

**尚学堂JavaSE 软件工程师**

**认证考试试卷**

**笔试**

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

**尚学堂JavaSE软件工程师认证考试试卷**

1. **填空题（共20个题目，总计20分）**
2. 一个Java源程序是由若干个类组成。如果源文件中有多个类时，则只能有一个类是

类，并且这个类必须与源文件名同名。

1. Java中有两种类型的选择结构的控制语句，分别是if语句和 。
2. 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 。
3. 在循环体中 语句只结束本次循环，而不是终止整个循环的执行。
4. 用关键字 修饰的成员变量是类变量，类变量是指不管类创建了多少对象，系统仅在第一次调用类的时候为类变量分配内存，所有对象共享该类的类变量。
5. 是一种特殊方法，它的名字必须与它所在的类的名字完全相同，并且不书写返回值类型。
6. char c=’a’;System.out.println(c+1);运行结果为: 。
7. 包是Java语言的核心类库，它包含了运行Java程序必不可少的系统类，使用该包下的类和接口不需要使用import导入。
8. Date类可以得到系统的当前日期和时间，但是格式并不是平常看到的格式，并且时间也不是精确到毫秒。但是 类却可以获取年月日时分秒毫秒，然后根据自己喜欢的格式输出日期。
9. 抽象类是指在class前加使用 关键字修饰，且可以存在抽象方法和普通方法的类。
10. 仅仅是给出了数组名字和元素的数据类型，要想真正的使用数组还必须为它分配内存空间。
11. 机制是一种非常有用的辅助性程序设计方法。采用这种方法可以使得在程序设计时将程序的正常流程与错误处理分开，有利于代码的编写和维护。
12. 输入流的唯一目的是提供通往数据的通道，程序可以通过这个通道读取数据，

方法给程序提供了一个从输入流中读取数据的基本方法。

1. File对象调用方法 创建一个目录，不包括所有必需但不存在的父目录，当且仅当已创建目录时，返回true；否则返回false。
2. 一个正在执行的线程可能被人为地中断，让出CPU的使用权，暂时中止自己的执行，进入 状态。
3. 在Java中 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。
4. 创建数组后，系统会给每一个数组元素一个默认的值，如String类型元素的默认值是是 。
5. 队列和堆栈有些相似，不同之处在于 。
6. 是统一资源定位器的简称，它表示Internet上某一资源的地址。
7. 是一种由节点组成的数据结构，每个节点都包含数据元素，并且有一个或多个子节点，每个子节点指向一个父节点。
8. **选择题（共25个题目，总计25分）**

|  |  |  |
| --- | --- | --- |
| **1)** | **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() {  // some code…  }  } | |
|  |  |  |
|  | **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 |

|  |  |  |
| --- | --- | --- |
| **2)** | **已知如下定义：String s=”story”;下面（ ）表达式是合法的。（选择一项）** | |
|  |  |  |
|  | **A** | s+=”books” |
|  | **B.** | char c=s[1] |
|  | **C.** | int len=s.length; |
|  | **D.** | String t=100; |

|  |  |  |
| --- | --- | --- |
| **3)** | **指出下列程序运行的结果（ ）。（选择一项）** | |
|  | **public** **class** Example {  String str = **new** String("good");  **char**[] ch = { 'a', 'b', 'c' };  **public** **static** **void** main(String[] args) {  Example ex = **new** Example();  ex.change(ex.str, ex.ch);  System.*out*.print(ex.str + "and");  System.*out*.print(ex.ch);  }  **public** **void** change(String str, **char** ch[]) {  str = "test ok";  ch[0] = 'g';  }  } | |
|  |  |  |
|  | **A** | goodandabc |
|  | **B.** | goodandgbc |
|  | **C.** | test okandabc |
|  | **D.** | test okandgbc |

|  |  |  |
| --- | --- | --- |
| **4)** | **在异常处理中，如释放资源、关闭文件等由（ ）来完成。（选择一项）** | |
|  |  |  |
|  | **A** | try子句 |
|  | **B.** | catch子句 |
|  | **C.** | finally子句 |
|  | **D.** | throw子句 |

