

JAVA 预热班讲义

讲师：毕向东

String类

- 字符串是一个特殊的对象。
 - 字符串一旦初始化就不可以被改变。
 - `String str = "abc";`
 - `String str1 = new String("abc");`
- 有什么区别？

String类部分方法

- `char charAt(int index)`
- `int length()`
- `char[] toCharArray();`
- `int indexOf(String str);`
- `boolean endsWith(String str);`
- `String[] split(String reg);`
- `String substring(int index);`
- `String(char[] arr);`

练习

- 子串在整串中出现的次数。
- 两个字符串的最大相同子串。
- 对字符串中字符进行自然顺序排序。

StringBuffer

- 字符串的组成原理就是通过该类实现的。
- **StringBuffer**可以对字符串内容进行增删。
- **StringBuffer**是一个容器。
- 很多方法与**String**相同。
- **StringBuffer**是可变长度的。

StringBuffer特有方法

- StringBuffer append(int x);
- StringBuffer delete(int start, int end);
- StringBuffer insert(int index,String str);
- StringBuffer reverse();
- JDK1.5出现一个StringBuilder，区别是
StringBuffer是同步的，StringBuilder是非同步的。

基本数据类型对象包装类

- 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。
- 常用的操作之一：用于基本数据类型与字符串之间的转换。
- 例：Integer的parseInt方法，intValue方法。
- 例程

基本数据类型对象包装类新特性

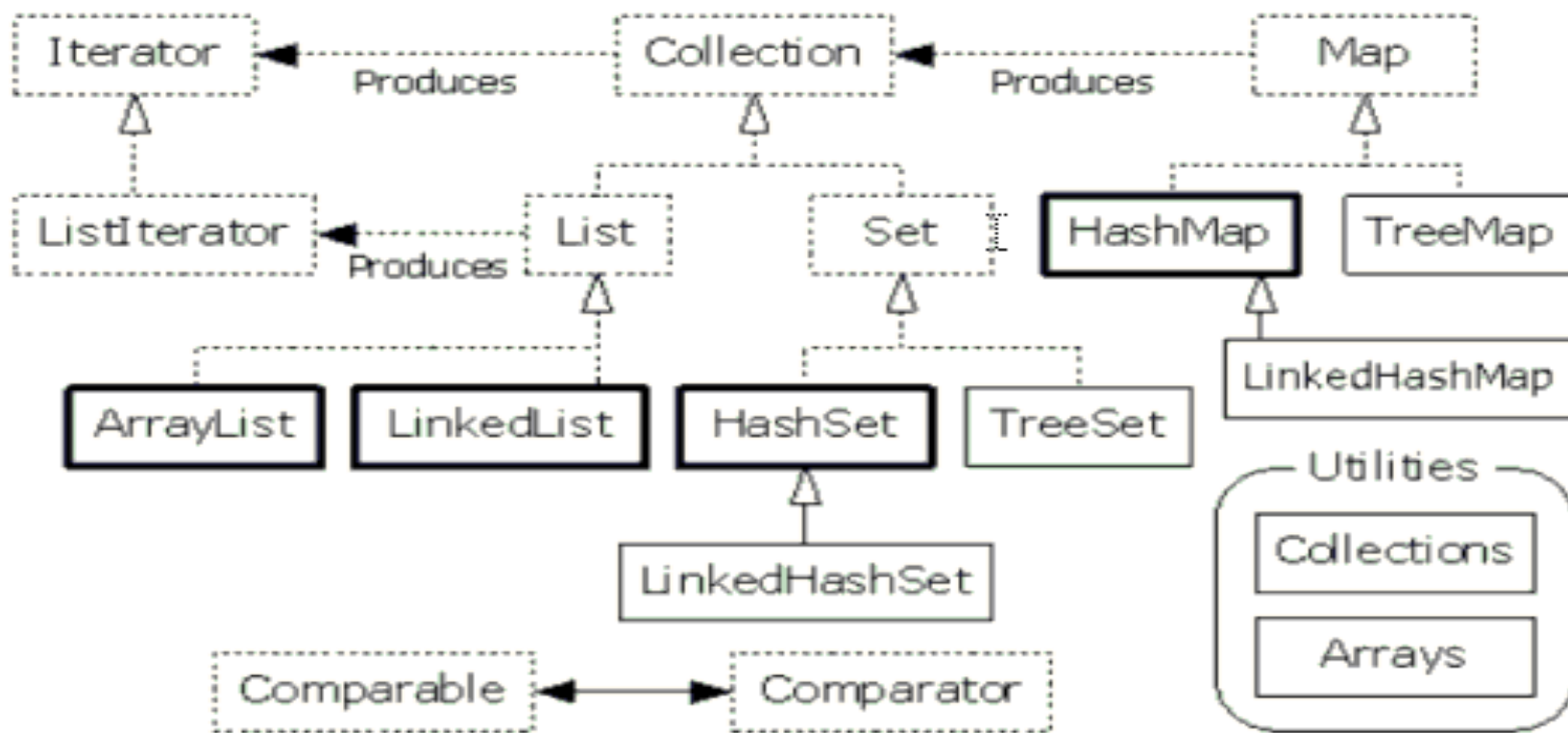
- JDK1.5以后，简化了定义方式。
 - `Integer x = new Integer(4);`可以直接写成
 - `Integer x = 4;`//自动装箱。
 - `x = x + 5;`//自动拆箱。通过`intValue`方法。
- 需要注意：
 - 在使用时，`Integer x = null;`上面的代码就会出现`NullPointerException`。

