、

容器作业

一、填空题

1. Java 集合框架提供了一套性能优良、使用方便的接口和类，包括 Collection 和 Map 两大类，它们都位于 java.util 包中
2. 队列和堆栈有些相似，不同之处在于栈是先进后出，队列是先进先出 。
3. 结构是一种由多个节点组成的线性数据结构，并且每个节点包含有数据以及指向下一个节点的引用。
4. __List___是一种集合类，它采用链表作为的存储结构，便于删除和添加元素，但是按照索引查询元素效率低下。
5. TreeSet 是一种 Collection 类型的集合类，其中元素唯一，并采用二叉树作为存储结构，元素按照自然顺序排列。
6. 如果希望将自定义类 Student 的多个对象放入集合 TreeSet，实现所有元素按照某个属性的自然顺序排列，则需要 Student 类实现____Comparable____接口。
7. 在 Java 中 Map 集合的访问时间接近稳定，它是一种键值对映射的数据结构。这个数据结构是通过数组来实现的。
8. 迭代器 Iterator 为集合而生，专门实现集合遍历，该接口有三个方法，分别是 hasNext() 、
   ___next()___、remove()。

二、选择题

1. 以下选项中关于 **Java** 集合的说法错误的是（ AC）。（选择二项）

A. List 接口和 Set 接口是 Collections 接口有两个子接口

B. List 接口中存放的元素具有有序，不唯一的特点

C. Set 接口中存放的元素具有无序，不唯一的特点

D. Map 接口存放的是映射信息，每个元素都是一个键值对

2. 如下 **Java** 代码，输出的运行结果是（ A）。（选择一项） **public class**

```
Test {
    public static void main(String[ ] args) {
        List<String> list=new
        ArrayList<String>(); list.add("str1");
        list.add(2, "str2");
        String s=list.get(1);
        System.out.println(s);
    }
}
```

A 运行时出现异常

B 正确运行，输出 str1

C 正确运行，输出 str2

D 编译时出现异常

3. 以下 **Java** 代码的作用是首先将一个数组的内容存入集合，然后判断集合中是否有指定的元素存在，其中共有（D）处错误。（选择一项）

```
import java.util.List; public class Test
{ public int getIndexofArray(float[]
f){ int rtn=-1; float objf=3.4; List
list=null; for(int
i=0;i<f.size( );i++){ list.add(f[i]);
        }
        for(int
            i=0;i<list.size( );i++){ floa
            t tmp=(float)list.get(i);
            if(objf==tmp){ rtn=i;
            }
        }   return
        rtn;
    }
}
```

A 0

B. 1

C. 2

D. 3

4. 分析如下 **Java** 代码，编译运行后将输出（B）。（选择一项）

```
public class Test { public Test() {
    }
    static void print(List<Integer> al)
        { ox123 al.add(2); al = new
```

```
        ArrayList<Integer>(); ox234
        al.add(3); al.add(4);
    }
    public static void main(String[] args)
        { List<Integer> al = new ArrayList<Integer>();
        al.add(1);
        print(al); //ox123 把 al 集合得地址 传递走了
        System.out.println(al.get(1)); ox123
    }
}
```

A 1

B. 2

C. 3

D. 4

5. 在 **Java** 中,下列集合类型可以存储无序、不重复的数据的是（D）。（选择一项）

A ArrayList

B LinkedList

C TreeSet

D HashSet

6. 以下代码的执行结果是（C）。（选择一项）

```
Set<String> s=new HashSet<String>();
s.add("abc");
s.add("abc");
s.add("abcd");
s.add("ABC");
System.out.println(s.size());
```

A 1 **B.** 2 **C.**3 **D.**4

7. 给定如下 **Java** 代码，编译运行的结果是（C）。（选择一项）

```
public class Test { public static void
    main(String[] args) {
      Map<String, String> map = new HashMap<String, String>();
        String s = "code";
        map.put(s, "1");
        map.put(s, "2");
        System.out.println(map.size());
    }
}
```

A 编译时发生错误 **B.** 运行时引发异常

C. 正确运行，输出：1

D. 正确运行，输出：2

8. 下面集合类中属于非线程安全，且结构采用了哈希表的是（C）。（选择一项）

A. Vector **B.** ArrayList **C.** HashMap

D. Hashtable

9. 在 **Java** 中，**LinkedList** 类与 **ArrayList** 类同属于集合框架类，下列（D）选项中是 **LinkedList** 类有而 **ArrayList** 类没有的方法。（选择两项） **A** add(Object o)

