

TECH FOR N00BS & L33TS


MAP INTERFACES

NESTED CLASSES

Summarizer



What is hash...

HASH	 ORIGIN	WEED
BUD RESIN	THC	BUD FLOWER
40%	MEDICAL USE	10-20%
✓	EFFECTS	✓
STRONGER EFFECTS	EDIBLE	STANDARD EFFECTS
VERY INTENSE		INTENSE



Finalmente um
tema
interessante...



What is hash(ing)... in JAVA!

Hash function – Magical function that can give a number to any object!

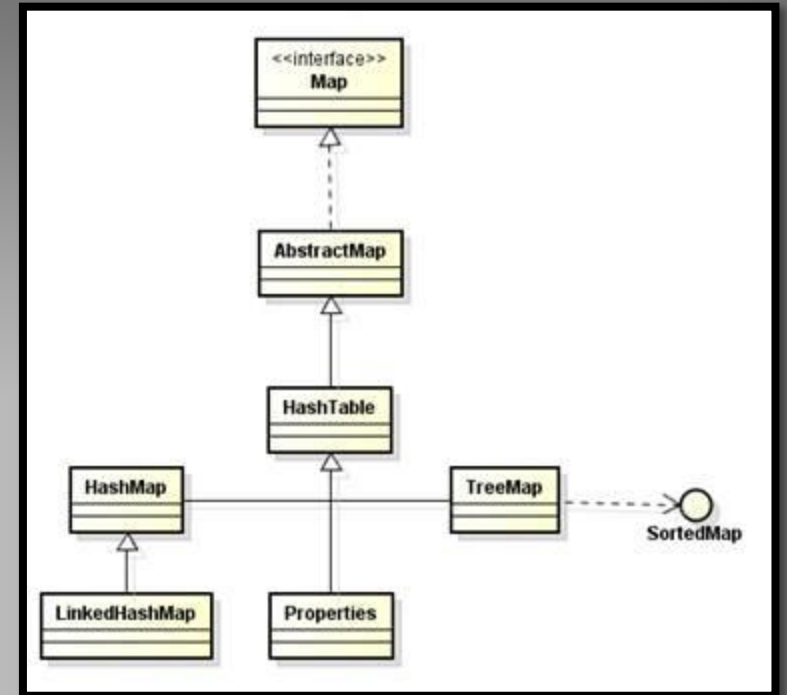
Uses a HashTable (data structure that can map keys and values)

Allows...

Rapid data lookup!

But... what if two objects have the same number?

Collisions make them less useful



Map Interface

It's a Collection!

...maybe not...

but...

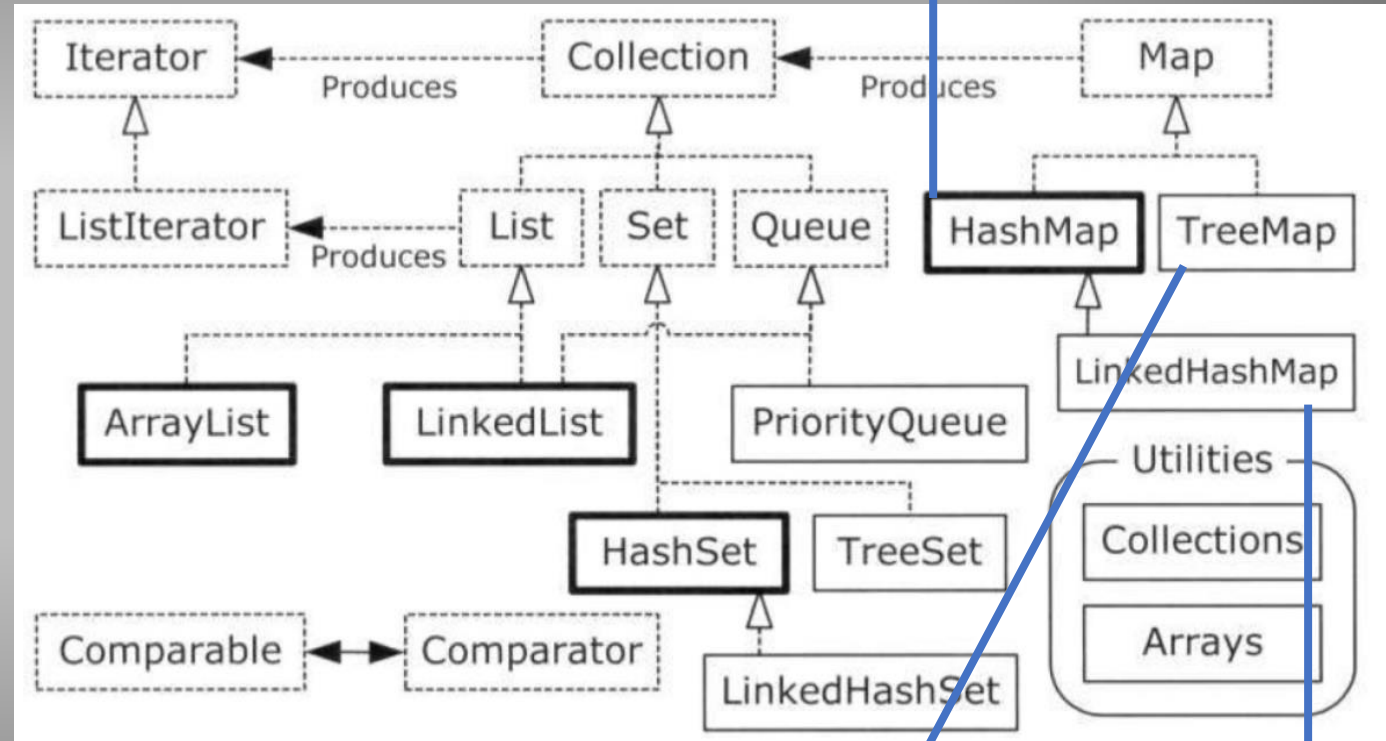
So it's kind of a Collection!

