Object对象

面向对象的核心思想:“找合适的对象，做适合的事情”。

合适的对象：

自己描述类，自己创建对象。

sun已经描述了好多常用的类，可以使用这些类创建对象。

API（Application Program Interface）

sun定义的那么多类的终极父类是Object。Object描述的是所有类的通用属性与方法。

toString方法

toString() 返回对象的描述信息　java.lang.Object@de6ced　类名@哈希码值的十六进制形式。

直接输入一个对象的时候，会调用对象的toString方法。

练习：自定义一个Person类，打印该对象的描述信息，要求描述信息为：姓名 — 年龄

问题：调用p的toString方法时，打印出来的信息是类名+内存地址值。不符合要求。根据我们之前学的继承，假如父类的指定的功能不能满足要求，那么子类可以复写父类的功能函数。那么该对象再调用toString()方法时，则会调用子类复写的toString方法。

编程习惯：开发者要对自定义的类重写toString()，对对象做详细的说明

equals方法

equals()　返回的是比较的结果　如果相等返回true，否则false，比较的是对象的内存地址值。

问题：比较两个人是否是同一个人，根据两个人的名字判断。

问题：如果根据名字去作为判断两个人是否是同一个时，明显p与p1是同一个人，但是程序输入却不是同一个人。不符合我们现实生活的要求。

解决：根据我们学的继承中的函数复写，如果父类的函数不能满足我们目前的要求，那么就可以在子类把该功能复写，达到复合我们的要求。

编程习惯：开发者要对自定义的类重写equals()，使得比较两个对象的时候比较对象的属性是否相等，而不是内存地址。

hashCode方法

hashCode() 返回该对象的哈希码值： 采用操作系统底层实现的哈希算法。 同一个对象的哈希码值是唯一的。
java规定如果两个对象equals返回true，那么这两个对象的hashCode码必须一致。

String类

String类描述的是文本字符串序列。 留言 QQ 写日志。
创建String类的对象的两种方式：
""直接赋值法
new关键字法

字符串对象的比较

String Str = “jack”这个语句会先检查字符串常量池是否存放这个”jack1”这个字符串对象，如果没有存在，那么就会在字符串常量池中创建这个字符串对象，如果存在直接返回该字符串的内存地址值。
String str3 = new String(“jack”) 该语句会创建两个对象,首先会先检查字符串常量池中存不存在jack这个字符串对象，如果不存在就会创建，如果存在就返回内存地址值。创建了出来之后，new String这个语句就会在堆内存中开辟一个字符串对象。总共两个对象。

获取方法

int length() 获取字符串的长度
char charAt(int index) 获取特定位置的字符 (角标越界)
int indexOf(String str) 获取特定字符的位置(overload)
int lastIndexOf(int ch) 获取最后一个字符的位置

判断方法

boolean endsWith(String str) 是否以指定字符结束
boolean isEmpty()是否长度为0 如：“” null V1.6
boolean contains(CharSequences) 是否包含指定序列 应用：搜索
boolean equals(Object anObject) 是否相等
boolean equalsIgnoreCase(String anotherString) 忽略大小写是否相等

转换方法

String(char[] value) 将字符数组转换为字符串
String(char[] value, int offset, int count)
Static String valueOf(char[] data)
static String valueOf(char[] data, int offset, int count)
char[] toCharArray() 将字符串转换为字符数组

其他方法

String replace(char oldChar, char newChar) 替换
String[] split(String regex) 切割
String substring(int beginIndex)
String substring(int beginIndex, int endIndex)截取字串
String toUpperCase() 转大写
String toLowerCase() 转小写
String trim() 去除空格

练习

去除字符串两边空格的函数。

```
public class Demo1 {
// 定义一个祛除字符串两边空格的函数
public static String trim( String str ){

   // 0、定义求字串需要的起始索引变量
   int start = 0;
   int end = str.length()-1;
   // 1. for循环遍历字符串对象的每一个字符
   for (int i = 0; i<str.length() ; i++ )
   {
        if ( str.charAt(i) == ' ' )
        {
             start++;
        }else{

            break;
       }
   }
   System.out.println( start );
   for (; end<str.length() && end >= 0;  )
   {
        if ( str.charAt(end) == ' ' )
        {
             end--;
        }else{
            break;
       }
   }
   System.out.println( end );
   // 2. 求子串
   if( start < end ){

    return str.substring( start , (end+1) );
   }else{

    return "_";
   }
```

获取上传文件名 "D:\\20120512\\day12\\Demo1.java"。

```
public static String getFileName2( String path ){
      return path.substring( path.lastIndexOf("\\") + 1 );
    }
}
```

将字符串对象中存储的字符反序。

```
// 将字符串对象中存储的字符反序
    public static String reaverseString( String src ){

        // 1. 将字符串转换为字符数组
        char chs[] = src.toCharArray();
        // 2. 循环交换
        for ( int start = 0 , end = chs.length - 1; start < end ;
start++,end-- )
        {
            // 3. 数据交换
            char temp = chs[end];
            chs[end] = chs[start];
            chs[start] = temp;
        }
       // 4. 将字符数组转换为字符串
         return new String( chs );
    }
```

4. 求一个子串在整串中出现的次数

```
 public static int getCount( String src , String tag ){
      // 0. 定义索引变量和统计个数的变量
      int index = 0;
      int count = 0;
      // 1. 写循环判断
      while ( ( index = src.indexOf(tag) ) != -1 )    // jackjava
      {
       // 2. 求字串
          System.out.println( src );
         src = src.substring( index + tag.length() );    // index 4 + 4
= 8
      System.out.print( src.length() + " : " + index + " :  " +
      tag.length() );
              // 3. 累加
        count++;
          }
          return count;
       }
```

StringBuffer

StringBuffer：由于String是不可变的，所以导致String对象泛滥，在频繁改变字符串对象的应用中，需要使用可变的字符串缓冲区类。

特点：

默认缓冲区的容量是16。

StringBuffer ： 线程安全的所有的缓冲区操作方法都是同步的。效率很低。

添加方法

StringBuffer("jack")　在创建对象的时候赋值
append()　　在缓冲区的尾部添加新的文本对象
insert()　　在指定的下标位置添加新的文本对象

```
StringBuffer sb = new StringBuffer("jack");
sb.append(true);
sb.append('a');
sb.append(97).append(34.0).append(new char[]{'o','o'});    // 链式编程
System.out.println( sb.toString() );           // 输出缓冲区的中文本数据
sb = new StringBuffer("jack");
sb.insert( 2, "java" );                        // jajavack
System.out.println( sb.toString() );
```

查看

toString() 返回这个容器的字符串
indexOf(String str) 返回第一次出现的指定子字符串在该字符串中的索引。
substring(int start) 从开始的位置开始截取字符串

修改(U)

replace(int start int endString str) 使用给定 String 中的字符替换此序列的子字符串中的字符。该子字符串从指定的 start 处开始，一直到索引 end - 1 处的字符

setCharAt(int index char ch) 指定索引位置替换一个字符

删除(D)

反序

reverse() 把字符串反序输出。

StringBuilder

StringBuilder 是JDK1.5之后提出的，线程不安全，但是效率要高。用法与StringBuffer类似。

System

System 可以获取系统的属性。

Runtime

Runtime 类主要描述的是应用程序运行的环境。

Date

Date 类封装的是系统的当前时间.。但是Date已经过时了，sun推荐使用Calendar类。
Calendar: 该类是一个日历的类，封装了年月日时分秒时区。

日期格式化类：SimpleDateFormat

Math

Math：类封装了很多数学的功能。

练习：生成一个随机码