**考试试卷**

**笔试**

**A卷**

考试时间150分钟

总分 100分

姓 名\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

身份证号\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

准考证号\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

考试结束后考卷必须收回，否则考试成绩无效

**考试试卷**

1. **填空题（共20个题目，总计20分）**
2. Java技术按照用途不同分为三大版本，分别是JavaSE、 和JavaMe。
3. Java语言规定标识符由字母、下划线、美元符号和数字组成，并且第一个字符不能是 。
4. 三种控制结构分别是顺序结构、选择结构、 。
5. 在JDK版本为1.7的情况下，switch中的表达式的结果可以使用byte、char、short、int、枚举、 类型。
6. 面向对象拥有三大特性，封装、继承、 。
7. 封装的几个关键字有：private、default、 、public。
8. 多态的实现有三个必要条件，分别是: 继承关系、方法重写、以及 。
9. ArrayIndexOutOfBoundsException是 异常。
10. 数组的三种初始化方式分别为：静态初始化、动态初始化、以及 。
11. 数组元素下标(或索引)的范围是\_\_\_\_\_\_\_\_\_\_\_\_。
12. StringBuffer与StringBuilder中，线程安全的为\_\_\_\_\_\_\_\_\_\_\_\_。
13. "hamburger".substring(4, 8) 返回的结果是\_\_\_\_\_\_\_\_\_\_\_\_。
14. List和Set有一个共同的父接口是\_\_\_\_\_\_\_\_\_\_\_\_。
15. Vector与ArrayList二者中，线程安全的是\_\_\_\_\_\_\_\_\_\_\_\_。
16. HashTable与HashMap二者中，线程安全的为\_\_\_\_\_\_\_\_\_\_\_\_。
17. IO流中，I表示Input，O表示\_\_\_\_\_\_\_\_\_\_\_\_。
18. IO流，按照功能不同进行区分，可以分为节点流和\_\_\_\_\_\_\_\_\_\_\_。
19. \_\_\_\_\_\_\_\_\_\_\_关键字可以保证代码块的同步。
20. 当调用一个线程对象的start方法后，线程马上进入\_\_\_\_\_\_\_\_\_\_\_状态。
21. 可以让一个线程进入阻塞状态的方法是\_\_\_\_\_\_\_\_\_\_\_。
22. **选择题（共25个题目，总计25分）**

|  |  |  |
| --- | --- | --- |
| **1)** | **以下选项中是对一个Java源文件进行正确编译的语句是（ ）（选择一项）** | |
|  |  |  |
|  | **A.** | javac Test.java |
|  | **B.** | javac Test |
|  | **C.** | java Test |
|  | **D.** | java Test.class |

|  |  |  |
| --- | --- | --- |
| **2)** | **在Java中，byte数据类型的取值范围是（ ）。（选择一项）** | |
|  |  |  |
|  | **A** | -128 ~ 127 |
|  | **B.** | -228 ~128 |
|  | **C.** | -255 ~ 256 |
|  | **D.** | -255 ~ 255 |

|  |  |  |
| --- | --- | --- |
| **3)** | **下面的代码段中，执行之后i和j的值是（ ）。（选择一项）** | |
|  | **int** i=1; **int** j;  j=i++; | |
|  | **A** | 1，1 |
|  | **B.** | 1，2 |
|  | **C.** | 2，1 |
|  | **D.** | 2，2 |

|  |  |  |
| --- | --- | --- |
| **4)** | **以下do-while循环代码的执行结果是（ ）。（选择一项）** | |
|  | **int** a=0;  **int** c=0;  **do**{  --c;  a=a-1;  }**while**(a>0);  System.*out*.println(a+" "+c); | |
|  | **A.** | -1 -1 |
|  | **B.** | 死循环 |
|  | **C.** | -1 -2 |
|  | **D.** | -1 0 |

|  |  |  |
| --- | --- | --- |
| **5)** | **while循环和do-while循环的区别是（ ）。（选择一项）** | |
|  | **A.** | 没有区别，这两个结构在任何情况下效果一样 |
|  | **B.** | while循环比do-while循环执行效率高 |
|  | **C.** | while循环是先循环后判断，所以循环体至少被执行一次 |
|  | **D.** | do-while循环是先循环后判断，所以循环体至少被执行一次 |