集合类

- 为什么出现集合类?
 - 面向对象语言对事物的体现都是以对象的形式，所以为了方便对多个对象的操作，就对对象进行存储，集合就是存储对象最常用的一种方式。
- 数组和集合类同是容器，有何不同?
 - 数组虽然也可以存储对象，但长度是固定的；集合长度是可变的。数组中可以存储基本数据类型，集合只能存储对象。
- 集合类的特点
 - 集合只用于存储对象，集合长度是可变的，集合可以存储不同类型的对象。

集合框架的构成及分类

java 中集合类的关系图



集合框架中的常用接口

Collection接口有两个子接口:

List（列表），**Set**（集）

List: 可存放重复元素，元素存取是有序的。

Set: 不可以存放重复元素，元素存取是无序的。

List接口中常用类

- **Vector**: 线程安全, 但速度慢, 已被 **ArrayList** 替代。
- **ArrayList**: 线程不安全, 查询速度快。
- **LinkedList**: 链表结构, 增删速度快。

取出List集合中元素的方式:

- **get(int index)**: 通过脚标获取元素。
- **iterator()**: 通过迭代方法获取迭代器对象。

迭代

- 迭代是取出集合中元素的一种方式。
- 因为Collection中有iterator方法，所以每一个子类集合对象都具备迭代器。
- 用法：

```
for(Iterator iter = iterator();iter.hasNext(); )   Iterator iter = l.iterator();
{                                                    while(iter.hasNext())
    System.out.println(iter.next());                {
}                                                    System.out.println(iter.next());
}                                                    }
```

迭代注意事项

- 迭代器在Collection接口中是通用的，它替代了Vector类中的Enumeration(枚举)。
- 迭代器的next方法是自动向下取元素，要避免出现NoSuchElementException。
- 迭代器的next方法返回值类型是Object，所以要记得类型转换。

思考：为什么next方法的返回类型是Object的呢？

Set接口中常用的类

- **HashSet:** 线程不安全，存取速度快。
它是如何保证元素唯一性的呢？
- **TreeSet:** 线程不安全，可以对**Set**集合中的元素进行排序。
它的排序是如何进行的呢？

Set集合元素唯一性原因

- **HashSet:** 通过equals方法和hashCode方法来保证元素的唯一性。
- **TreeSet:** 通过compareTo或者compare方法中的来保证元素的唯一性。元素是以二叉树的形式存放的。

泛型

- JDK1.5以后出现的机制
- 泛型出现的原因
- 泛型出现的好处
- 泛型的书写格式
- 自定义泛型类

泛型的特点

- 提高了程序的安全性
- 将运行期遇到的问题转移到了编译期
- 省去了类型强转的麻烦
- 泛型类的出现优化了程序设计

例程

Map集合

- Map与Collection不同
- Map集合存储于取出元素的方式
- Map集合的特点
- Map集合中常用类

Map与Collection

- Map与Collection在集合框架中属并列存在
- Map存储的是键值对
- Map存储元素使用put方法，Collection使用add方法
- Map集合没有直接取出元素的方法，而是先转成Set集合，在通过迭代获取元素
- Map集合中键要保证唯一性

Map集合常用类

- **Hashtable**: 线程安全, 速度慢, 不允许存放null键, null值, 已被HashMap替代。
- **HashMap**: 线程不安全, 速度快, 允许存放null键, null值。
- **TreeMap**: 对键进行排序, 排序原理与TreeSet相同。

例程

练习: 自定义一个可以同时存放三元素的Map集合。

集合框架中的工具类

- Collections

- 对集合进行查找
- 取出集合中的最大值，最小值
- 对List集合进行排序
-

- Arrays

- 将数组转成List集合
- 对数组进行排序
- 对数组进行二分查找

新增for语句

- Collection在JDK1.5后出现的父接口Iterable就是提供了这个for语句。
- 格式：
for(数据类型 变量名：数组或集合)
{
 执行语句;
}
- 简化了对数组，集合的遍历。

函数的另一种表现形式

- 返回值类型 函数名(参数类型... 形式参数)
{
 执行语句;
}
- 其实接收的是一个数组，可以指定实际参数个数。