplos de colecciones de nuestra vida diaria.
* Un frasco de dinero es un ejemplo de una colección. Contiene un grupo de monedas. ¿Se pone una moneda en el frasco en un orden específico? ¿recuperamos/obtenemos/sacamos las monedas del frasco en un orden específico? ¿podemos poner muchas monedas del mismo tipo en el frasco? ¿podemos remover/eliminar todas las monedas del frasco de na sola vez o debemos sacarlas una por una? ¿nuestro teclado es una colección? ¿no es una colección de teclas? ¿un teclado tiene teclas repetidas/duplicadas? No, no podemos tener teclas duplicadas en un teclado. Sin embargo, podemos tener monedas duplicadas en nuestro frasco de monedas.
* Consideremos una cola de clientes en un mostrador en una oficina de correos. La cola de los clientes es una colección. La cola sigue unas reglas específicas, el primero en entrar, es el primero en salir o ser servido. (FIFO: First In, First Out – Primero en Entrar, Primero en Salir).
* Consideremos una pila de libros en el escritorio. También es una colección. Asumiendo que manejamos un libro a la vez, la regla a seguir es que el último libro que fue puesto en la pila tiene que quitarse/removerse primero. Esta regla es lo opuesto a la regla de la colección de clientes en una cola de un mostrador en la oficina de correos. En este caso, la pila de libros sigue la regla de Primero en Entrar, Primero en Salir. (FIFO: First In, First Out).
* Se mencionaron unos ejemplos de colecciones que siguen diferentes reglas para manejar/administrar sus elementos. ¿Qué harías si tuvieras que modelar estas colecciones de objetos en un programa de java? Primero tendríamos que categorizar todos los posibles tipos de colecciones con los que trabajaríamos en nuestros programas. Luego, se tendrían que escribir algunas interfaces y clases genéricas reusables que tendríamos que usar en situaciones donde necesitamos lidiar con colecciones de objetos. La buena noticia es que no tenemos que escribir código genérico para manejar/administrar colecciones. Los diseñadores del lenguaje java se dieron cuenta de la necesidad de ellas e incorporaron un framework en librería java, la cual es llamada **Collection Framework.**
* El Collection Framework está formado por interfaces, clases implementadas y algunas clases de utilidad que nos permiten manejar la mayoría de los tipos de colecciones que no encontraremos en una aplicación java. Si nos encontramos con un tipo de colección para la cual java no proporciona una implementación, podemos crear nuestra propia implementación, que funcionará a la perfección con el Collection Framework.
* Existen diferentes tipos de colecciones disponibles en el Collection Framework, la siguiente figura muestra cinco tipos, una bolsa, una lista, una cola, una pila y un mapa.

|  |
| --- |
|  |

* La colección map en la figura, es una colección de pares nombre-teléfono. Mapea un nombre a un teléfono. En este punto, estas figuras no están asociadas con ningún tipo específico de colecciones de Java. Nos ayudan a visualizar que las colecciones de Java son las mismas colecciones de nuestra vida diaria. Las flechas en algunas colecciones indican la entrada y salida de un elemento hacia y desde la colección. Se puede observar también que algunas colecciones imponen que un elemento debe ser agregado de cierta manera a la colección y debe ser removido de la colección de cierta maneta. Por ejemplo, en una cola, los elementos entran desde un extremo y salen desde el otro extremo. En una pila, los elementos entran y salen del mismo extremo.

**2014 Big Java**

**COLECCIONES**

* Un **ArrayList** es una clase que contiene un arreglo que se va expandiendo según sea necesario.
* Si no te preocupa la eficiencia, puedes usar ArrayList cuando necesitemos coleccionar objetos.
* Sin embargo, varias operaciones son ineficientes con ArrayList. En particular, si un elemento es agregado o removido, los elementos en posiciones más grandes (altas) deben ser movidos.
* Una **LinkedList** permite insertar y remover elementos en medio de la lista de manera eficiente.
* **Una lista es una colección que recuerda el orden de sus elementos.**
* Las Listas (**List)** se usan cuando se quiere conservar el orden que se estableció.
* Ejemplo: en un estante de libros, se puede ordenar los libros por temas.
* Una lista es una estructura de datos apropiada para dicha colección porque el orden es importante para nosotros.
* **Un set es una colección no ordena de elementos únicos.**
* Sin embargo, en algunas aplicaciones no nos interesa el orden de los elementos en una colección.
* Por ejemplo: considere un distribuidor de libros por correo. Sin clientes buscando en los estantes, no hay necesidad de ordenar libros por tema.
* Dicha colección, que no tiene un orden intrínseco es llamada **set** (conjunto).
* Ya que un **set** no rastrea el orden de los elementos, puede organizar los elementos para que las operaciones de **búsqueda, agregar y eliminar** sean más eficientes.
* Los científicos de la computación han inventado mecanismos para este propósito. La librería de java proporciona clases basadas en dos mecanismos llamadas **hash tables** y **binary search trees**.