Implements:

```
put();  
get();  
keyset();
```

Map for the best hash around!

Framework

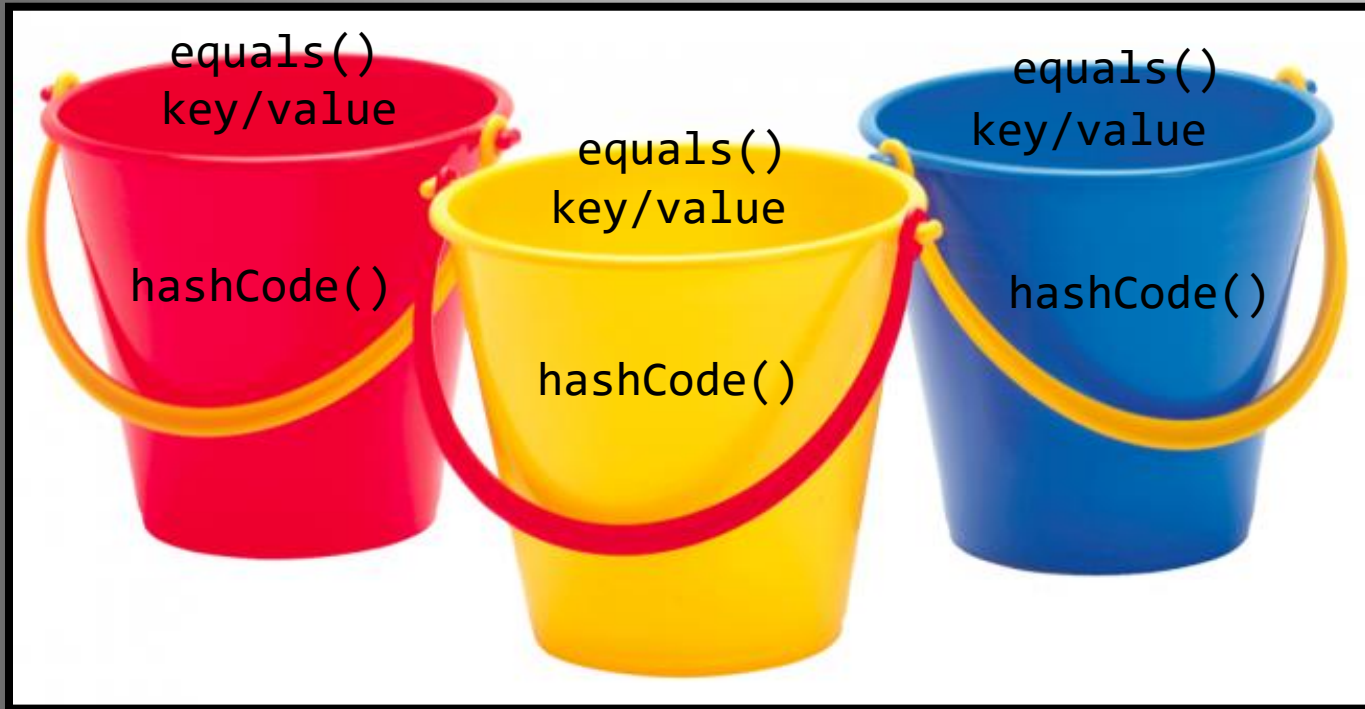


Organized in a sorted order
defined by the Comparable
Interface

HashMap but in
insertion order when
iterating

Hash Map

What about that Map...



