**Methods hashCode and equals**

Object.equals compares according the address, not content

Object.hashCode calculates an object’s hash code based on its address, not its contents

Most predefined classes also override method hashCode

When you override equals method, it is better to override hashCode as well. Otherwise, you violate the following rule:

If obj1.equals(obj2) is true, then obj1.hashCode() == obj2.hashCode()

Both methods should use the same data field(s) of the class

**Implementing HashSetOpen**

We can modify the hash table methods to implement a hash set.

* For example, the Set contains method performs a test for set membership instead of retrieving a value

Table

Description automatically generated

You can delegate Set methods to Map methods easily. To avoid writing new methods from scratch, implement HashSetOpen as an adapter class:

Text, letter

Description automatically generated

Text, letter

Description automatically generated

**Implementing the Java Map and Set Interfaces**

The Java API uses a hash table to implement both the Map and Set interfaces

The task of implementing the 2 interfaces is simplified by the inclusion of abstract classes AbstractMap and AbstarctSet in the Collection hierarchy

We overrode the O(n) implementations of the get and put methods (in AbstractMap) with O(1) implementations in **HashtableOpen** and **HashtableChain**

In AbstractMap, get operation is linear time. get is implemented by iterator (Set and Map are Iterable). AbstractMap has iterator method that returns an iterator. That iterator has to go through all the elements one by one. So it is linear time for the worst case.

In HashMap class, we can use get in constant time.

**Nested Interface Map.Entry**

Key-value pairs for a Map object must implement the interface Map.Entry<K, V> which is an inner interface of interface Map

An implementer of the Map interface must contain an inner class that provides code for the methods in the table below:

Graphical user interface, application

Description automatically generated

**Creating a Set View of a Map**

Method entrySet creates a set view of the entries in a Map

The members of the set returned are the key-value pairs defined for the Map object

* Example: if a key is “0123” and the corresponding value is “Jane Doe”, the pair (“0123”, “Jane Doe”) is an element of the set view

The set is called a view because it provides an alternative way to access the contents of the Map

entrySet usually is called by a statement of this form:



Method entrySet and Classes EntrySet and SetIterator

Text, letter, email

Description automatically generated