测试题1

一、选择题:每题3分,共45分

CD1. List和Set集合的上层接口是(    )

A. java.util.Map  B. java.util.Collection   C. java.util.List     D. java.util.Set

A2. ArrayList类的底层数据结构是(   )

A. 数组结构  B. 链表结构  C.哈希表结构  D.红黑树结构

A3.LinkedList类的特点是(  )

A.查询快  B. 增删快    C.元素不重复  D. 元素自然排序

A4. 以下程序将ArrayList中的数据取出，转换后存入整形数组，程序最有可能报告何种异常。（ ）

public class Test1{

public static void main(String[] args){

List list = new ArrayList();

list.add(3.14);

list.add(4.13);

list.add(1.43);

int[] arr = new int[4];

for(int i = 0;i<list.size();i++){

arr[i]=(Integer)list.get(i);

}}}

A. ArrayIndexOutOfBoundsException

B. ArithmeticException

C. NullPointerException

D. ClassCastException

D5. (单选题)下列代码的输出结果是：（）。

public static void main(String[] args) {   
 String str = "LEA";

Deque<Character> stack = new LinkedList<Character>();

for (int i = 0; i < str.length(); i++) {

stack.push(str.charAt(i));

}

stack.pop();

stack.pop();

stack.pop();

System.out.println(stack.peek());

}

A.L B.E C.A D.null

D6. (单选题)请看下列代码的输出结果是：（D）。

public static void main(String[] args) {

List<Integer> list = new ArrayList<Integer>();

for (int i = 0; i < 10; i++) {

list.add(i);

}

List<Integer> subList = list.subList(4, 9);

for (int i = 0; i < subList.size(); i++) {

subList.set(i, subList.get(i) \* 10);

}

System.out.println(list);

}

A.[0, 1, 2, 3, 40, 50, 60, 70, 80, 90]

B.[0, 1, 2, 3, 4, 50, 60, 70, 80, 90]

C.[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

D.[0, 1, 2, 3, 40, 50, 60, 70, 80, 9]

A7.(单选题)下面的代码用于输出字符数组ch中每个字符出现的次数，应该填入的代码是（ ）。

public static void main(String[] args) {

char[] ch = { 'a', 'c', 'a', 'b', 'c', 'b' };

HashMap map = new HashMap();

for (int i = 0; i < ch.length; i++) {

//应该填入

}

System.out.println(map);

}

A.if (map.contains(ch[i])) {

map.put(ch[i], map.get(ch[i]) + 1);

}else{

map.put(ch[i], 1);

}

B.if (map.contains(ch[i])) {

map.put(ch[i], (Integer) map.get(ch[i]) + 1);

} else {

map.put(ch[i], 1);

}

C.if (map.containsValue(ch[i])) {

map.put(ch[i], map.get(ch[i]) + 1);

} else {

map.put(ch[i], 1);

}

D.if (map.containsKey(ch[i])) {

map.put(ch[i], map.get(ch[i]) + 1);

} else {

map.put(ch[i], 1);

}

C8.(单选题)阅读代码：

public class Cell implemens Comparable<Cell>{

int row;

int col;

//getter()和setter()

//code here

}

//code here处，需要添加的代码段是（）。

A.public void compareTo(Cell o){

this.row - o.row;

}

B.public boolean compareTo(Cell o){

return this.row　＞　o.row;

}

C.public int compareTo(Cell o){

return this.row - o.row;

}

D.public int compareTo(Object o){

return this.row - o.row;

}

B9.(单选题)请看以下代码：

Map<String,Double> map=new HashMap<String,Double>();

map.put("tom",123.6);

map.put("jim",234.5);

map.put("jerry",45.3);

《插入代码》

《插入代码》处要实现的功能是把key为"jim"的value值在原有数字的基础上加100

正确的是（ ）。

A.map.put("jim",map.get("jim")+100);

B.map.set("jim",map.get("jim")+100);

C.map.put("jim",234.5);

D.map.set("jim",234.5);

C10.(单选题)下列代码的运行结果是（ ）。

public class TestException {

static void fun() throws Exception {

try {

System.out.print("try1 ");

throw new Exception();

} catch (Exception e) {

System.out.print("exception1 ");

throw new Exception(e);

}

}

public static void main(String[] args){

try {

fun();

} catch (Exception e) {

System.out.print("exception2 ");

}

System.out.print("finish ");

}

}

A.try1 exception2 finish

B.try1 exception1 exception2

C.try1 exception1 exception2 finish

D.try1 exception1 finish

D11.(单选题)运行下面程序:

public class Test01{

public static void main(String[] args) {

try {

test();

System.out.println("1...");

} catch (ArithmeticException e) {

System.out.println("2...");

} catch (Exception e) {

System.out.println("3...");

} finally {

System.out.println("end...");

}

}

public static void test() {

String str = "cc";

str.compareTo("abc");

}

}

输出的结果是：（）。

A.1...

end...

