

Tarea investigativa de la estructura de datos SET

Básicamente la estructura Set tiene exactamente el mismo interfaz que Collection, y no hay ninguna funcionalidad extra, como en el caso de las Listas. Un Set es exactamente una Colección, pero tiene utilizada en un entorno determinado, que es ideal para el uso de la herencia o el polimorfismo. Un Set sólo permite que exista una instancia de cada objeto, es decir donde no puede haber elementos repetidos, y cuyos elementos no se almacenan necesariamente siguiendo un orden particular.

En este tipo de interfaz los elementos incorporados al Conjunto deben tener definido el método equals(), en aras de establecer comparaciones para eliminar duplicados.

La estructura set tiene diferentes implementaciones una de ellas y la más conocida es HashSet, un HashSet es una estructura de datos que contiene un conjunto de objetos. Es una implementación de Set y permite buscar un objeto dentro del conjunto de forma rápida y fácil. Internamente gestiona un array y guarda los objetos utilizando un índice calculado con un código hash del objeto.

El nombre HASH, hace referencia a una técnica de organización de archivos llamada hashing o "dispersión" en el cual se almacenan registro en una dirección del archivo que es generada por una función que se aplica sobre la llave del registro. Todos los objetos de Java pueden producir un código de tipo hash, y hashCode() es un método de la clase raíz Object. Un HashSet toma un hashCode() del objeto y lo utiliza para localizar rápidamente la clave. Esto redundante en una mejora dramática de rendimiento.

Algunas características:

- Los elementos de un HashSet no están ordenados
- Para añadir un elemento al HashSet se utiliza el método add(Object obj).
- Para borrar un elemento se utiliza remove(Object obj).
- Para borrar todos los elementos se utiliza clear().
- El tamaño del HashSet se puede obtener con la función size()

Ejemplo add set.

```
1 Set<String> names = new HashSet<>();
2 names.add("Tom");
3 names.add("Mary");
4
5 if (names.add("Peter")) {
6     System.out.println("Peter is added to the set");
7 }
8
9 if (!names.add("Tom")) {
10     System.out.println("Tom is already added to the set");
11 }
```

Output:

```
1 Peter is added to the set
2 Tom is already added to the set
```

Palabras Clave:

Collection:

La interfaz collection es la superinterfaz de donde heredan la mayoría de las interfaces utilizadas para el manejo de las colecciones. Es la interfaz raíz de la jerarquía de interfaces. La interfaz collection forma parte del Collection Framework, un conjunto de interfaces y clases que representan distintos modos de agrupar objetos, según distintas políticas de manejo de memoria o acceso a ellos. Representan un conjunto de objetos, también llamados elementos. Una clase que quiera comportarse como una Collection deberá implementar esta interfaz, por lo tanto sus métodos.

Equals:

Nos dará el valor true si son iguales y false si no. Las subclases pueden sobrescribir el método equals() para hacer una comparación entre dos objetos. En la lista de argumentos del método equals() hay que pasarle un argumento de tipo Object. sino se sobrecarga el método, no se sobreescribe.