|  |  |  |
| --- | --- | --- |
| **5)** | **Java中main方法的返回值是（ ）。（选择一项）** | |
|  |  |  |
|  | **A** | String |
|  | **B.** | int |
|  | **C.** | char |
|  | **D.** | void |

|  |  |  |
| --- | --- | --- |
| **6)** | **分析下面的Java代码片段，编译运行后的输出结果是（ ）。（选择一项）** | |
|  | **for** (**int** i = 0; i < 6; i++) {  **int** k = ++i;  **while** (k < 5) {  System.*out*.print(i);  **break**;  }  } | |
|  |  |  |
|  | **A** | 024 |
|  | **B.** | 02 |
|  | **C.** | 123 |
|  | **D.** | 13 |

|  |  |  |
| --- | --- | --- |
| **7)** | **下面（ ）方法是public void example(){…}的重载方法。（选择二项）** | |
|  |  |  |
|  | **A** | private String example(){…} |
|  | **B.** | public int example(String str){…} |
|  | **C.** | public void example2(){…} |
|  | **D.** | public int example(int m,float f){…} |

|  |  |  |
| --- | --- | --- |
| **8)** | **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 |

|  |  |  |
| --- | --- | --- |
| **9)** | **在Java中,如下代码段的输出结果为（ ）。（选择一项）** | |
|  | **public** **class** Test {  **public** **static** **void** main(String[] args) {  **int** x = 3, y = 4;  x = (y < x++) ? 1 : 0;  System.*out*.print(x);  }  } | |
|  |  |  |
|  | **A** | 0 |
|  | **B.** | 1 |
|  | **C.** | 3 |
|  | **D.** | 4 |

|  |  |  |
| --- | --- | --- |
| **10)** | **下面选项中有关Java异常处理模型的说法错误的是（ ）。（选择二项）** | |
|  |  |  |
|  | **A** | 一个try块只能有一条catch语句 |
|  | **B.** | 一个try块中可以不使用catch语句 |
|  | **C.** | catch块不能单独使用，必须始终与try块在一起 |
|  | **D.** | finally块可以单独使用，不是必须与try块在一起 |

|  |  |  |
| --- | --- | --- |
| **11)** | **给出下面的代码段，在代码说明//assignment x=a,y=b处写入如下（ ）个代码是正确的。（选择一项）** | |
|  | **public** **class** Base {  **int** w, x, y, z;  **public** Base(**int** a, **int** b) {  x = a;  y = b;  }  **public** Base(**int** a, **int** b, **int** c, **int** d) {  // assignment x=a,y=b  w = d;  z = c;  }  } | |
|  |  |  |
|  | **A** | Base(a,b); |
|  | **B.** | x=a,y=b; |
|  | **C.** | this(a),this(b); |
|  | **D.** | this(a,b) |

|  |  |  |
| --- | --- | --- |
| **12)** | **在Java中,下列集合类型可以存储无序、不重复的数据的是（ ）。（选择一项）** | |
|  |  |  |
|  | **A** | ArrayList |
|  | **B.** | LinkedList |
|  | **C.** | TreeSet |
|  | **D.** | HashSet |

|  |  |  |
| --- | --- | --- |
| **13)** | **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 |

|  |  |  |
| --- | --- | --- |
| **14)** | **分析如下Java 代码，编译运行后将输出（ ）。（选择一项）** | |
|  | **public** **class** Test {  **public** Test() {  }  **static** **void** print(List<Integer> al) {  al.add(2);  al = **new** ArrayList<Integer>();  al.add(3);  al.add(4);  }  **public** **static** **void** main(String[] args) {  List<Integer> al = **new** ArrayList<Integer>();  al.add(1);  *print*(al);  System.*out*.println(al.get(1));  }  } | |
|  |  |  |
|  | **A** | 1 |
|  | **B.** | 2 |
|  | **C.** | 3 |
|  | **D.** | 4 |