B. add(int index,Object o)

C. getFirst()

D. removeLast()

三、 判断题

1. 数组和集合中的元素可以是任何数据类型，包括基本类型和引用类型。(F)
2. 容器指的是“可以容纳其他对象的对象”。（T）
3. Java 集合中的 Set 接口和 List 接口都是从 Collection 接口派生出来的。（T）
4. Collection 接口存储一组不唯一，有序的对象，它有两个子接口：List 和 Set。（F）

5. Collection 是 Java 集合顶级接口，其中的元素无序，唯一。Java 平台不提供这个接口任何直接的实现。（F）
6. List 是有序的 Collection，使用此接口能够精确的控制每个元素插入的位置。用户能够使用索引来访问 List 中的无素，这类似于 Java 的数组。（T）
7. HashSet 采用哈希表存储结构，特点是查询速度快，但是其中元素无序排列。（T）
8. LinkedHashMap 是一种有序的 HashMap，查询速度快，便于添加删除操作。（F）
9. 基本数据类型的值可以被直接存储在 Vector 对象中。（T）
10. 泛型是 JavaSE1.7 的新特性，泛型的本质是参数化类型，也就是说所操作的数据类型被指定为一个参数。Java 语言引入泛型的好处是安全简单。（F）
11. Collection 是专门操作集合的工具类，提供一系列静态方法实现对各种集合操作。（F）
12. Iterator 接口可以遍历任何 Collection 接口的实现类，可以从一个 Collection 中使用 iterator( ) 方法来获取迭代器实例。迭代器取代了 Java 集合框架中的 Enumeration。（T）
13. 采用增强 for 循环遍历 List 或者 Set，如果 List 或者 Set 没有加泛型，也能遍历。（T）
14. 在类已经重写 equals 和 hashCode 方法的前提下，equals 返回 true，hashcode 一定相等。（T）

四、 简答题

1. 集合和数组的比较

集合可变，数组一旦声明长度就固定了

2. 简述 List、Set、Collection、Map 的区别和联系。

Collection 有 List 和 Set2 个子接口，List 通过索引的链表结构，是可以重复的，Set 的值不可以重复，Map 是通过键值对来存储的

3. ArrayList 和 LinkedList 的区别和联系。它们的底层分别是用什么实现的?

ArrayList 的查询效率不高，删改程度低，线程不安全；LinkList 的查询效率高，删改程度高，线程不安全。ArrayList 的底层是由数组来实现，LinkList 是由双链表进行实现。

4.HashSet 采用了哈希表作为存储结构，请说明哈希表的特点和实现原理。

提示：结合 Object 类的 hashCode()和 equals()说明其原理

HashSet：equals 返回 true，hashCode 返回相同的整数；哈希表；存储的数据是无序的。成员可为任意 Object 子类的对象，但如果覆盖了 equals 方法，同时注意修改 hashCode 方法。

HashMap 键成员可为任意 Object 子类的对象，但如果覆盖了 equals 方法，同时注意修改 hashCode 方法。

5.你简述 HashMap 和 Hashtable 的区别?

Hashtable：Synchronize;在多线程并发的情况下，能够直接使用 Hashtable，不要自己为它的方法实现同步

HashMap：在缺省情况下是非 Synchronize 的;使用 HashMap 的时候就需要自己增加同步处理;HashMap 是线程不安全的

6.说明 isEmpty 的作用，并说明下面代码有问题吗?

```
Collection c = null;
System.out.println(c.isEmpty());
```

isEmpty 判断集合是否为空，c 没有任何集合会出现空指针异常

7.写出 List、Set、Map 中使用泛型的例子。

List<T>、Set<T>、Map<T>.

8.使用泛型有什么好处?

使用泛型遍历集合中，不需要进行数据类型的强制转换。提高了代码的可读性和安全性。

9.每个对象都有一个哈希码吗？哈希码是根据什么生成的？会不会重复?