|  |  |  |
| --- | --- | --- |
| **6)** | **给出如下代码，如何使成员变量m被方法fun()直接访问（ ）。（选择一项）** | |
|  | **class** Test {  **private** **int** m;  **public** **static** **void** fun() {  }  } | |
|  | **A** | 将private int m 改为protected int m |
|  | **B.** | 将private int m 改为public int m |
|  | **C.** | 将private int m 改为static int m |
|  | **D.** | 将private int m 改为int m |

|  |  |  |
| --- | --- | --- |
| **7)** | **以下关于this和super关键字的说法错误的是（ ）。（选择二项）** | |
|  | **A.** | this关键字指向当前对象自身，super关键字指向当前对象的直接父类 |
|  | **B.** | 在main方法中可以存在this或super关键字，但不能同时存在。 |
|  | **C.** | this和super关键字都可以访问成员属性，成员方法和构造方法 |
|  | **D.** | 在一个类的构造方法中可以同时使用this和super来调用其他构造方法 |

|  |  |  |
| --- | --- | --- |
| **8)** | **编译运行如下Java代码，输出结果是（ ）。（选择一项）** | |
|  | **class** Base {  **public** **void** method(){  System.***out***.print ("Base method");  }  }  **class** Child **extends** Base{  **public** **void** methodB(){  System.***out***.print ("Child methodB");  }  }  **class** Sample {  **public** **static** **void** main(String[] args) {  Base base= **new** Child();  base.methodB();  }  } | |
|  | **A** | Base method |
|  | **B.** | Child methodB |
|  | **C.** | Base method  Child methodB |
|  | **D.** | 编译错误 |

|  |  |  |
| --- | --- | --- |
| **9)** | **下列选项中，关于Java的抽象类和抽象方法说法正确的是（ ）。（选择二项）** | |
|  | **A** | 抽象类中可以含有0个或多个抽象方法 |
|  | **B.** | 抽象类中不可以有构造方法 |
|  | **C.** | 一个类中若有抽象方法，则这个类必为抽象类 |
|  | **D.** | 子类必须重写父类所有的抽象方法 |

|  |  |  |
| --- | --- | --- |
| **10)** | **阅读如下Java代码,其中错误的行是（ ）。（选择二项）** | |
|  | **public** **class** Student {  **private** String stuId;  **public** **void** setStuId(String stuId) **throw** Exception { // 1  **if** (stuId.length() != 4) { // 2  **throws** **new** Exception("学号必须为4位!"); // 3  } **else** {  **this**.stuId = stuId; //4  }  }  } | |
|  | **A** | 1 |
|  | **B.** | 2 |
|  | **C.** | 3 |
|  | **D.** | 全部正确 |

|  |  |  |
| --- | --- | --- |
| **11)** | **编译并运行如下Java程序，将输出（ ）。（选择一项）** | |
|  | **public** **static** **void** main(String[] args) {  **try** {  **int** num1 = 2; **int** num2 = 0;  **int** result = num1 / num2;  System.*out*.println(result);  **throw** **new** NumberFormatException( );  } **catch** (ArrayIndexOutOfBoundsException e) {  System.*out*.print("1");  } **catch** (NumberFormatException e) {  System.*out*.print("2");  } **catch** (Exception e) {  System.*out*.print("3");  } **finally** {  System.*out*.print("4");  }  System.*out*.print("5");  } | |
|  | **A** | 134 |
|  | **B.** | 2345 |
|  | **C.** | 1345 |
|  | **D.** | 345 |

|  |  |  |
| --- | --- | --- |
| **12)** | **分析下面的Java源程序，编译后的运行结果是（ ）。（选择一项）** | |
|  | **import** java.util.\*;  **public** **class** Test {  **public** **static** **void** main(String[ ] args) {  **int** [ ] numbers=**new** **int**[ ]{1,2,3};  System.*out*.println(Arrays.*binarySearch*(numbers, 2));  }  } | |
|  | **A** | 输出：0 |
|  | **B.** | 输出：1 |
|  | **C.** | 输出：2 |
|  | **D.** | 输出：3 |