Buckets!



hashCode() and equals() contract

hashCode() and equals() should always use Object properties for keys and values (uses object address by default)

No mutable data;

If(ObjectA.equals(ObjectB)) → must return the same hashCode value;

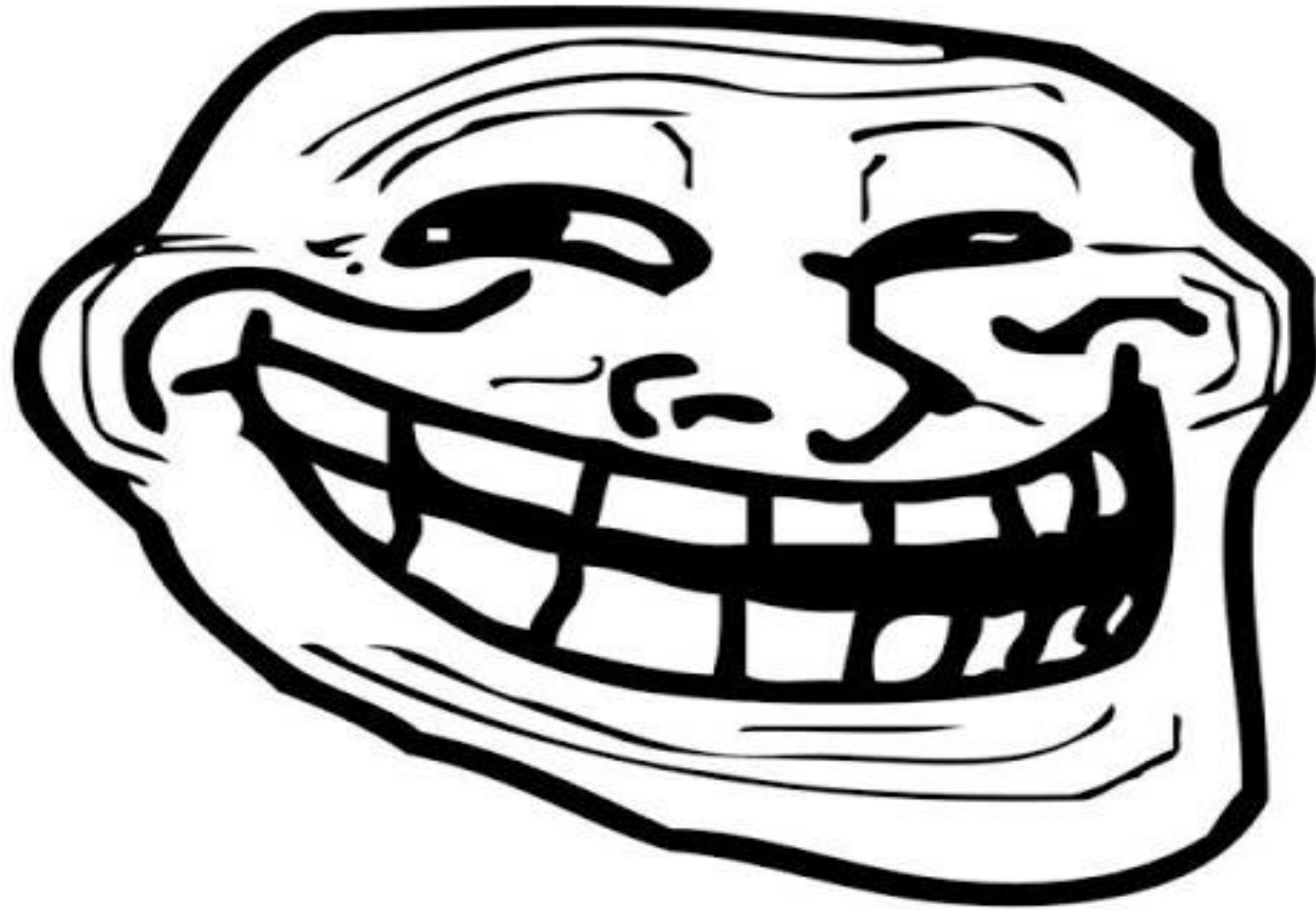
If(!ObjectA.equals(ObjectB)) → they (shouldn't but..) can return the same hashCode value



WOMEN AREN'T *OBJECTS*

**THEY'RE EACH IN
A CLASS
OF THEIR OWN**

imgflip.com



problem?

