	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Edgar Tista Garcia

Asignatura: Estructura de Datos y Algoritmos II

Grupo: 5

No de Práctica(s): 5

Integrante(s): Peralta Correa José Roberto

Semestre: 2019-1

Fecha de entrega: 23/09/2018

Observaciones:

CALIFICACIÓN: _____

Objetivo: El estudiante conocerá e identificará las características necesarias para realizar búsquedas por transformación de llaves.

Ejercicio 0 esqueleto de la práctica:

```
Bienvenido a la practica 5 elige una opcion:

1)Funcion Hash por modulo
2)Encadenamiento
3)Manejo de tablas hash en java
4)Salir

Opcion:1
```

Ejercicio 1. Funciona Hash por modulo
Cree en la clase HashModulo en donde uno de sus atributos es la lista de Integers. Para definir el tamaño de esta lista utilicé el constructor donde se le pasa como parámetro el número de elementos. Como en la hora de laboratorio tuve un error me di cuenta que hay que ingresar elementos ya que, si no no permite hacer el set, por lo que ingreso 20 veces el 0 en la lista.

Dentro de esta clase cree un método para el submenú de Hash por modulo, para esto no tuve ninguna complicación,

a) Submenú

```
Funcion Hash por Modulo elige una opcion:

1)Agregar elementos
2)Imprimir lista
3)Buscar elementos
4)Volver a menu principal
```

b) Y c) Hash modulo al agregar/buscar elementos
Aquí no tuve complicaciones, cree un método que devuelve el hash con el módulo de 17, y un método para buscar, así como para agregar, ya que al buscar hay que igualarlo al valor dado, y al agregar hay que hacer otras validaciones, como que la posición este dentro del rango y que no exista ya un valor en esa posición.

Agregar:

```
Opcion:1

Dime el numero: 3
Posicion:3
Posicion:3

El numero 3 ha sido insertado en la posicion: 3

[0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Buscar:

```
Opcion:3

Dime el numero: 4
Posicion:4

El numero 4 se encuentra en la posicion en la posicion: 4
```

Con colisiones:

Para las colisiones ya tuve mayor dificultad, investigué como sería conveniente para después de elevar el hash anterior al cuadrado poder obtener los dígitos centrales. Así que lo convertí a un String y de allí saque los caracteres y con base al número de caracteres pude hacer que tomara los centrales.

Estado de la lista antes de la colisión:

```
[0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]


Dime el numero: 3
Posicion:3

Colision, doble direccion hash por hashCuadrado:

numero: 9 tamanio: 1
Posicion:9

El numero 3 ha sido insertado en la posicion: 9

[0, 0, 0, 3, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Intento de agregar el numero 3 una tercera vez:

```
Opcion:1

Dime el numero: 3
Posicion:3

Colision, doble direccion hash por hashCuadrado:

numero: 9 tamanio: 1

Colision, doble direccion hash por hashCuadrado:

numero: 81 tamanio: 2

Colision, doble direccion hash por hashCuadrado:

numero: 6561 tamanio: 4
Digitos centrales: 5 y : 6
Nueva posicion: 56

No se pudo insertar ese numero en la lista

[0, 0, 0, 3, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Al llegar a 81 no lo inserta por ser un número mayor a la capacidad de nuestra lista, por lo que vuelve a hacer la doble dirección de hash, y al no encontrar una posición valida la tercera iteración marca como que no se puede insertar.

Ejercicio 2: Encadenamiento

Para este ejercicio no hubo mayor problema, ayudo mucho lo aprendido en el ejercicio anterior. Para la función aleatoria utilicé math.random() multiplicado por 14 ya que la lista tiene 15 espacios y haciéndole un cast a int porque regresa un double. Cree y añadí las 15 listas a la lista de listas, y para insertar un elemento simplemente lo agregué a la lista que está en la posición dada por la función aleatoria.

Al agregar un numero:

```
Bienvenido a la practica 5 elige una opcion:

1)Funcion Hash por modulo
2)Encadenamiento
3)Manejo de tablas hash en java
4)Salir

Opcion:2

Encadenamiento elige una opcion:

1)Agregar elementos
2)Volver a menu principal

Opcion:1

Dime el numero: 331

El numero 331 ha sido agregado a la lista de posicion: 10
[[], [], [], [], [], [], [], [], [], [], [331], [], [], [], []]
```

Después de agregar más números:

```
Dime el numero: 445

El numero 445 ha sido agregado a la lista de posicion: 11
[[], [], [], [], [], [], [], [], [], [], [331], [232, 445], [], [], []]
```

Después de agregar aún más números:

```
Encadenamiento elige una opcion:

1)Agregar elementos
2)Volver a menu principal

Opcion:1

Dime el numero: 665

El numero 665 ha sido agregado a la lista de posicion: 14
[[], [23], [], [2326], [13], [432, 452, 12], [623], [666, 31], [], [111], [331, 765], [232, 445, 213], [54, 46], [777], [665]]
```

Ejercicio 3. Tablas Hash en Java

Contenido del hashtable

```
hashtable = {Piernas=Pantalon, Cabeza=Sombrero, Pies=Calcetin, Cuerpo=Playera}
```

- Contains: Busca el valor dado dentro del hashtable y regresa un true si este se encuentra o un false en caso contrario.

```
hashtable.contains()
El hashtable contine Playera ? = true
El hashtable contine Calzones? ? = false
```

- containsKey: Busca la llave dada dentro del hashtable y regresa un true si encuentra un elemento con esa llave, false en caso contrario.

```
hashtable.containsKey()
El hashtable contine algo para la Cabeza ? = true
El hashtable contine algo para las manos? = false
```

- containsValue: es lo mismo que el contains lo único que cambia es el nombre del método:

```
hashtable.containsValue()
El hashtable contine Playera ? = true
El hashtable contine Calzones? ? = false
```

- equals: sirve para comparar el valor dentro de los objetos. El objeto debe tener definido un método equals, en caso contrario actuara como un ==. En este caso el objeto es un Wrapper String por lo que comparara las cadenas:

```
hashtable.get("Cabeza").equals("Playera")
Lo que el hashtable tiene para la cabeza es un Sombrero ? = false
hashtable.get("Cuerpo").equals("Chaqueta")
Lo que el hashtable tiene para el cuerpo es Chaqueta ? = false
```

- get: devuelve el objeto almacenado en la llave dada.

```
hashtable.get("Cabeza")
Para la cabeza contiene = Sombrero
```

- put: agrega el valor con su respectiva llave en el hashtable.

```
hashtable = {Piernas=Pantalon, Cabeza=Sombrero, Pies=Calcetin}

hashtable.put("Cuerpo","Playera")
hashtable = {Piernas=Pantalon, Cabeza=Sombrero, Pies=Calcetin, Cuerpo=Playera}
```

- remove: remueve del hash table el elemento que está asignado a la llave dada.

```
hashtable = {Piernas=Pantalon, Cabeza=Sombrero, Pies=Calcetin, Cuerpo=Playera}
hashtable.remove("Pies");
Despues de remover la prenda para pies
hashtable = {Piernas=Pantalon, Cabeza=Sombrero, Cuerpo=Playera}
```

- Size: regresa la cantidad de llaves dentro del hashtable.

```
hashtable = {Piernas=Pantalon, Cabeza=Sombrero, Cuerpo=Playera}

hashtable.size();
Numero de elementos del hashtable = 3
```

Conclusiones

Considero que se lograron los objetivos de aprendizaje, ya que pude llevar a cabo todos los ejercicios de la práctica, en un principio en el laboratorio tuve muchas complicaciones ya que no entendía muy bien los conceptos del hash cuadrado y doble dirección hash, pero repasando los apuntes e investigando logre resolver mis dudas y hacer funcionar el programa.

El encadenamiento en realidad se me hizo un concepto fácil de comprender y con base a lo que vi en el ejercicio 1 no tuve complicaciones en su implementación.

En cuanto a las tablas hash fue simplemente investigar los métodos y elaborar una clase, para hacer esos métodos en un hashtable, así explique cada uno de los métodos mostrando su ejecución.

La práctica me pareció muy larga, creo a nadie le dio tiempo de terminarla en el laboratorio y parte de eso fue que no teníamos muy bien los conceptos del hash, ya que venía cosas que no vimos muy a fondo en la teoría, por ejemplo en la función hash cuadrado no sabía si tomar uno o dos dígitos ya que como solo había 20 espacios en la lista al tomar 2 dígitos puede dar muchos valores no válidos, por lo que a mi criterio tome 2 dígitos y puse una validación si estaba dentro del rango del tamaño del arreglo, aunque no sé si lo correcto sería tomar esos 2 dígitos y hacer módulo de 20 o algo por el estilo.