第四周学习笔记

06月07日

一：集合的理解

1.简介

因为数组只能存放同一类型的数据，且长度固定在实际编程中会存在一些麻烦，所以集合应运而生；不存储对象而是用于存放对象的引用，所以可以以引用类型存放不同的类型且不限数量。Java 集合框架主要包括两种:储存元素的集合 Collocation 和储存键值对映射的图 Map。

2.List:

继承于 Collocation，属于集合的一种，是有序的队列，List 中每一个元素有一个索引，第一个索引值为0，List 中允许有重复的元素；

3.Set:

继承于 Collocation，是不包括重复元素的集合的一种，维持内部排序，所以随机访问没有意义；

4.Map:

是由一系列键值对组成的图,提供了键值 Key 到 Value 的映射，且该映射一一对应，不允许多对一，允许一对多；

二：Hashtable

1.简介

实现 Map 接口的类，以哈希表数据结构实现，每个键、值都是一个对象，采用拉链法实现哈希表，性能不如 HashMap;

三：TreeMap

1.简介

有序散列表，实现 SortedMap 接口，储存 K-V 键值对，底层通过红黑树实现，可以实现元素的自动排序。

四：HashMap

1. 简介

实现 Map 接口的类，以哈希表数据结构实现，通过哈希函数将元素的键值转化为索引，为快速查询而设计的；

五：三者的异同

1. 相同点

都对 Map 接口进行了实现；

2. 相异点

Hashtable 是线程安全的、同步的，不支持 null 和空值，同步导致开销大；HashMap 大体与 Hashtable 一致，但 HashMap 支持 null 和空值，不是同步的开销较小；treeMap 是基于红黑树实现的可以进行排序、顺序访问的 Map

六：IO 基本操作 ，可以找下 IO 文件操作的模板代码

1. 简介

JavaIO 包中有五个类（File、InputStream、OutPutStream、writer、reader）和一个接口，类中除了 File 之外全部属于抽象类，那些抽象类有很多复杂且麻烦的子类，且这些类在使用完成后需要手动关闭流。字符流只能用来复制纯文本文件，字节流可以复制所以类型的文件。

2. IO 基本操作

创建与指定路径文件相连的输入流，输出流，用于读写文件：

```
FileInputStream (File file||String name);
```

```
FileOutputStream (File file||String name);
```

创建缓冲输入流、输出流，用于缓存字节数据的字节流：

```
BufferedInputStream (InputStream in,null||int size);
```

```
BufferedOutputStream (InputStream in,null||int size);
```

创建读取、写入字符的字符流：

FileReader (File file||String name);

FileWriter (File file||String name);

创建文本缓冲输入流、输出流，用于缓存字符数据的字符流：

BufferedReader (Reader in,null||int size);

BufferedWriter (Reader in,null||int size);

字节流转字符流：

InputStreamReader (InputStream in, null||Charset cs);

字符流转字节流：

OutputStreamWriter(OutputStream out,null||Charset cs);

从流中读取下一个字节，返回字节值，若读到流尾返回-1：

Read (byte[] b,null||-int off,int len-);

将字节写入流中：

Read (byte[] b,null||-int off,int len-);

关闭流：

Close ();

刷新：

Flush ();