|  |  |  |
| --- | --- | --- |
| **15)** | **已知表达式int [] m={0,1,2,3,4,5,6};下面（ ）表达式的值与数组最大下标数相等。（选择一项）** | |
|  |  |  |
|  | **A** | m.length() |
|  | **B.** | m.length-1 |
|  | **C.** | m.length()+1 |
|  | **D.** | m.length+1 |

|  |  |  |
| --- | --- | --- |
| **16)** | **对于语句String s="my name is kitty"，以下选项中可以从其中截取”kitty”的是（ ）（选择二项）** | |
|  |  |  |
|  | **A** | s.substring(11,16) |
|  | **B.** | s.substring(11) |
|  | **C.** | s.substring(12,17) |
|  | **D.** | s.substring(12,16) |

|  |  |  |
| --- | --- | --- |
| **17)** | **阅读下列文件写入的Java代码，错误的行是（ ）。（选择二项）** | |
|  | **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** | 1 |
|  | **B.** | 2 |
|  | **C.** | 3 |
|  | **D.** | 4 |

|  |  |  |
| --- | --- | --- |
| **18)** | **File类中的（ ）方法可以用来判断文件或目录是否存在。（选择一项）** | |
|  |  |  |
|  | **A** | exist() |
|  | **B.** | exists() |
|  | **C.** | fileExist() |
|  | **D.** | fileExists() |

|  |  |  |
| --- | --- | --- |
| **19)** | **使用Java IO流实现对文本文件的读写过程中，需要处理下列（ ）异常。（选择一项）** | |
|  |  |  |
|  | **A** | ClassNotFoundException |
|  | **B.** | IOException |
|  | **C.** | SQLException |
|  | **D.** | RemoteException |

|  |  |  |
| --- | --- | --- |
| **20)** | **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 |

|  |  |  |
| --- | --- | --- |
| **21)** | **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 |

|  |  |  |
| --- | --- | --- |
| **22)** | **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 |

|  |  |  |
| --- | --- | --- |
| **23)** | **在Java中,下面代码的输出结果为（ ）。（选择一项）** | |
|  | **public** **static** **void** main(String[] args) {  **int**[] arrA = { 12, 22, 8, 49, 3 };  **int** k = 0;  **int** len = arrA.length;  **for** (**int** i = 0; i < len; i++) {  **for** (**int** j = i + 1; j < len; j++) {  **if** (arrA[i] > arrA[j]) {  k = arrA[i];  arrA[i] = arrA[j];  arrA[j] = k;  }  }  }  **for** (**int** i = 0; i < arrA.length; i++) {  System.*out*.print(arrA[i]);  **if** (i < arrA.length - 1) {  System.*out*.print("，");  }  }  } | |
|  |  |  |
|  | **A** | 3，8，12，22，49 |
|  | **B.** | 12，22，8，49，3 |
|  | **C.** | 49，22，12，8，3 |
|  | **D.** | 编译错误 |

|  |  |  |
| --- | --- | --- |
| **24)** | **分析下面的Java程序段，编译运行后的输出结果是（ ）。（选择一项）** | |
|  | **public** **class** Test {  **public** **void** changeString(StringBuffer sb) {  sb.append("stringbuffer2");  }  **public** **static** **void** main(String[] args) {  Test a = **new** Test();  StringBuffer sb = **new** StringBuffer("stringbuffer1");  a.changeString(sb);  System.*out*.println("sb = " + sb);  }  } | |
|  |  |  |
|  | **A** | sb = stringbuffer2stringbuffer1 |
|  | **B.** | sb = stringbuffer1 |
|  | **C.** | sb = stringbuffer2 |
|  | **D.** | sb = stringbuffer1stringbuffer2 |

|  |  |  |
| --- | --- | --- |
| **25)** | **下列关于Thread类提供的线程控制方法的说法中，错误的一项是（ ）。（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 |

