###13.01_常见对象(StringBuffer类的概述)

* A:StringBuffer类概述

    * 通过JDK提供的API，查看StringBuffer类的说明

    * 线程安全的可变字符序列

* B:StringBuffer和String的区别

    * String是一个不可变的字符序列

    * StringBuffer是一个可变的字符序列

###13.02_常见对象(StringBuffer类的构造方法)

* A:StringBuffer的构造方法：

    * public StringBuffer():无参构造方法

    * public StringBuffer(int capacity):指定容量的字符串缓冲区对象

    * public StringBuffer(String str):指定字符串内容的字符串缓冲区对象

* B:StringBuffer的方法：

    * public int capacity()：返回当前容量。 理论值(不掌握)

    * public int length():返回长度（字符数）。 实际值

* C:案例演示

    * 构造方法和长度方法的使用

###13.03_常见对象(StringBuffer的添加功能)

* A:StringBuffer的添加功能

    * public StringBuffer append(String str):

        * 可以把任意类型数据添加到字符串缓冲区里面,并返回字符串缓冲区本身

* public StringBuffer insert(int offset,String str):
    * 在指定位置把任意类型的数据插入到字符串缓冲区里面,并返回字符串缓冲区本身

###13.04_常见对象(StringBuffer的删除功能)
* A:StringBuffer的删除功能
    * public StringBuffer deleteCharAt(int index):
        * 删除指定位置的字符，并返回本身
    * public StringBuffer delete(int start,int end):
        * 删除从指定位置开始指定位置结束的内容，并返回本身

###13.05_常见对象(StringBuffer的替换和反转功能)
* A:StringBuffer的替换功能
    * public StringBuffer replace(int start,int end,String str):
        * 从start开始到end用str替换
* B:StringBuffer的反转功能
    * public StringBuffer reverse():
        * 字符串反转

###13.06_常见对象(StringBuffer的截取功能及注意事项)
* A:StringBuffer的截取功能
    * public String substring(int start):
        * 从指定位置截取到末尾
    * public String substring(int start,int end):
        * 截取从指定位置开始到结束位置，包括开始位置，不包括结束位置

* B:注意事项

    * 注意:返回值类型不再是StringBuffer本身

###13.07_常见对象(StringBuffer和String的相互转换)

* A:String -- StringBuffer

    * a:通过构造方法

    * b:通过append()方法

* B:StringBuffer -- String

    * a:通过构造方法

    * b:通过toString()方法

    * c:通过subString(0,length);

###13.08_常见对象(把数组转成字符串)

* A:案例演示

    * 需求：把数组中的数据按照指定个格式拼接成一个字符串

    *

举例：

int[] arr = {1,2,3};

输出结果：

"[1, 2, 3]"

用StringBuffer的功能实现

###13.09_常见对象(字符串反转)

* A:案例演示

*

需求：把字符串反转

举例：键盘录入"abc"

输出结果："cba"

用StringBuffer的功能实现

###13.10_常见对象(StringBuffer和StringBuilder的区别)

* A:StringBuilder的概述

    * 通过查看API了解一下StringBuilder类

* B:面试题

    * String,StringBuffer,StringBuilder的区别

    * StringBuffer和StringBuilder的区别

    * StringBuffer是jdk1.0版本的,是线程安全的,效率低

    * StringBuilder是jdk1.5版本的,是线程不安全的,效率高

    * String和StringBuffer,StringBuilder的区别

    * String是一个不可变的字符序列

    * StringBuffer,StringBuilder是可变的字符序列

###13.11_常见对象(String和StringBuffer分别作为参数传递)

* A:形式参数问题

    * String作为参数传递

    * StringBuffer作为参数传递

* B:案例演示

    * String和StringBuffer分别作为参数传递问题

###13.12_常见对象(数组高级冒泡排序原理图解)