是的，因为每个对象的内存地址不一样，所以哈希吗也不一样，hashcode 根据内存生成，不会重复

五、 编码题

1. 使用 List 和 Map 存放多个图书信息，遍历并输出。其中商品属性：编号，名称，单价，出版社；使用商品编号作为 Map 中的 key。

```java
package Homeworks.HomeworkDay11;

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
21 个用法  新 *
public class Book implements Comparable<Book>{
    6 个用法
    private Integer id;
    4 个用法
    private String name;
    4 个用法
    private double price;
    4 个用法
    private String publishing;

    6 个用法  新 *
    public Book(Integer id, String name, double price, String publishing) {
        this.id = id;
        this.name = name;
        this.price = price;
        this.publishing = publishing;
    }

    3 个用法  新 *
    public Integer getId() { return id; }

    0 个用法  新 *
    public void setId(Integer id) { this.id = id; }

    新 *
    public String getName() { return name; }
```

```
    public void setId(Integer id) { this.id = id; }

    新 *
    public String getName() { return name; }

    新 *
    public void setName(String name) { this.name = name; }

    0 个用法  新 *
    public double getPrice() { return price; }

    0 个用法  新 *
    public void setPrice(double price) { this.price = price; }

    0 个用法  新 *
    public String getPublishing() { return publishing; }

    0 个用法  新 *
    public void setPublishing(String publishing) { this.publishing = publishing; }

    新 *
    @Override
    public String toString() {
        return "Book{" +
                "id='" + id + '\'' +
                ", name='" + name + '\'' +
                ", price=" + price +
                ", publishing='" + publishing + '\'' +
                '}';
    }

    新 *
    @Override
    public int compareTo(Book o) { return this.id-o.id; }
}
```

```java
package Homeworks.HomeworkDay11;

import ...

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
0 个用法  新 *
public class Demo01 {
    新 *
    public static void main(String[] args) {
        List<Book> list = new ArrayList<>();
        Book b1 =new Book( id: 1, name: "天书奇谈", price: 20.0, publishing: "北方出版社");
        Book b2 =new Book( id: 2, name: "红楼梦", price: 50.0, publishing: "南方出版社");
        Book b3 =new Book( id: 3, name: "西游记", price: 250.0, publishing: "北方出版社");
        list.add(b1);
        list.add(b2);
        list.add(b3);
        Map<Integer,Book> map = new HashMap<>();
        map.put(b1.getId(),b1);
        map.put(b2.getId(),b2);
        map.put(b3.getId(),b3);
        for (Map.Entry<Integer, Book> stringBookEntry : map.entrySet()) {
            System.out.println(stringBookEntry.toString());
        }

    }
}
```

2. 使用 HashSet 和 TreeSet 存储多个商品信息，遍历并输出；其中商品属性：编号，名称，单价，出版社；要求向其中添加多个相同的商品，验证集合中元素的唯一性。

   提示：向 HashSet 中添加自定义类的对象信息，需要重写 hashCode 和 equals( )

   向 TreeSet 中添加自定义类的对象信息，需要实现 Comparable 接口，指定比较规则

```java
package Homeworks.HomeworkDay11;

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
21 个用法  新 *
public class Book implements Comparable<Book>{
    6 个用法
    private Integer id;
    4 个用法
    private String name;
    4 个用法
    private double price;
    4 个用法
    private String publishing;

    6 个用法  新 *
    public Book(Integer id, String name, double price, String publishing) {
        this.id = id;
        this.name = name;
        this.price = price;
        this.publishing = publishing;
    }

    3 个用法  新 *
    public Integer getId() { return id; }

    0 个用法  新 *
    public void setId(Integer id) { this.id = id; }

    新 *
    public String getName() { return name; }
```

```java
public class Demo01 {
    新 *
    public static void main(String[] args) {
        List<Book> list = new ArrayList<>();
        Book b1 =new Book( id: 1, name: "天书奇谈", price: 20.0, publishing: "北方出版社");
        Book b2 =new Book( id: 2, name: "红楼梦", price: 50.0, publishing: "南方出版社");
        Book b3 =new Book( id: 3, name: "西游记", price: 250.0, publishing: "北方出版社");
        list.add(b1);
        list.add(b2);
        list.add(b3);
        Map<Integer,Book> map = new HashMap<>();
        map.put(b1.getId(),b1);
        map.put(b2.getId(),b2);
        map.put(b3.getId(),b3);
        for (Map.Entry<Integer, Book> stringBookEntry : map.entrySet()) {
            System.out.println(stringBookEntry.toString());
        }

    }
}
```