七：网络编程中 TCP 和 UDP 协议开发的基本流程，网上也有很多 Demo 代码

1.TCP（C/S 模式）：

TCP 通过连接的方式进行通信数据以流的形式传送,流程如下：

a. 创建服务器端，设定服务器端口和 ip 地址；

b. 创建客户端，设定服务器的端口和 ip 地址以及本地端口；

c. 客户端通过 Socket 对象将数据以流的形式在网络中传输；

d. 服务器端通过 ServerSocket 对象监听客户端的连接；

e. 服务器端通过 Socket 对象获取客户端发来服务器端通过 Socket 对象反馈信息回客户端；

f. 关闭套接字；

2.UDP（交互式）：

UDP 以无连接的方式进行通信数据以数据报的形式传送：

a. 创建 UdpSocket 用于发送接收数据；

b. 创建数据包 DatagramPacket 储存数据；

c. 调用 DatagramSocket 方法发送、接收数据包；

d. 若接收到报文则通过调用数据包的方法获取接收到的数据；

e. 关闭套接字；

八、语法知识：

1、集合中 add 方法的参数为什么是 Object？（14-2）

答：JDK1.5后引入的泛型 Object 的概念，在此处为了贴合集合设计的理念，设计参数 Object 以便于接收任意类型的对象

2、set 和 list 的区别？（毕14-4）

答：List 支持有序的插入重复对象，set 插入的对象是无序且不重复的

3、ArrayList 和 LinkedList，Vector 的区别？（毕14-6）

答：ArrayList 与 Vecter 相比 Vector 是线程安全的，两种都有查询速度快、增删改慢的特点；

ArrayList 与 LinkedList 相比 LinkedList 具有增删改快、查慢的特点，两者都是非线程安全的。

4、ArrayList，HashSet,TreeSet 他们判断元素是否相同或者数保证元素唯一性的依据（或者说底层会调用什么方法）是什么？（毕14-14，15-3）

答：ArrayList 底层调用 contians（）方法实现 equals（）方法直接判断两个元素是否相同；
HashSet 根据哈希地址和实现 equals()方法递进式的判断两个元素是否具有相同的地址，又是否具有相同的属性；
TreeSet 通过实现 comparable 接口，使其元素具有可比性，所以具体实现是通过覆盖、重写 compareto（）方法直接判断元素的属性是否相同。

5、如何避免迭代器需要进行强制类型转换？（15-7）

答：通过泛型。

6、<? extends E>和<? super E> 的区别？（毕15-13）

答：前者用于方法返回参数必须是 E 的子类型，限定了参数类型的上界；
后者用于限定方法入参，参数类型只能是 E 的超类型，限定了参数类型的下界。

7、Map 集合和 Collection 集合的区别？（毕16-1）

答：Map 储存键值对，Collection 存放对象的引用。

8、keySet();和 entrySet();的区别？（毕16-4,16-5）

答：keySet 返回 map 所有键的 Set 集合，entrySet 返回 map 集合中的映射关系，所以 KeySet 遍历 map 的性能不如 entrySet。

9、Collection 和 Collections 的区别？（毕17-10）

答：前者是一个接口，为各种具体的集合提供同一操作方式；后者是一个工具类，提供一系列静态方法用于对集合中元素进行排序、搜索以及线程安全等各种操作。

10、下面语句的输出结果？（毕17-12）

```
List<String> list = new ArrayList<>();
list.add("aa");
list.add("kjaljd");
list.add("alsd");
list.add("akjka");
int x = Collections.binarySearch(list,"alsd");
System.out.println("x = "  + x);
```

答：x = -2

11、Collections 中 reverse 和 reverseOrder 的区别？（17-14）

答：前者反转一个 List 或 ArrayList 类；后者返回一个比较器，强行逆转实现了 Comparable 接口的对象 collection 的自然顺序。

12、Arrays 中 asList 方法将数组变成 list 集合后，为什么不能进行增删操作。（17-16）

答：该 list 集合是 Arrays 里面的一个静态内部类，非 ArraryList 类，且该类中无对应方法实现增删操作。

13、传统 for 循环与增强 for 循环的区别？（毕17-18）

答：增强型 for 必须有遍历的目标，传统 for 可以定义下标。

14、写出下面代码的输出结果？（毕17-18）

```
List<String> list = new ArrayList<>();
  list.iterator();

  list.add("java01");
  list.add("java02");
  list.add("java03");
  for(String s : list){
     s = "kk";
  }
  System.out.println(list);
```

答：[java01,java02,java03]

15、写出下面代码的输出结果（毕17-19）

```
show("a","b","c");
public static void show(String ss,String... s){
    for(int x =0;x < s.length;x++){
      System.out.println(s[x]);
    }
}
```

答：b
c

16、写出下面代码的输出结果（毕18-05）

```
System.out.println(Math.ceil(-17.20));
```

答：-17.0

17、下面两句代码的不同？（毕18-19）

（1）FileWriter fw = new FileWriter(“demo.txt”);

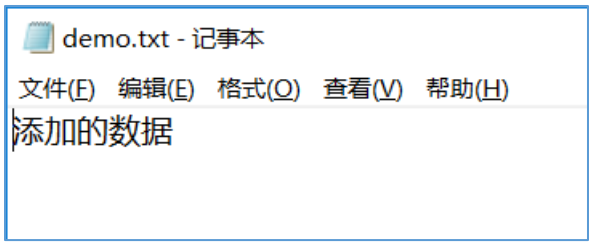
（2）FileWriter fw = new FileWriter(“demo.txt”, true);