* A:画图演示

*

需求：

数组元素：{24, 69, 80, 57, 13}

请对数组元素进行排序。

冒泡排序

相邻元素两两比较，大的往后放，第一次完毕，最大值出现在了最大索引处

###13.13_常见对象(数组高级冒泡排序代码实现)

* A:案例演示

* 数组高级冒泡排序代码

###13.14_常见对象(数组高级选择排序原理图解)

* A:画图演示

* 需求：

* 数组元素：{24, 69, 80, 57, 13}

* 请对数组元素进行排序。

* 选择排序

* 从0索引开始，依次和后面元素比较，小的往前放，第一次完毕，最小值出现在了最小索引处

###13.15_常见对象(数组高级选择排序代码实现)

* A:案例演示

* 数组高级选择排序代码

###13.16_常见对象(数组高级二分查找原理图解)

* A:画图演示

    * 二分查找

    * 前提：数组元素有序

###13.17_常见对象(数组高级二分查找代码实现及注意事项)

* A:案例演示

    * 数组高级二分查找代码

* B:注意事项

    * 如果数组无序，就不能使用二分查找。

        * 因为如果你排序了，但是你排序的时候已经改变了我最原始的元素索引。

###13.18_常见对象(Arrays类的概述和方法使用)

* A:Arrays类概述

    * 针对数组进行操作的工具类。

    * 提供了排序，查找等功能。

* B:成员方法

    * public static String toString(int[] a)

    * public static void sort(int[] a)   //排序

    * public static int binarySearch(int[] a,int key)
//二分查找

###13.19_常见对象(基本类型包装类的概述)

* A:为什么会有基本类型包装类

    * 将基本数据类型封装成对象的好处在于可以在对象中定义更多的功能方法操作该数据。

* B:常用操作

* 常用的操作之一：用于基本数据类型与字符串之间的转换。

* C:基本类型和包装类的对应

*

byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

###13.20_常见对象(Integer类的概述和构造方法)

* A:Integer类概述

* 通过JDK提供的API，查看Integer类的说明

* Integer 类在对象中包装了一个基本类型 int 的值，

* 该类提供了多个方法，能在 int 类型和 String 类型之间互相转换，

* 还提供了处理 int 类型时非常有用的其他一些常量和方法

* B:构造方法

* public Integer(int value)

* public Integer(String s)

* C:案例演示

* 使用构造方法创建对象

###13.21_常见对象(String和int类型的相互转换)

* A:int -- String
	* a:和""进行拼接
	* b:public static String valueOf(int i)
	* c:int -- Integer -- String(Integer类的toString方法())
	* d:public static String toString(int i)(Integer类的静态方法)
* B:String -- int
	* a:String -- Integer -- int
	* public static int parseInt(String s)

###13.22_常见对象(JDK5的新特性自动装箱和拆箱)
* A:JDK5的新特性
	* 自动装箱：把基本类型转换为包装类类型
	* 自动拆箱：把包装类类型转换为基本类型
* B:案例演示
	* JDK5的新特性自动装箱和拆箱

	* Integer ii = 100;
	* ii += 200;
* C:注意事项
	* 在使用时，Integer x = null;代码就会出现NullPointerException。
	* 建议先判断是否为null，然后再使用。

###13.23_常见对象(Integer的面试题)
* A:Integer的面试题

*

看程序写结果

```
Integer i1 = new Integer(97);
Integer i2 = new Integer(97);
System.out.println(i1 == i2);
System.out.println(i1.equals(i2));
System.out.println("----------");

Integer i3 = new Integer(197);
Integer i4 = new Integer(197);
System.out.println(i3 == i4);
System.out.println(i3.equals(i4));
System.out.println("----------");

Integer i5 = 97;
Integer i6 = 97;
System.out.println(i5 == i6);
System.out.println(i5.equals(i6));
System.out.println("----------");

Integer i7 = 197;
Integer i8 = 197;
System.out.println(i7 == i8);
System.out.println(i7.equals(i8));
```

###13.24_day13总结

* 把今天的知识点总结一遍。