3. 实现 List 和 Map 数据的转换。具体要求如下：

功能 1：定义方法 public void listToMap( )将 List 中 Student 元素封装到 Map 中

1) 使用构造方法 Student(int id,String name,int age,String sex)创建多个学生信息并加入 List
2) 遍历 List，输出每个 Student 信息
3) 将 List 中数据放入 Map，使用 Student 的 id 属性作为 key，使用 Student 对象信息作为 value
4) 遍历 Map，输出每个 Entry 的 key 和 value

功能 2：定义方法 public void mapToList( )将 Map 中 Value 值 Student 信息封装到

List

1) 创建实体类 StudentEntry，可以存储 Map 中每个 Entry 的信息
2) 使用构造方法 Student(int id,String name,int age,String sex)创建多个学生信息，并使用 Student 的 id 属性作为 key，存入 Map
3) 创建 List 对象，每个元素类型是 StudentEntry
4) 将 Map 中每个 Entry 信息放入 List 对象

功能 3：说明 Comparable 接口的作用，并通过分数来对学生进行排序。

```
    public void setAge(int age) { this.age = age; }

    0 个用法  新 *
    public String getSex() { return sex; }

    0 个用法  新 *
    public void setSex(String sex) { this.sex = sex; }

    0 个用法  新 *
    public double getScore() { return score; }

    新 *
    @Override
    public String toString() {
        return "Student{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", age=" + age +
                ", sex='" + sex + '\'' +
                ", score=" + score +
                '}';
    }

    0 个用法  新 *
    public void setScore(int score) { this.score = score; }

    新 *
    @Override
    public int compareTo(Object o) {
        Student s = (Student)o;
        return (int)(this.score - s.score);
    }
}
```

```java
package Homeworks.HomeworkDay11;

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
5 个用法  新 *
public class StudentEntry {
    2 个用法
    private int key;
    2 个用法
    private Student stu;
    1 个用法  新 *
    public int getKey() { return key; }
    1 个用法  新 *
    public void setKey(int key) { this.key = key; }
    1 个用法  新 *
    public Student getStu() { return stu; }
    1 个用法  新 *
    public void setStu(Student stu) { this.stu = stu; }
}
```

```java
package Homeworks.HomeworkDay11;

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */

24 个用法  新 *
public class Student implements Comparable{
    4 个用法
    private int id;
    4 个用法
    private String name;
    4 个用法
    private int age;
    4 个用法
    private String sex;
    6 个用法
    private double score;

    6 个用法  新 *
    public Student(int id, String name, int age, String sex, double score) {
        this.id = id;
        this.name = name;
        this.age = age;
        this.sex = sex;
        this.score = score;
    }

    3 个用法  新 *
    public int getId() { return id; }

    0 个用法  新 *
    public void setId(int id) { this.id = id; }
```

```java
package Homeworks.HomeworkDay11;

import ...

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
1个用法  新 *
public class ListToMap {
    1个用法  新 *
    public static void listToMap(){
        Student s1 = new Student( id: 1, name: "小红", age: 17, sex: "男", score: 10);
        Student s2 = new Student( id: 3, name: "小航", age: 15, sex: "女", score: 20);
        Student s3 = new Student( id: 2, name: "肖子", age: 13, sex: "女", score: 15);
        List<Student> list = new ArrayList<>();
        list.add(s1);
        list.add(s2);
        list.add(s3);
        for (Student student : list) {
            System.out.println(student.toString());
        }
        Map<Integer,Student> map = new HashMap<>();
        map.put(1,s1);
        map.put(2,s2);
        map.put(3,s3);
        for (Map.Entry<Integer, Student> integerStudentEntry : map.entrySet()) {
            System.out.println(integerStudentEntry.toString());
        }
    }
}
```

```java
package Homeworks.HomeworkDay11;

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
0 个用法  新 *
public class Demo03 {
    新 *
    public static void main(String[] args) {
        ListToMap.listToMap();
        MapToList.mapToList();

    }
}
```

```java
import ...

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
1个用法  新 *
public class MapToList {
    1个用法  新 *
    public static void mapToList(){
        Student stu1 = new Student( id: 110,  name: "小红",  age: 23,  sex: "女", score: 10.2);
        Student stu2 = new Student( id: 111,  name: "小强",  age: 21,  sex: "男", score: 18.67);
        Student stu3 = new Student( id: 112,  name: "小花",  age: 12,  sex: "女", score: 17.17);

        Map<Integer, Student> map = new HashMap<~>();
        map.put(stu1.getId(), stu1);
        map.put(stu2.getId(), stu2);
        map.put(stu2.getId(), stu3);
        List<StudentEntry> list = new ArrayList<~>();
        for (Map.Entry<Integer, Student> entry : map.entrySet()) {
            StudentEntry studentEntry = new StudentEntry();
            studentEntry.setKey(entry.getKey());
            studentEntry.setStu(entry.getValue());
            list.add(studentEntry);
        }

        for (StudentEntry se : list) {
            System.out.println(se.getKey() + "\t" + se.getStu());
        }

    }
}
```