B.2...

end...

C.2...

3...

end...

D.1...

2...

3...

end...

A12.(多选题)下列选项中，对Object类描述正确的是（ ）。

A.Java中所有的类，都默认自动继承了Object类

B.Java中所有的类都从Object类中继承了toString()方法、hashCode()方法和equals()等方法

C.Object类中toString()方法的返回值是一个字符串，equals方法的返回值为1或0

D.定义类的时候，类中必须重写Object类中的toString(),equals(),hashCode()方法

D13.(多选题)关于集合框架，说法正确的是（ ）。

A.与数组不同，集合框架是容量可变的容器

B.集合框架不能保存基本类型，但是可以将它们转换为包装类后保存

C.集合框架中的List, Set, Map接口都是Collection接口的子接口

D.集合类除了提供可变容量的存储外，还提供了对数据的操作方法

14.(多选题)下列能够正确的创建一个线程并启动线程的是（ ）。

A.class Test01 {

public static void main(String[] args)throws Exception {

Thread t1 = new Thread1();

t1.run();

}

}

class Thread1 extends Thread{

public void run() {

System.out.println("haha");

}

}

B.class Test01 {

public static void main(String[] args)throws Exception {

Thread t1 = new Thread1();

t1.start();

}

}

class Thread1 extends Thread{

public void run() {

System.out.println("haha");

}

}

C.class Test01 {

public static void main(String[] args)throws Exception {

Thread t = new Thread(new Thread1());

t.start();

}

}

class Thread1 implements Runnable{

public void run() {

System.out.println("haha");

}

}

D.class Test01 {

public static void main(String[] args)throws Exception {

Thread t = new Thread(new Thread1());

t.run();

}

}

class Thread1 implements Runnable{

public void run() {

System.out.println("haha");

}

}

B15.(多选题)对下列运算结果，判断正确的是（ ）；

A."1az098".matches("\\d[a-z]{2,8}[0-9]+")结果为true

B."1az098".matches("\\d[a-z]{2,8}[0-9]+")结果为false

C."张三,,,李四,,王五,,,,,,马六,,".split("[,]+").length == 1;该表达式结果返回true

D."张三,,,李四,,王五,,,,,,马六,,".split("[,]+").length == 4;该表达式结果返回true

二、程序题（共55分）

第一大题：（共40分）

1. 创建员工类Emp，类中的成员有：（10分）

私有属性： String name， int age, String gender ，double salary

无参构造方法，全参数构造方法

每个属性的set和get方法

重写toString方法，返回属性拼接成的字符串，格式为："name=张三，age=22，gender=男,salary=5000.0"

重写equals方法，要求若2个员工对象name相同，则对象equals比较的结果为true

实现Comparable接口，定义Emp之间默认的排序规则为：按照年龄降序排列

2. 创建SortDemo类，在main方法中完成以下操作：--（15分）

要求用户从控制台输入若干员工信息，格式为：

tom,22,男,5000.0;amy,33,女,5500;...... 每个用户之间用;隔开，用户的属性之间用,隔开

将每个员工信息解析出来并封装成Emp对象，并存入list集合

使用增强for循环遍历集合，输出每个员工信息

然后对集合中的员工按照年龄进行排序（降序），再次遍历集合，输出结果查看是否排序成功

3. 以上案例中，已知员工之间排序的默认规则是按照年龄排序，现在此基础上，要求对以上集合中的员工按照工资进行降序排列并遍历集合输出排序后的员工信息--（15分）

第二大题：（共15分）

1. 创建一个List<String>集合，并添加元素“ok.txt“，“hello.jpg”，“day01.rar”，“world.jpg”,"no.txt"，

获取子集“hello.jpg”，“day01.rar”，“world.jpg“并输出子集

再将子集中以“jpg”结尾的元素的后缀都替换成“png”;然后输出原集合查看内容