|  |  |  |
| --- | --- | --- |
| **13)** | **以下选项中关于int和Integer的说法错误的是（ ）。（选择二项）** | |
|  | **A.** | int是基本数据类型，Integer是int的包装类，是引用数据类型 |
|  | **B.** | int的默认值是0，Integer的默认值也是0 |
|  | **C.** | Integer可以封装了属性和方法提供更多的功能 |
|  | **D.** | Integer i=5;该语句在JDK1.5之后可以正确执行，使用了自动拆箱功能 |

|  |  |  |
| --- | --- | --- |
| **14)** | **在Java中，以下File类的方法中（ ）用来判断是否是目录。（选择一项）** | |
|  | **A** | isFile( ) |
|  | **B.** | getFile( ) |
|  | **C.** | isDirectory( ) |
|  | **D.** | getPath( ) |

|  |  |  |
| --- | --- | --- |
| **15)** | **以下代码的执行结果是（ ）。（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 |

|  |  |  |
| --- | --- | --- |
| **16)** | **如下Java代码，输出的运行结果是（ ）。（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 |

|  |  |  |
| --- | --- | --- |
| **17)** | **下面集合类中属于非线程安全，且结构采用了哈希表的是（ ）。（选择一项）** | |
|  | **A.** | Vector |
|  | **B.** | ArrayList |
|  | **C.** | HashMap |
|  | **D.** | Hashtable |

|  |  |  |
| --- | --- | --- |
| **18)** | **给定如下Java代码，编译运行的结果是（ ）。（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 |

|  |  |  |
| --- | --- | --- |
| **19)** | **阅读下列文件定入的Java代码，共有（ ）处错误。（选择一项）** | |
|  | **import** java.io.\*;  **public** **class** TestIO {  **public** **static** **void** main(String []args){  String str ="文件写入练习";  FileWriter fw = **null**; //1  **try**{  fw = **new** FileWriter("c:\mytext.txt"); //2  fw.writerToEnd(str); //3  }**catch**(IOException e){ //4  e.printStackTrace();  }**finally**{  //此处省略关闭流  }  }  } | |
|  | **A** | 0 |
|  | **B.** | 1 |
|  | **C.** | 2 |
|  | **D.** | 3 |

|  |  |  |
| --- | --- | --- |
| **20）** | **以下说法中关于线程通信的说法错误的是（ ）?。（选择一项）** | |
|  | **A.** | 可以调用wait()、notify()、notifyAll()三个方法实现线程通信 |
|  | **B.** | wait()、notify()、notifyAll()必须在synchronized方法或者代码块中使用 |
|  | **C.** | wait()有多个重载的方法，可以指定等待的时间 |
|  | **D.** | wait()、notify()、notifyAll()是Object类提供的方法，子类可以重写 |

|  |  |  |
| --- | --- | --- |
| **21)** | **以下选项中关于如下代码的说法正确的是（ ）。（选择二项）** | |
|  | **public** **class** TestBuffered {  **public** **static** **void** main(String[] args) **throws** IOException {  BufferedReader br =  **new** BufferedReader(**new** FileReader("d:/bjsxt1.txt"));  BufferedWriter bw =  **new** BufferedWriter(**new** FileWriter("d:/bjsxt2.txt"));  String str = br.readLine();  **while**(str !=**null**){  bw.write(str);  bw.newLine();  str = br.readLine();  }  br.close();  bw.close();  }  } | |
|  | **A.** | 该类使用字符流实现了文件复制，将d:/bjsxt1.txt复制为d:/bjsxt2.txt |
|  | **B.** | FileReader和FileWriter是处理流，直接从文件读写数据 |
|  | **C.** | BufferedReader和BufferedWriter是节点流，提供缓冲区功能，提高读写效率 |
|  | **D.** | readLine()可以读取一行数据，返回值是字符串类型，简化了操作 |

