Proyecto 1 de Programacion Orientada a Objetos

Universidad Nacional Autónoma de México

18 de marzo de 2019

Programación Orientada a Objetos

Grupo: 1
Semestre: 2019-2
Profesor: M.C. Tista García Edgar

Integrantes:

- Velasco Vanegas Ricardo Alonso
- Félix Flores Paul Jaime
- Garcia Lazcano Carlos David

Introducción

La programación orientada a objetos es un ámbito bastante importante para el desarrollo de las habilidades de un programador. Además de otorgar nuevas clases de conocimiento, su buen manejo permite el desarrollo de programas los cuales resultarian demasiado complicados de crear en un lenguaje estructurado.

Uno de los lenguajes mas usados a la hora de decidir programar con objetos es Java. Tal lenguaje, además de ser el lenguaje de programación mas usado del mundo, tiene bastantes puntos a favor para su uso, como el hecho de que es multiplataforma (Puede ser usado en Windows, Linux o Mac), es gratuito en su totalidad, se actualiza de forma continua, tiene gran soporte y documentación disponible en la red, etc.

Una de las varias herramientas que Java nos proporciona a la hora de programar con objetos son las colecciones.

Desarrollo

Las colecciones, en resumen, son grupos de objetos (También conocidos como elementos) los cuales pueden ser ordenados de diversas maneras diferentes, añadiendo incluso la opción de interactuar con los elementos de diversas maneras.

Los 3 tipos de colecciones que vamos a ver principalmente en este documento son los siguientes:

Listas (List)

Las listas en Java son principalmente usadas cuando se tiene un conjunto de elementos los cuales tienen un orden secuencial. De forma parecida a un Array, los elementos de una lista se encuentran localizados en determinadas posiciones de la misma.

Los dos subtipos de lista que podemos usar para una integración mas especializada son ÁrrayList'y 'Vector'.

ArrayList

El ArrayList crea una lista de tamaño dinámico. Esto quiere decir que entre mas elementos se le vayan agregando, su tamaño irá aumentando de forma proporcional. Aun si se tiene una capacidad inicial, tal límite puede ser incrementado sin problema.
Uno de los problemas principales con el ArrayList es que su implementación no se encuentra sincronizada. Esto quiere decir que si múltiples hilos deciden acceder a un mismo ArrayList existe la posiblidad de que ocurran problemas de

consistencia entre los datos (Los cuales puede que sean sobreescritos o ignorados).

Vector

El Vector es una implementación parecida al ArrayList, con su diferencia principal siendo que el uso del Vector si permite la sincronización de datos. De igual manera una ventaja que tienen los vectores es que su implementación en el lenguaje se encuentra disponible incluso para las versiones mas antiguas de Java. Sin embargo, se debe tomar en cuenta que algunos métodos definidos de Vectores puede que no funcionen en estas versiones anteriores.

Conjuntos (Set)

Los conjuntos son, en resumen, grupos de elementos no repetidos. Esto significa que en caso de tener dos objetos de diferente nombre (Como podria ser 'carro1'y 'carro2') pero que comparten en absoluto todas sus características, el programa no permitiría que ambos sean ingresados en el mismo conjunto.
Los subtipos principales de los conjuntos son 'HashSet', 'LinkedHashSet'y 'TreeSet'.

HashSet / LinkedHashSet

Los elementos ingresados dentro del conjunto se manejan dentro de una tabla de dispersión, además de que las operaciones básicas que pueden ocurrir sobre los elementos son realizados en un tiempo constante mientras se itera el programa.
Esto provoca que el rendimiento se vea progresivamente ralentizado conforme la cantidad de objetos dentro del conjunto aumenta, puesto que se tiene que ir revisando uno a uno, pero permite una mayor estabilidad del programa y evita que ocurran errores (Como podria ocurrir en algunas ocasiones con los ArrayList).
El LinkedHashSet funciona de la misma manera que el HashSet anteriormente mencionado, con la diferencia de que en este subtipo la tabla de dispersión se encuentra doblemente enlazada.
Esto permite que la estabilidad se vea considerablemente aumentada y que los posibles errores que puedan ocurrir sean disminuidos todavia en mayor cantidad, pero esto a su vez hace que el programa corra con menor velocidad debido a que se tiene que tener en cuenta los enlaces en todo momento, lo cual en conjuntos de demasiados objetos reduciría la velocidad de forma considerable.