[Link to](#)

[-done!](#)

Revision Concept

Easy question of the day: *What is a Local Variable?*

Properties lifecycle

When Statics are born

Class Loading

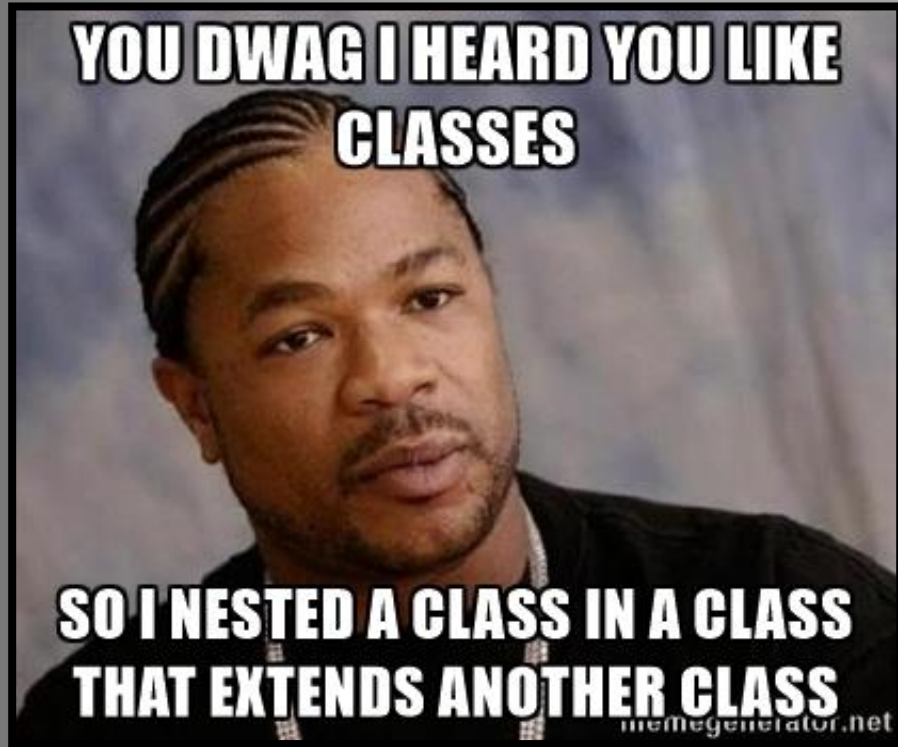
Encapsulation

Members of a Class

Biggest Intro of all time for...

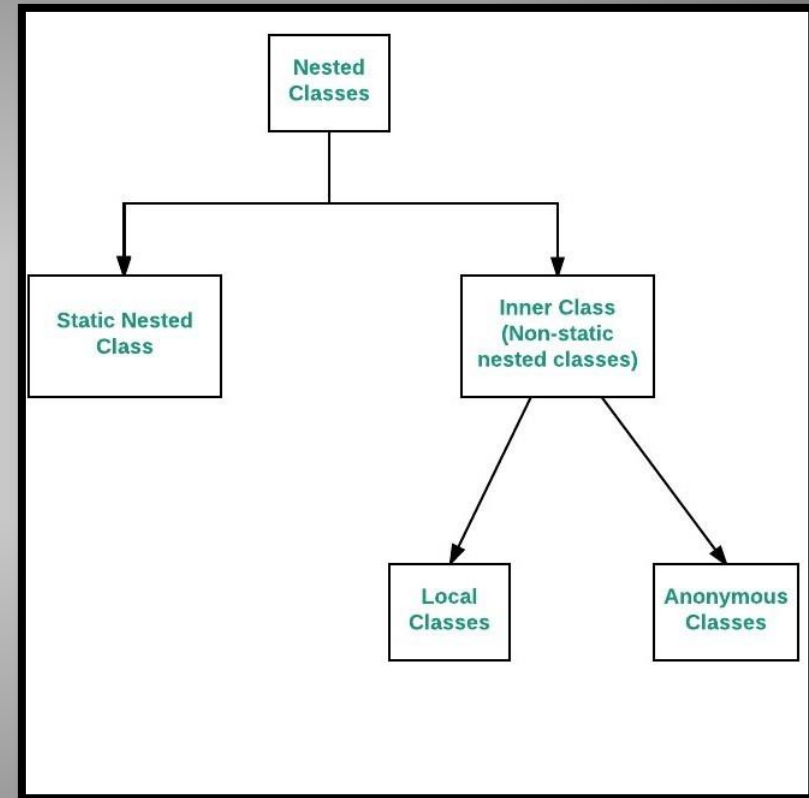


Nested Classes



Supporting Classes are meaningful
But make no sense on their own!

