

## \*Coding Tricks

How to find source code? command + click

Code fragment searching? command + F

**About java style? class naming/variable naming/method naming/attribute naming/formatting?**

About your HW (leetcode 88, code see moodle)

## 第一部分 Inheritance (继承)

Definition: A class that is derived from another class is called subclass and **inherits** all fields and methods of its superclass. Only single **inheritance** is allowed.

理解: 这里single inheritance指一个class只能继承一个class, 不能继承多个。这和multiple-inheritance相对立。

考点: 什么是multiple-inheritance呀? Java有没有multiple-inheritance呀?

问题: Java的三大特性(or simply put the three great properties of OOP)?

public boolean equals(Object other)

能override就override, 因为不更新的话, 调用时候的还是parent class **Object** 里面的那个起效果。

What is hashCode()? Relationship with equals()?

回忆148中hashing是怎么回事。说白了就是一种mapping的关系。

问题: 是否一定要求一一对应?

不一定, 只是尽量map, 可能有collision (碰撞)。

对于hashCode的要求:

两个object的.equals()返回值为true那么hash-code一定相同

注意: hashCode()返回值相同, equals()不一定为true。

What does super mean?

super.attribute  
super(par1, par2...) ——> Constructor

## Inheritance + Constructor? (常考)

No explicit super? 没有明确的super调用的时候发生了什么?

默认constructor被调用! 默认? 不接受参数

如果superclass没有默认的constructor怎么办?

## Method Overriding

Allows a subclass or child class to **provide a specific implementation** of a method that is already provided by one of its superclasses or parent classes.

理解: 更新!!!

形式上

i. @override

ii. 两个method, 一个在parent class里面, 一个在child class里面,  
两个method名字完全一致, parameter数量, type, 顺序完全一致。

## Method Overloading

**Method Overloading** is a feature that allows a class to have two or more **methods** having same name, if their argument lists are different.

理解: 名字一样, 但是接受的参数不同了。

最常见的例子: 多个 Constructor

形式上:

i. @overload

ii. 本质来讲两个method都在同一个class里面, 但是吃进去的东西不同。(可能是parameter数量不同, type不同, 顺序不同)

## Name Lookup (最常考, 最易错, 最难理解, 最糊涂的, 例题参见Big List)

首现, 我们分两类:

1. static类(static, overload, final)

这一类全部按照reference type

问题: 什么是reference type

2. 非static类 (override)

首选object的

然后, subclass没有找到, 一律向上找

最后最易错点：能否通过compile?

技巧就一个：看能否找到

## Abstract Class

不能instantiate!!! (和148相同)

需要abstract keyword

## Class Composition

Instance variables referring to other class objects.

理解：一个class用其他class作为attribute

使用：

```
class Engine {  
}  
  
class Car {  
    Engine e;  
}
```

Always prefer composition to inheritance.

理解：能用composition就用，能不用inheritance就不用。

原因：composition也能达到效果，且更安全。class之间联系紧密有风险。

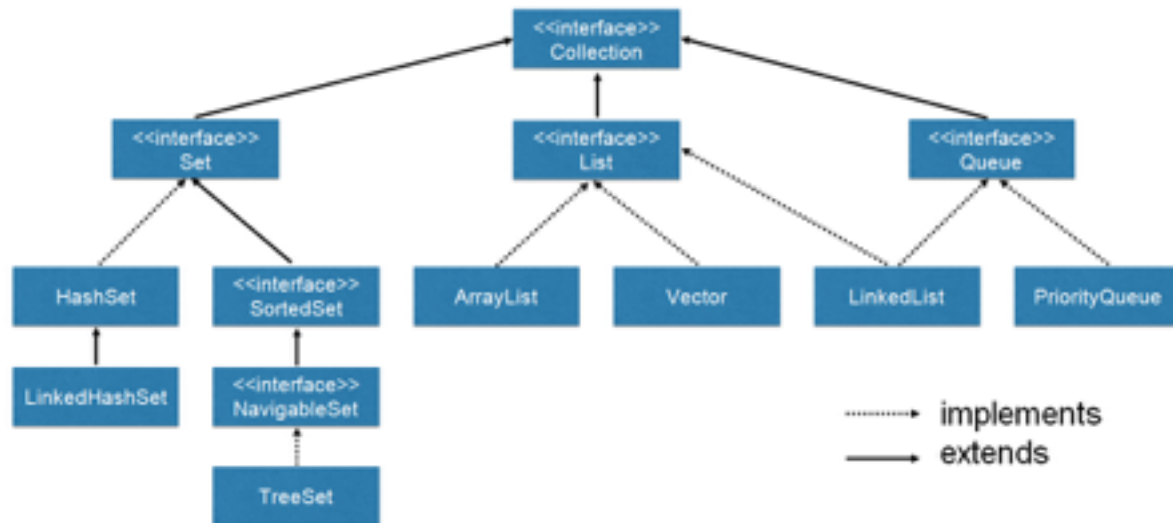
## 第二部分

### Collection (Java Collection Framework, JCF)

Definition. 一组实现了基本data structure的class和interface。基本data structure指  
LinkedList, Queue, Stack... (回忆148)

ArrayList, LinkedList

# Collection Interface



ArrayList和LinkedList都是Java Collection里面的data structure。两者implements List这个interface。

两者有微小区别。一般使用ArrayList就可以。

思考一下的几种情况：

List a = new ArrayList(); // 出现warning， compiler推荐你说明type

List<String> a = new ArrayList(); // 没有任何问题

List<int> a = new ArrayList(); // compiler报错， 因为type必须是class， 非primitive

ArrayList<String> a = new ArrayList();

Map (HashMap, TreeMap), Set(HashSet, TreeSet)类别

Stack, Queue, PriorityQueue类别

**在这里要强调， 最重要的三类built-in data type: List, Map, Set**

作业：用array implement Stack, Queue, \*PriorityQueue.

课前小问题答案：

1. 两个（或多个）人在同一位置做了不同修改修改，后做修改的人没有提前pull，而是在修改后pull。resolve需要一个人做edit，然后add+commit+push，另一个人第一时间pull。
2. 用.equals()这个method。（注意不是什么\_\_eq\_\_，已经不是python了）。用这个method的根本原因是因为Java中String不是primitive了，而是class。

3. private, public, package-private (也就是什么都不写的情况下是这个), protected (今天讲)
4. Constructor相当于python的\_\_init\_\_, 用于instantiate (中文“新建”) class对应的object。Java中允许多个constructor, 甚至可以没有constructor。Python中每个class只能有一个constructor。

Next Lecture: GUI (A1 part2需要)、UML深入  
HW: leetcode 437