|  |  |  |
| --- | --- | --- |
| **22)** | **下列关于Thread类提供的线程控制方法的说法中，错误的是（ ）。（选择一项）** | |
|  | **A** | 线程A中执行线程B的join()方法，则线程A等待直到B执行完成 |
|  | **B.** | 线程A通过调用interrupt()方法来中断其阻塞状态 |
|  | **C.** | 若线程A调用方法isAlive()返回值为false，则说明A正在执行中，也可能是可运行状态 |
|  | **D.** | currentThread()方法返回当前线程的引用 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **23)** | | **以下选项中可以填写到横线处，让代码正确编译和运行的是（ ）。（选择一项）** | | |
|  | | **public** **class** Test **implements** Runnable {  **public** **static** **void** main(String[] args) {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  t.start();  System.*out*.println("main");  }  **public** **void** run() {  System.*out*.println("thread1!");  }  } | | |
|  | | **A.** | | Thread t = new Thread(new Test()); |
|  | | **B.** | | Test t = new Test(); |
|  | | **C.** | | Thread t = new Test(); |
|  | | **D.** | | Thread t = new Thread(); |
| **24)** | **以下协议都属于TCP/IP协议栈，其中位于传输层的协议是（ ）。（选择二项）** | | | |
|  | **A** | TCP | | |
|  | **B.** | HTTP | | |
|  | **C.** | SMTP | | |
|  | **D.** | UDP | | |

|  |  |  |
| --- | --- | --- |
| **25)** | **ServerSocket的监听方法accept( )方法的返回值类型是（ ）。（选择一项）** | |
|  | **A.** | Socket |
|  | **B.** | void |
|  | **C.** | Object |
|  | **D.** | DatagramSocket |

1. **判断题（共20个题目，总计10分，正确√，错误×）**
2. 和C相比，Java语言的核心优势是跨平台性。（ ）
3. 以下两种写法：

a. 70L\*60\*24\*365\*70\*20

b. 70\*60\*24\*365\*70\*20L

没有任何区别?（ ）

1. 下边这段代码没有任何错误，可以完成i的打印。（ ）

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

System.out.println(i); //循环体

}

System.out.println("(((((("+i);

1. 在循环体中，break与continue的作用相同，都可以结束当前循环。（ ）
2. 接口中的常量一定是public static final的。（ ）
3. 接口中既不能定义普通变量，也不能定义普通方法。（ ）
4. 多态的使用时，涉及到两种转型方式，向上转型与向下转型。（ ）
5. Java中，不存在多继承，但一个类可以实现同时多个接口。（ ）
6. Throwable是一个接口。（ ）
7. 数组是引用类型，数组也是对象。（ ）
8. 数组中的元素就像对象中的成员变量一样，初始化方式也一样。（ ）
9. String类的trim()方法可以去掉字符串中所有的空格。（ ）
10. File类可以代表一个目录，也可以代表一个文件。（ ）
11. 容器指的是可以容纳其他对象的对象。（ ）
12. Dictionary建立了关键字和值的映射，只要提供一个关键字，Dictionary就会返回一个相 应的值。（ ）
13. 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。（ ）
14. Word文档可以使用字符流进行操作。（ ）
15. 不希望序列化的属性，可以添加transient关键字。（ ）
16. A线程的优先级是10，B线程的优先级是1，那么当进行调度时一定会先调用A。（ ）
17. TCP、UDP都是面向链接的。（ ）
18. **简答题（共5个题目，总计25分）**
19. 一个构造方法调用另一个构造方法怎么调用? this()，这样的调用方式必须位于第  
    一句吗?（5分）
20. File类的方法mkdir与mkdirs有什么区别？（5分）
21. HashSet和TreeSet的区别以及各自的优缺点？（5分）
22. 序列化和反序列化指的是什么?（5分）
23. 死锁是怎么造成的？用文字表达?（5分）
24. **编码题（共3个题目，总计20分）**
25. 模拟实现Comparable和Comparator接口。（6分）
26. 综合使用List、Map容器存放如下数据, 并从map中取出“李四”。（6分）

姓名：张三 年龄：18 体重：90 地址：北京

姓名：李四 年龄：28 体重：50 地址：上海

1. 使用UDP的方式，完成对象的传递（8分）