

Pregunta 1

given

```
class MyKeys{
    Integer key;
    MyKeys(Integer K){key = k;}
    public boolean equals(Object o){
        return ((MyKeys)o).key == this.key;
    }
}
```

And this code snippet:

```
Map m = new HashMap();
MyKeys m1 = new MyKeys(1);
MyKeys m2 = new MyKeys(2);
MyKeys m3 = new MyKeys(1);
MyKeys m4 = new MyKeys(new Integer(2));
m.put(m1, "car");
m.put(m2, "boat");
m.put(m3, "plane");
m.put(m4, "bus");
System.out.print(m.size());
```

What is the result?

- A. 2
- B. 3
- C. 4
- D. Compilation fails

Explicacion

Para entender el resultado del código, necesitamos analizar la clase `MyKeys` y cómo se comporta en el contexto de un `HashMap`.

Clase `MyKeys`

- La clase tiene un campo `Integer key`.
- El constructor inicializa `key`.
- El método `equals` compara los valores de `key` de dos objetos `MyKeys` usando `==`.

Análisis

1. Creación de objetos **MyKeys**:

- m1 es un **MyKeys** con key igual a 1.
- m2 es un **MyKeys** con key igual a 2.
- m3 es un **MyKeys** con key igual a 1.
- m4 es un **MyKeys** con key igual a 2.

2. Inserciones en el **HashMap**:

- `m.put(m1, "car");`
 - Inserta m1 como clave con valor "car". No hay conflicto, se añade al mapa.
- `m.put(m2, "boat");`
 - Inserta m2 como clave con valor "boat". No hay conflicto, se añade al mapa.
- `m.put(m3, "plane");`
 - Aquí es donde las cosas se ponen interesantes. Aunque m1 y m3 tienen la misma key (1), no se consideran iguales en términos de **HashMap** porque la clase **MyKeys** no sobrescribe `hashCode()`.
 - Por lo tanto, m3 es tratada como una clave separada y se añade al mapa.
- `m.put(m4, "bus");`
 - Similarmente, m2 y m4 tienen la misma key (2), pero son tratados como claves diferentes debido a la falta de `hashCode()`.
 - Por lo tanto, m4 es tratada como una clave separada y se añade al mapa.

3. Resultado:

- Al final, hay 4 claves diferentes (m1, m2, m3, m4) en el **HashMap** porque `hashCode()` no se sobrescribe, y por lo tanto cada objeto se trata como una clave única.
- `m.size()` devuelve 4.

Conclusión

El resultado es **C. 4**.