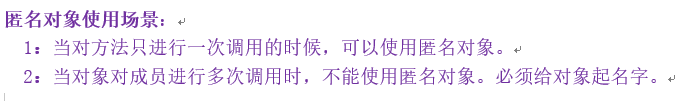
# 基础知识学习总结

这一部分知识，皆有我会的，又有一部分是我原来不会的。原来我会的看得比较快，并且对这些知识又有了一些新的理解。而我原来不会的那一部分，经过这几天的学习，通过从网上找资料，翻阅书籍等等，读这些只是也有了一个大概的了解。

## 我重新学到的

1.



因为以前使用匿名对象较少，对匿名对象了解的不够深彻。通过这一次又重新了解了一遍。知道了如何使用匿名对象以及什么时候使用匿名对象。

2.



以前一直以为构造函数不是函数，现在知道我以前一直是错的。

3.



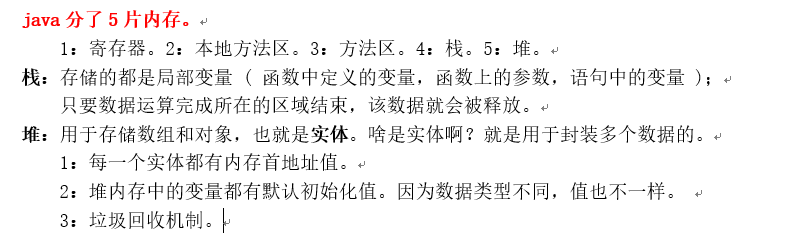
以前一直以为java中只有单继承，不存在多继承。下那知道了接口可以实现多继承。

4.

知道了递归的优越性，并且在使用递归的时候一定要加上合适的退出条件。

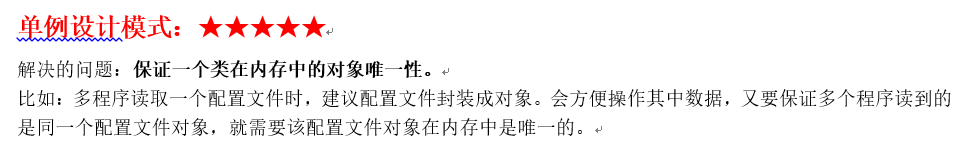
## 新的理解

1.



以前只知道java中是有好几块内存空间的，具体是什么不清楚。然后在这几天通过上网查资料有了一个叫深入的理解。

2..



以前听过这种设计模式，知道设计的原理，但是不知道如何用代码实现，现在不仅知道了如何实现，还知道了饿汉式和懒汉式两种思路。

3.



当子类实例化的时候，会调用父类的构造方法，因此抽象类也会有构造函数，并且可以有多个构造函数。Java中的多态机制就是调用父类的构造方法。

4.

对字符串的讲解看完整个文档后不是很懂，从网上找了一些知识点又重新看了一遍，知道了String,StringBuffer,StringBuilder三种格式操作字符串，并通过代码了解到StringBuffer和StringBuilder的高效率。并且知道了对着三种格式字符串的基本操作，比如插入，删除，修改等等。

5.

知道了API是在开发过程中很重要的一个工具，就好比是java中的一个字典，里面包含了java中的所有方法，并且通过类名或接口对这些方法进行了整理。

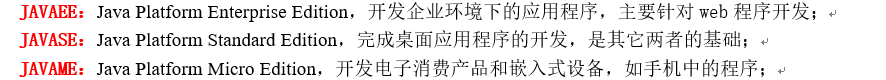
6.

IO流分为字节流和字符流，并且在对文件进行读写操作的时候，采用装饰设计模式，几乎所有的读写操作都会加上缓冲区。而且在异常处理的时候，最后都要加上finally进行资源的关闭。并且IO流和文件file类有密切的联系。而且IO流的偶作基本是固定的，相对来说比较简单。

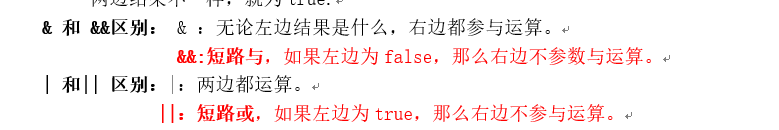
7.

文件类里有很多方法，可以用来在文件读写的时候进行判断。方便对文件的读写。

## 新学到的



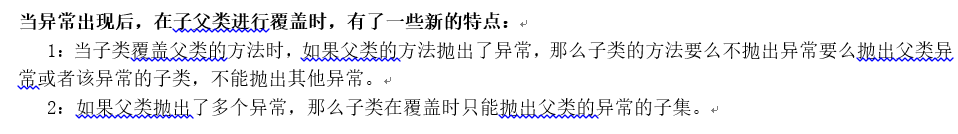
上面这部分让我对java 有了一个系统的理解。以前听别人说一直不明白这三个词具体代表什么。



这部分是我以前不知道的，现在对java为了提高效率，有了一个新的理解。



以前一直不太关心这些比较基础的东西，只关心如何用代码实现功能。



因为平时对异常的处理比较少，因此对异常的理解不够深入，只对异常有一个最基本的理解。

知道了集合可以用来操作对象，可以很方便地对其中的元素进行增删改查，还可以用迭代器对其中的元素进行遍历（以前只会用循环），而且在java中万物皆对象，因此可以用集合来操作所有数据，尤其是在jdk更新之后，对基本数据类型有了自动装箱和自动拆箱，使用集合更加方便了。

使用泛型的主要应用场景好像是和集合结合到一起使用，用来指定集合中存放的数据类型。避免将多个类型的数据放到同一个集合中。

多线程的效率高，但是不安全。单线程的效率低，但是安全性高。我们实现多线程可以自己写一个实体类继承Thread这个类，然后重写里面的run方法，这个方法里面写我们要实现的功能，或者是实现runable这个接口也可以实现多线程。

## 有疑惑的

1.



当子类继承父类，然后实现重载时，算不算在两个类中实现重载？

2.





如果在子类的构造函数中指明调用父类的带参构造，是不是不会在调用父类的无参构造？

## 总结

经过对java基础的重新梳理，发现了很多以前自己不会的，甚至有很多是自己理解错了的，感觉这一阶段的学习很有必要，也很重要，要不然可能将来会造成很大的错误和麻烦。也给我以后的学习有了很大的警示，不管学的是什么，既然学了，就要认真学，尤其是要把基础打牢，不然将来还要花时间重新梳理自己的知识体系，还不一定能梳理出来。因此，应该重视基础，认真对待每一件事，既然做了就应该尽自己的最大努力去做好一件事。