Algoritmos y Estructuras de Datos
UT6-PD7

Santiago Blanco

08-06-2025

Ejercicio #1

Tal parece que:

- Para la clase base `Object` en Java implementa el método `hashCode()` que se basa en la dirección de memoria del objeto para aplicar una función hash que devuelva un identificador único por defecto.
- En la clase `Integer` el método hashCode devuelve directamente el valor entero almacenado.
- La clase `String` implementa un algoritmo de hashing que combina los caracteres del String y múltiplica por potencias de 31, lo que genera una buena dispersión de valores y minimiza colisiones en estructuras como `HashMap`.

Ejercicio #2

Un HashMap en Java está compuesto por:

- Array de celdas que almacena referencias a nodos.
  - Cada nodo contiene:
    - La clave (key)
    - El valor (value)
    - El hash de la clave
    - Una referencia al siguiente nodo para manejar las colisiones, utiliza el método de **encadenamiento separado (Separate Chaining)**

Index	Contenido
0	
1	
2	“Hola”
3	
4	“Colecciones”
5	
6	
7	“HolaMundo”
8	“HashMap”
9	
10	
11	
12	
13	
14	
15	

Ejercicio #3

```java
public class Alumno {
    private int ID;
    private String fullName;
    private String email;

    @Override
    public boolean equals(Object obj) {
      /* evidente, si el objeto es nulo o si es de diferente clas no los
      * compara y devuelve false */
        if (obj == null || getClass() != obj.getClass()) {
          return false;
        }
        Alumno muchacho = (Alumno) obj;
        // Tiene que estar igualito
        return ID == muchacho.ID &&
               fullName.equals(muchacho.fullName) &&
               email.equals(muchacho.email);
    }

    @Override
    public int hashCode() {
        int resultado = Integer.hashCode(ID);
        resultado = 31 * result + fullName.hashCode();
        resultado = 31 * result + email.hashCode();
        return resultado;
    }
}
```