常用类

Object

是所有类的父类

有十一种方法，子类能继承这十一种方法

getClass（） 判断对象的类型是否相同

toString（） 将对象内容以字符串形式输出

equals（）

public boolean equals（Object o） 比较对象内容是否相同

finalize（） 垃圾回收器，在回收垃圾对象时会自动调用

C++：创建对象（构造函数） 回收对象（析构函数）内存泄漏

java：创建对象 我们不能直接干预垃圾回收

1. 如何判定一个垃圾对象
2. 垃圾回收的时机

（Oracle--JVM--hotspot）

垃圾回收：耗费资源

垃圾回收的认定：零引用算法

垃圾回收的时机：延时回收策略

包装类

int--Integer

byte--Byte

short--Short

long--Long

float--Float

double--Double

直接父类：Number

boolean--Boolean

char--Character

基本类型和包装类的转换

int a-->Integer 1.Integer i=new Integer(a);

2.Integer i=Integer.valueOf(a);

Integer i-->int int a=i.intValue();

Integer i=10; //自动封箱 编译器自动完成基本类型和包装类型的转换

基本类型和字符串类型的转换

int a-->String String s=String.valueOf(a);

a+””;

String s-->int int a=Integer.praseInt();

包装类和字符串类型的转换

Integer i-->String String s=i.toString();

String s-->Integer 1.Integer i=new Integer(s);

2.Integer i=Integer.valueOf();

String

1. new String(char[] c);//把字符数组转换成字符串
2. toCharArray() //把字符串转换成字符数组
3. toUpperCase() //转大写
4. toLowerCase //转小写
5. contains(String s) //判断子串是否存在
6. compareTo(String nextString) //比较两个字符串
7. start’With(String s) //判断子串s是否开头
8. endWith(String s)//判断子串s是否结束
9. IndexOf(String s)//判断子串s最初的下标
10. lastIndexOf(String s)//判断子串s最末尾的下标
11. replace(String s1,String s2)//把s1替换为s2
12. subString(int a,int b)//返回子串[a,b)下标的子串
13. length()//长度
14. split(String s)//将s作为分隔符，拆分成String[]
15. trim()//去除字符串前后的空格和回车

String不可变性

在串池创建对象，可以节省空间，但是字符串长度经常改变就不会节省空间

StringBuilder

可变长字符串

例子：

String s1=”a”+”b”+”c”+”d”+”e”;

System.out.println(s1);

String s2=”a”;

StringBuilder sb=new StringBuilder(s2)

sb.append(“b”);

sb.append(“c”);

sb.append(“d”);

sb.append(“e”);

s2=sb.toString();

System.out.println(s2);

集合框架

集合：是一种工具类，是对象的容器，可用于存储数量不同的多个对象，代替数组

所有的集合：java.util

最根本接口：Collection

list:元素按顺序存储，且元素可以重复（有下标的概念）

set:元素是无序的，元素不可重复（没有下标的概念）

Collection： java.util.Collection(JDK1.2)

1.boolean add(Object o); 把元素o添加到集合中

2.boolean addAll(Collection c); 把c集合中所有的元素添加到老集合中国

3.void clear(); 清空集合中的所有元素

4.boolean contains(Object o)判断集合中是否含有o元素

5.boolean isEmpty()判断集合是否为空

6.boolean remove(Object o)从集合中删除元素o

7.int size() 返回集合长度

8.Object[] toArray()把集合转换成数组

9.boolean equals(Object o)判断o是否和集合相等

List

是一个接口

add(int pos,Object o):把元素o插入到pos位置

get(int pos):返回pos位置的元素

indexOf（Object o）：返回对象o在数组中的首下标

indexOf（Object o）：返回对象o在数组中的末下标

remove（int pos）：删除pos位置的元素

set（int pos，Object o）：把pos位置的元素替换为o

subList(int start,int end) 获得start到end的集合（[a,b)//a能取到，b取不到）

List集合的遍历

下标遍历：

迭代遍历：

for-each遍历：

forEach()遍历：

lamda表达式遍历： con.forEach(System.out::println) //JDK1.8

小结：

遍历：

1. 下标遍历：List
2. 迭代遍历：Collection（不能删除原集合中的元素）
3. for-each：Collection 数组 底层用迭代遍历
4. forEach（Consumer c）：Collection
5. lamda表达式遍历：

List接口的实现类

1.ArrayList（重点）：

数组结构存储

轻量级，查询快，增删慢

JDK1.2 操作速度快，线程不安全（并发效率高）

1. Vector

数组结构存储

重量级，查询快，增删慢

JDK1.0 操作速度慢，线程安全（并发效率低）

3.LinkedList

链表结构存储

查询快，增删慢

addFirst(Object O):把元素O添加到首位置

addLast(Object O):把元素O添加到末位置

getFirst():获得第一个元素

getLast():获得最后一个元素

removeFirst():删除第一个元素

removeLast():删除最后一个元素

泛型：

JDK1.5 主要是为了保证类型安全

<E> <T>:泛型E、T代表一种类型

泛型没有多态，赋值语句前后泛型必须一致

通配符：？

？表示任何类型

？ extends A A或A的子类

？ super A A或A的父类

泛型方法

修饰符和返回值之间加泛型

static <T> void m(T[] List<T> list){}

<T extends A> A或A的子类

<T extends A & B>A或A的子类同时B的实现类

泛型类