答：第一条代码实现：删除原文件中的数据然后重头开始写；
第二条代码实现：从原文件的数据末尾处开始追加数据。

18、下面代码的输出结果不考虑异常声明（毕18-11）

```
FileReader fr = new FileReader("demo.txt");
    char[] arr = new char[2];
    fr.read(arr);
    System.out.print(new String(arr));
    fr.read(arr);
    System.out.print(new String(arr));
    fr.read(arr);
    System.out.print(new String(arr));
```

“demo.txt”中内容



答：添加的数据数

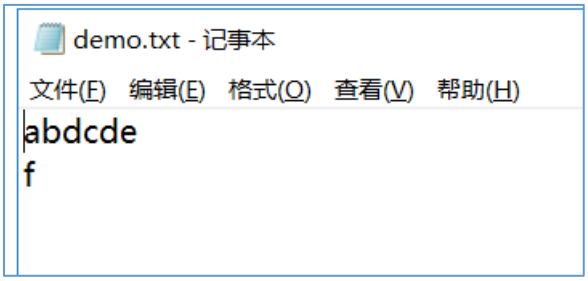
19、fileWriter.close(); 和 fileOutputStream.close(); 的区别是什么？（毕19-11）

答：前者字符流使用字符数组，后者字节流使用字节数组。

20、下面代码的运行结果是什么？不考虑异常声明(毕19-11)

```
FileInputStream fis = new FileInputStream("demo.txt");
System.out.println("len = " + fis.available());
fis.close();
```

演示文件：



答：len = 9

21、OutStreamWriter 是谁通向谁的桥梁？OutputStreamWriter 属于字节流体系还是字符流体系？(毕19-17)

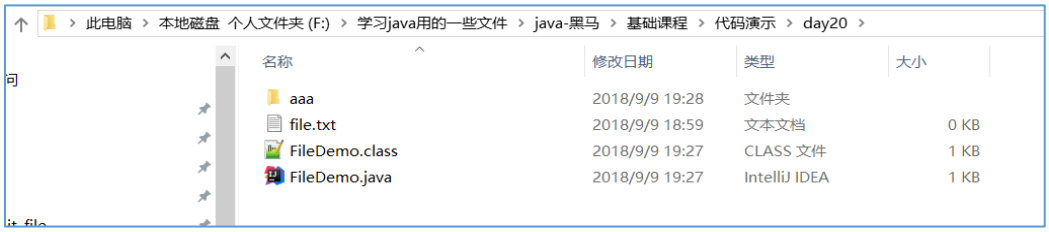
答：OutputStreamWriter 是从字符流到字节流的桥梁，属于字符流体系。

22、File 对象创建文件，和读取流对象创建文件有什么不同？(毕20-02)

答：前者在指定位置创建，如果文件已经存在则不创建，返回 false；后者一定建立文件，若文件已经存在则覆盖文件。

23、下面语句的输出结果，不考虑异常（毕20-3）

```
File file = new File("aaa\\bbb"); // 该文件所在目录如下图
System.out.println("mkdir 结果" + file.mkdirs());
```



答：mkdir 结果 true

24、ObjectOutputStream 中 write 和 writeInt。当传入的参数都是 a(int 类型)时，他们的区别？（毕21-01）

答：前者写入一个字符型数据；后者写入一个32位的数。

25、在对象流中，static 修饰的基本数据，为什么不能写入文件中（毕21-1）

答：static 修饰的基本数据不能被序列化。

26、TCP 和 UDP 的区别？

答：TCP 是面向连接提供可靠的服务，传输的数据不会丢失，以流的形式持续传输数据的点对点全双工可靠通信协议；

UDP 是面向无连接的提供尽最大努力交付的服务，以报文的形式传输数据的支持多对多、一对多等交互式通信的协议。

27、正则表达式的优缺点？（毕25-3）

答：优点是可以简化对字符串的复杂操作；缺点是当符号定义过多时正则很长，可读性差

28、正则匹配，组中“\n”和“$n”的区别？（毕25-4）

答：前者本质为字符，只在匹配时用于组的反向引用；后者是全局变量，只要发生正则匹配 Ruby 就会去更新。

29、Matcher 对象中 find 和 matches 方法有什么区别？（毕25-6）

答：前者是部分匹配，寻找匹配的正则表达式的子字符串序列；后者是完全匹配，只有整个字符串都匹配该正则表达式时才返回 true。