1. **判断题（共20个题目，总计10分）**
2. 方法Integer.parseInt()的作用是将一个整数转变成String。（ ）
3. 求x的y次方，其表达式为：Math.pow(x,y)。（ ）
4. 数组的某个元素被传递给一个方法并被该方法修改，当被调用方法执行完毕时，这个元素中含有修改过的数值。（ ）
5. 运算符“==”用于比较引用时，如果两个引用指向内存同一个对象，则返回true。（ ）
6. 对于数组int[][] t={{1,2,3},{4,5,6}}来说，t.length等于3，t[0].length等于2（ ）
7. 假设文件”a.txt”的长度为100字节，那么当正常运行语句”OutputStream f=new FileOutputStream(new File(“a.txt”));”之后，文件”a.txt”的长度变为0字节。（ ）
8. 如果没有为类中的某些成员赋初始值，Java会为类成员赋予固定的初始值，如数值变量的值为0，布尔变量的值为true，未初始化的引用为null。（ ）
9. 类的方法通常设为public，而类的实例变量一般也设为public。（ ）
10. 线程可以用yield使低优先级的线程运行。（ ）
11. 如果在定义一个类的时候没有用到关键字extends，则这个类没有直接父类。（ ）
12. Javadoc是一种从文档注释生成HTML帮助文件的工具。它可以处理在Java源程序中介于“/\*”和“\*/”之间的注释，并生成相应的程序文档。（ ）
13. System.out.println(“Hello java!”)中out是System类的一个成员变量。（ ）
14. 声明为final的类不能是超类。（ ）
15. 基本数据类型的值可以被直接存储在Vector对象中。（ ）
16. 对子类实例化对象需要调用超类的构造函数进行初始化工作，完成对子类中的超类实例变量的初始化，则子类可以继承超类的构造函数。（ ）
17. Java中变量名不区分大小写，number和NuMbEr代表同一个变量。（ ）
18. 只有在块中声明的变量具有自动生命周期。（ ）
19. 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。（ ）
20. 数组可以声明为任何数据类型，包括任何基本数据类型和引用数据类型。（ ）
21. Java集合中的Set接口和List接口都是从Collection接口派生出来的。（ ）
22. **简答题（共5个题目，总计25分）**
23. final、finally、finalize的区别。（3分）
24. Error和Exception的区别， Checked异常和Runtime异常的区别。（6分）
25. String、StringBuffer、StringBuilder区别与联系。（5分）

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（6分）

1. 实现Java反射技术的主要类有哪些，其作用分别是什么？（5分）
2. **编码题（共3个题目，总计20分）**
3. **编写应用程序，创建类的对象，分别设置圆的半径、圆柱体的高，计算并分别显示圆半径、圆面积、圆周长，圆柱体的体积。（7分）**

**提示:**（1）编写一个圆类Circle，该类拥有：

* + - 1. 一个成员变量，radius（私有，浮点型）;//存放圆的半径；
      2. 两个构造方法

Circle（） //将半径设为0

Circle（double r ） //创建Circle对象时将半径初始化为r

* + - 1. 三个成员方法

double getArea（） //获取圆的面积

double getPerimeter（） //获取圆的周长

void show（） //将圆的关径、周长、面积输出到屏幕

（2） 编写一个圆柱体类Cylinder，它继承于上面的Circle类。还拥有：

* + 1. 一个成员变量，double hight （私有，浮点型）; //圆柱体的高；
    2. 构造方法

//创建Cylinder对象时将半径初始化为r,高度初始化为h

Cylinder（double r,double h）

* + 1. 成员方法

double getVolume（） //获取圆柱体的体积

void showVolume（） //将圆柱体的体积输出到屏幕

1. **设计4个线程，其中两个线程每次对j增加1，另外两个线程对j每次减少1。（6分）**

**要求：**使用内部类实现线程，对j增减的时候不考虑顺序问题。

1. **使用Java Socket编程，完成如下功能：（7分）**
2. 要求从客户端录入几个字符，发送到服务器端。
3. 由服务器端将接收到的字符进行输出。
4. 服务器端向客户端发出“您的信息已收到”作为响应。
5. 客户端接收服务器端的响应信息。

**提示：**

服务器端:PrintWriter out =new PrintWriter(socket.getOutputStream(),true);

客户端:BufferedReader line=new BufferedReader(new InputStreamReader(System.*in*));