Increased Encapsulation



Readable and Maintainable code
(Manteneutenivéle ou lá como o Vando diz)

Static Nested Class

Package Convenience;

Cannot refer directly to
instance variables or methods;

Use methods and variables only
through a reference
Of the Outer Class;



```
public class ImTheBoss {  
    public static class HesMyBoss {  
    }  
}
```



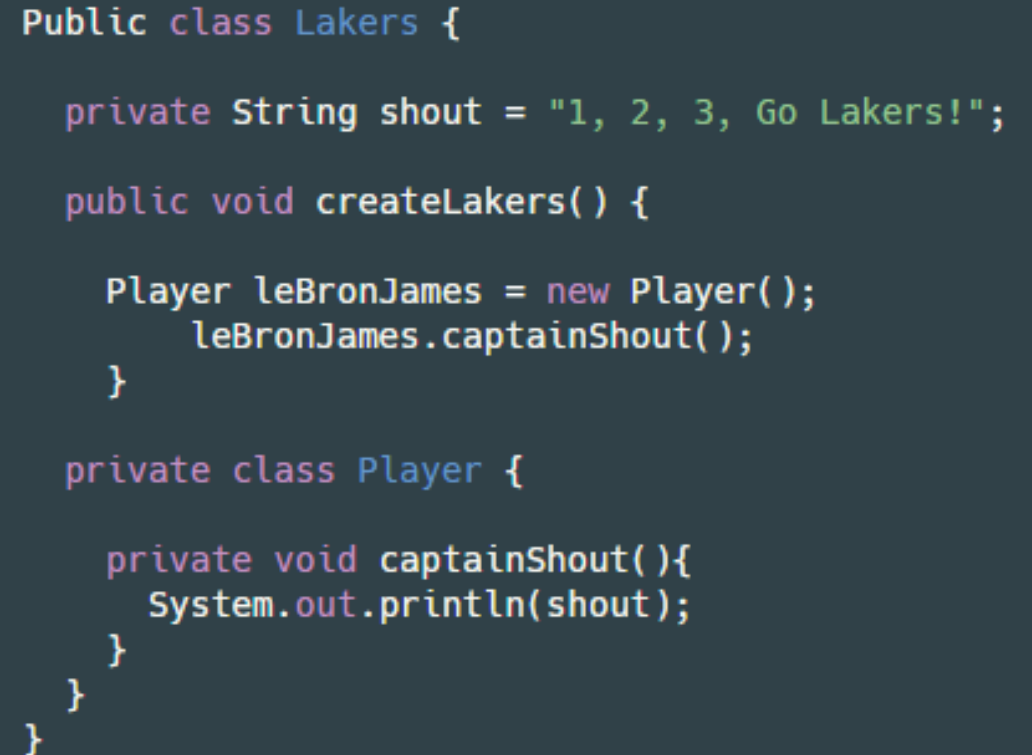
```
ImTheBoss.HesMyBoss newBoss = new ImTheBoss.HesMyBoss();
```


Inner Class

Only exists within an instance of its Outer Class;

Direct access to methods and properties of the Outer Class, even if private;

Cannot declare statics;



```
Public class Lakers {  
    private String shout = "1, 2, 3, Go Lakers!";  
    public void createLakers() {  
        Player leBronJames = new Player();  
        leBronJames.captainShout();  
    }  
    private class Player {  
        private void captainShout(){  
            System.out.println(shout);  
        }  
    }  
}
```

Local Inner Class

Defined inside any block;

Access to the members of their Outer Class, even if private;

Access to local final variables;



```
public class Cat {  
    public CutestCat cutestCatOfAllTime() {  
        final boolean cute = true;  
        String breed = "Norwegian Forest";  
        class CutestCat implements Adorable{  
            public String getBreed {  
                return breed; // COMPILER ERROR  
            }  
            public boolean isCute() {  
                return cute; // OK  
            }  
        }  
        return new CutestCat();  
    }  
}
```



{ Exercise }

Bi-directional integer Range class

- A Range (outer) class that implements Iterable
- A Bidirectional Iterator anonymous class

[Link to My Code, that I made, that is a mess and I am sure there is a better way, but... it works!](#)