TreeSet

De forma parecida que los dos ejemplos anteriores, la diferencia principal es que el TreeSet maneja los datos con una estructura de arbol, lo cual aumenta la precisión del programa y evita con mayor efectivdad la aparición de errores

a cambio de rendimiento progresivamente mas pesado.

Mapas (Hash)

Si hablamos desde un punto de vista técnico, los mapas no son colecciones ya que no heredan de la interfaz 'Collection', mas bien heredan de 'Map'. Sin embargo, su utilidad permite que puedan ser analizados en una categoria parecida, ya que permite el acomodo de grupos de objetos de una forma parecida a las dos categorias anteriormente revisadas.
Al heredar de 'Map', se usan mapas, los cuales son formas de relacionar a los objetos con 'claves'de diferentes valores. De esta forma, se puede hacer una búsqueda ordenada entre objetos, con una mayor eficiencia en caso de querer buscar instancias de objeto con elementos en común.
Los subtipos principales de Mapas son 'HashMap', 'TreeMap'y 'Hashtable'.

HashMap

Al usar esta categoria, la tabla de dispersión creada almacena la información de forma constante (Por medio de las operaciones de get y put). En general es una categoria simple y de bajo costo de procesamiento, pero al incrementar el número de iteración de la tabla y de los elementos ingresados se corre el riesgo de que no se respete el orden de las claves.

TreeMap

Al usar TreeMap, la forma de iterar la tabla de dispersión se asemeja a la de un arbol. Esto hace que el coste de eficiencia de las operaciones comunes se convierta en logarítmico, lo cual es bastante eficiente cuando no se incluyen muchos datos, pero aumenta de forma acelarada lo cual hace a un programa demasiado lento ya cuando se insertan demasiados datos.

HashTable

Con un funcionamiento parecido al de HashMap, HashTable tiene como diferencia principal que las implementaciones se encuentran sincroinzadas, ademas de no permitir claves nulas. Es recomendado usar este metodo por encima de HashMap, ya que se encuentra mas actualizado y ofrece metodos extra, además de ser mas estable.

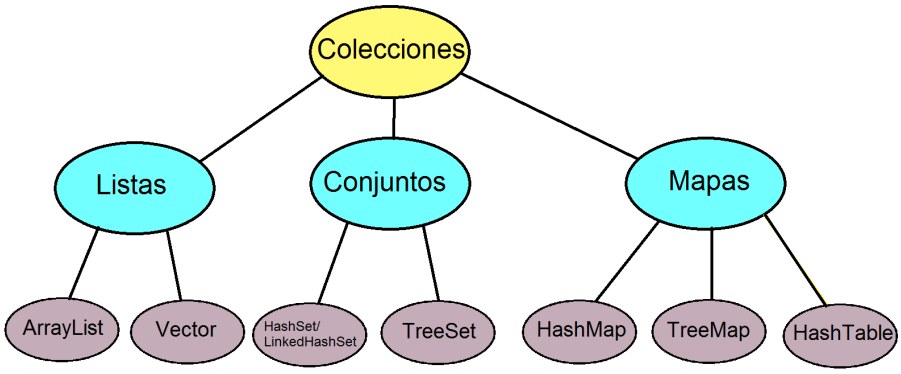

Figura 1: Jerarquia de Colecciones

¿Cuál escoger?

Para un lenguaje tan amplio y completo como lo es Java, puede resultar complicado discernir cuando debemos usar los diferentes métodos que tenemos a nuestra disposición. Para poder resolver este problema, hay que verificar desde un principio que estamos buscando hacer.

Si el problema que deseamos plantear en código consta de pares de claves y valores, entonces lo mas útil en este caso seria el uso de Mapas Hash, gracias a la facilidad que estos proporcionan cuando se le busca asignar cierta palabra clave a un atributo de algún elemento en particular.

De aca podemos discernir si el orden es importante o no. En caso de que el orden no sea importante para los pares de claves y valores entonces conviene usar HashMap, debido a que no utiliza métodos complejos e inecesarios para nuestro propósito, lo cual aceleraría el rendimiento y la implementación.

En cambio, si sí se requiere tener un orden, convendría saber si el orden es de inserción o de orden de la clave. Para el caso de inserción de datos, nuestra mejor opción seria LinkedHashMap. Para el caso de el orden de las claves (O llaves), sería la alternativa, TreeMap.

Si en cambio solo se busca implementar Valores, primero hay que verificar si existen los duplicados. En caso de que exista la probabilidad de que ocurran duplicados y no nos importe que existan, nuestra mejor opción sería ArrayList. Incluso si no existen duplicados, podemos usar ArrayList en caso de que no nos importe la búsqueda de elementos.