4. 用代码写出遍历 List 的三种方式。

```java
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
0 个用法  新 *
public class Demo04 {
    新 *
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        for (int i = 0; i < list.size(); i++) {
            list.get(i);
        }
        for (Integer integer : list) {
            System.out.println(integer);
        }
        Iterator<Integer> integerIterator = list.iterator();
        while (integerIterator.hasNext()){
            System.out.println(integerIterator.next().toString());
        }
    }
}
```

5. 用代码写出遍历 Set 的两种方式。

```java
package Homeworks.HomeworkDay11;

import ...

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
0 个用法  新 *
public class Demo05 {
    新 *
    public static void main(String[] args) {
        Set<Integer> set = new HashSet<>();
        set.add(1);
        set.add(2);
        set.add(3);
        set.add(4);
        for (Integer integer : set) {
            System.out.println(integer);
        }

        Iterator<Integer> integerIterator = set.iterator();
        while (integerIterator.hasNext()){
            System.out.println(integerIterator.next().intValue());
        }
    }
}
```

6. 用代码写出遍历 map 的方式。

```java
package Homeworks.HomeworkDay11;

import ...

/**
 * @author junhaocai
 * @email junhaocai01@gmail.com
 * @date 2023/1/12
 */
0 个用法  新 *
public class Demo06 {
    新 *
    public static void main(String[] args) {
        Map<Integer,String> map = new HashMap<>();
        map.put(1,"1");
        map.put(2,"2");
        map.put(3,"3");
        map.put(4,"4");
        for (Map.Entry<Integer, String> integerStringEntry : map.entrySet()) {
            System.out.println(integerStringEntry.toString());
        }
    }
}
```

六、 可选题

1. 假如有以下 email 数据“aa@sohu.com,bb@163.com,cc@sina.com,..”现需要把 email 中的用户部分和邮件地址部分分离，分离后以键值对应的方式放入 HashMap?

2. 由控制台按照固定格式输入学生信息，包括学号，姓名，年龄信息，当输入的内容为 exit 退出；将输入的学生信息分别封装到一个 Student 对象中，再将每个 Student 对象加入到一个集合中，要求集合中的元素按照年龄大小正序排序；最后遍历集合，将集合中学生信息写入到记事本，每个学生数据占单独一行。
   推荐步骤：
   a) 创建 Student 类，并指定按照年龄正序排列
   b) 通过控制台输入多个不同 Student 信息。格式规定为：编号#姓名#年龄 c) 取出字符串中相应信息放入 Student 对象，并将 Student 加入到集合中

   d) 遍历集合的过程中将学生的信息输入到记事本难点：
   e) 如何指定学生按照年龄正序排列
   f) 如果从字符串“编号#姓名#年龄”中提取学生信息
   g) 放入哪种集合后可以保证学生按照年龄大小正序排列
   h) 如何将集合中学生信息写入记事本，每个学生数据占单独一行

3. 针对 List 中新增的有关顺序的方法，如 add(int index, E element) , get(int index)等每

   个都进行测试。并且使用 debug 来帮助我们理解程序运行。

4. Collection 和 Collections 有什么区别?
5. Map 中，key 能否重复?如果重复，会有什么现象?
6. 请你简述 Set 和 List 的特点跟区别?