Java类库，目前我的版本1.8

@Retention 注解的生命周期

Super();//表示调用父类的无参构造方法

因为equals方法被重写了，所以对应的hashCode方法也得重写

System.out.println(s1);//当s1是一个对象时，自动调用父类或子类重写的toString()方法显示结果

包装类：把基本类型的变量包装成对象

Byte对应java.lang.Byte

Int 对应 java.lang.Integer, int类型包装成Integer类型称为装箱

...

String str1 = “abc”;

str1 = “123”;//改变的是str1的指向，而非”abc”字符串，不用的方法区放在字符串，等待GC垃圾 //回收机制回收，相同的字符串只留一份，放在常量池中

this.value = "".value; //空字符串””不是null

String str1 = new String(“Hello”);//创建2个对象：常量池中”Hello”，堆区中放一个字符串对象“Hello”

String str1 = "Hello"; //常量池

String str2 = "Hello"; //常量池中的

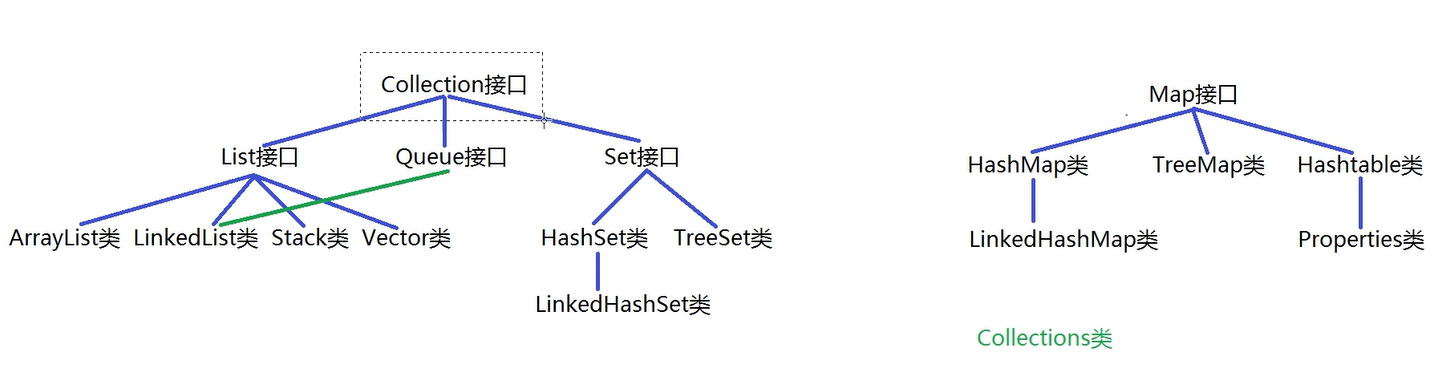
String str3 = new String("Hello");//堆区

String str4 = new String("Hello");

System.out.println(str1 == str2);//比较地址

System.out.println(str1.equals(str2));//比较内容

常用集合类型



System.out.println("集合中的元素： " + c1 ); 打印集合中的所有元素，即调用各个元素对象的同String()方法

自定义的类，一定要在类中重写Equals方法，否则使用contain方法只比较参数对象地址，而不是内容

c2.containsAll(c3);//计算交集并保留到c2中

Ctrl-F12查类方法

For each增加版For循环

Queue接口：先进先出FIFO

Stack接口：后进先出LIFO

F(int i); //i为形参

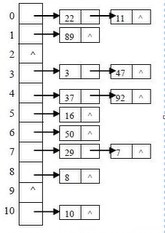
F(10); //10为实参

ArrayList<String> 泛型

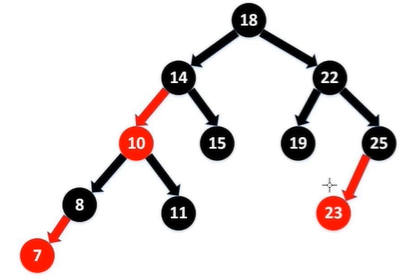
Class Substring<E> 保留父类的泛型

Set集合用于去重

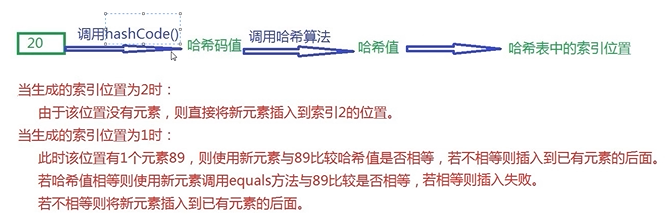
Hash表



红黑树（有序二叉树）



Hash码值和Hash值有什么区别？



为什么相等要放弃插入呢？

比较器规则排序：自定义排序规则

lamda表达式(参数列表)->{方法体}

Set集合作为Key插入Map集合