Si lo que ingresamos en la colección de valores nos interesa poder encontrar-

lo y buscarlo, pero el orden no nos importa realmente, una de nuestras mejores opciones es HashSet.

Si el orden de los valores ingresados es importante para nosotros, debemos primero verificar si buscamos el orden en la inserción de datos o en la ordenación de los elementos. En caso de querer orden en la inserción, usariamos LinkedHashSet. En cambio si buscamos un orden mas eficiente dentro de los mismos elementos, TreeSet es nuestra mejor opción.

Ejemplos

Anexado en esta misma carpeta del proyecto, se incluyó un programa el cual utiliza los tres métodos principales de colecciones de java anteriormente mencionados. Además de contener estos programas, se incluirá un manual de usuario el cual, además de explicar a detalle el funcionamiento de los programas y explicará en que casos estos programas no funcionan correctamente, tendrá un análisis a detalle de los métodos usados para de esta forma explicar con un ejemplo natural el funcionamiento de los métodos de las colecciones.

Conclusiones

Félix Flores Paul Jaime

En este primer proyecto los conjuntos que nos ofrece Java, así como reconocer las ventajas que nos ofrecen uno de otros programas. La implementación de los códigos fue fácil, no los repartimos los códigos para facilitar la carga de trabajo, aun así, el equipo estuvo en comunicación continua, por si algún código no nos salía. Creo que el código más complicado fue listas, ya que este llevaba una lógica un poco más avanzada, en este programa incrementamos más cosas de lo que nos pedía el proyecto, para sí tener más calificación. Al final nos sentimos contentos con el proyecto que estamos entregando. Creo que podemos mejorar los códigos teniendo un poco más de conocimientos en la materia y tratar de reducir las líneas de código, pero para una buena práctica pues está bien, me siento muy satisfecho con el trabajo que realice con mi equipo.

Velasco Vanegas Ricardo Alonso

El proyecto que fue realizado considero que fue de una dificultad compleja pero adecuada. Tanto yo como mis compañeros tuvimos que estudiar tanto entre nosotros como por nuestra cuenta las partes del proyecto que debimos realizar cada uno de forma casi autodidacta para poder entender como realizar los códigos. No solo tuvimos que estudiar cada uno por nuestra cuenta, pero cuando surgían dudas en nuestros códigos tuvimos que apoyarnos unos a los otros para encontrar posibles errores que habiamos encontrado de vez en cuando en nuestros códigos, lo cual tardó un poco de tiempo debido a un poco de falta de sincronización pero logramos terminar todo en tiempo y forma. De igual manera, el aprender a realizar un documento en LaTex fue extraño ya que en lo personal ha sido la primera vez que tengo que usar un programador de texto. Costó trabajo aprenderlo a usar mas que nada porque muchos de los tutoriales que habia en linea no eran de la versión adecuada y mostraban comandos que no funcionaban realmente, por lo que tardé algo de tiempo en encontrar de verdad información que sirviera para la versión que tenía disponible. En resumen estoy satisfecho con el resultado del proyecto y considero que tuvo una buena dificultad.

Garcia Lazcano Carlos David

Los Set en especial el TreeSet facilita el ordenamiento de los elementos puesto que el mismo conjunto lo ordena, con ayuda de su método compareTo, es decir no se necesita un algoritmo extra para ordenarlo, todo depende de que elemento se considere para esto (sí se usan objetos) son prácticos, pero sí uno no sabe como implementarlos de la manera adecuada se le hará tedioso. Otro punto a considerar es que debido al ordenamiento, y del elemento considerado, no se permiten duplicados en los Set, esto puede ser efectivo si solo se quieren hacer filtros de datos. Por todo esto es recomendable sí necesitas mantener tus datos ordenados al instante y no necesitas elementos repetidos.

Bibliografía

- Lenguaje Java Avanzado. Archivo PDF Pag. 34-40. http://www.jtech.ua.es/j2ee/publico/lja-2012-13/wholesite.pdf
- Colecciones en Java. Prof. Renny Batista, Octubre 2015. Presentacion PP. https://es.slideshare.net/rennybatista/java-colecciones
- Introducción a Colecciones en Java. Rafael Vindel Amor. https://www.adictosaltrabajo.com/2015/09/25/introduccion-a-colecciones-en-java/
- Tutorial de Latex. Múltiples páginas. https://www.latex-tutorial.com/