**尚学堂Java SE软件工程师认证考试试卷**

**答案精讲**

1. **填空题（共20个题目，总计20分）**
2. class
3. 8
4. String
5. break
6. 方法
7. 方法重载
8. package
9. Object
10. new
11. array[0].length
12. StringBuffer
13. java.lang .Number
14. finalize()
15. random()
16. throws
17. -1
18. Runnable
19. 链表
20. ServerSocket
21. 事件处理器
22. **选择题（共25个题目，总计25分）**
23. B

分析：java严格区分大小写,System是系统类，首写字母要大写。

1. AB

分析:’one’是字符串，不使用单引号,255.0是double类型 ，不直接赋值给int变量。

1. A
2. D

分析：i的初始值为0,i<10条件成立，进行for循环，使用if判断0%2!=0条件不成立，执行打印输出i的值（0），执行i++,i的值为1,1<10条件成立，进入for循环，使用if判断1%2!=0，条件成立，执行return,方法执行完毕。

1. AD

分析:答案B定义返回类型为String，实际返回类型是int，不相符。答案C定义void，不可以使用return返回值。

1. B

分析:二分查找，2在数据中的索引为1。

1. C

分析:调用count1()方法，修改成员变量count的值为10,再次调用方法count2()是输出的结果是10，因为成员变量的作用域是整个类。

1. CD
2. D

分析：因为字符串的concat方法拼接字符串会得到一个新的字符串对象，这个对象并没有被任何对象所引用，而原字符串str依然是null,所以会产生异常，应改为str=str.concat(“abc”);。

1. C

分析: 执行x=new StringBuffer("BBB");后，只是x的指向发生了变化，并没有改变main方法中a的指向和内容。

1. C
2. CD
3. A

分析：第一次添加数据，索引为0,第二次在指定位置上添加数据，位置的值2大于集合的长度size,所以会产生异常。

1. AC

分析：静态方法也称为类方法，是在类加载到内存里与类一起加载到内容，这个时候还没有对象的产生，所以不能使用this。this代表当前对象的引用。

1. B
2. AB

分析:抽象类不能用于创建对象，抽象类中的构造方法可以被子类调用去构造子类对象。

1. B

分析：Reader类没有read(int n )方法。

1. D

分析:关闭流时会产生IO异常，未对异常进行处理。

1. BD
2. BC

分析:线程的优先级也可使用[setPriority](mk:@MSITStore:F:\\01软件\\JDK_API_1_6_zh_CN.CHM::/java/lang/Thread.html" \l "setPriority(int))(int newPriority)方法进行设置，线程的优先级范围在1-10之间。

1. B

分析:Thread类属于java.lang包中，启动线程调用start()方法。

1. A
2. BD
3. AD

分析:属性可以出现在起始标记时在，不能出现在结束标记中。

1. D
2. **判断题（共20个题目，总计10分）**
3. √
4. ×

分析:编译命令是javac Hello.java，执行命令为java Hello。

1. ×

分析：变量命名要遵循命名规范。

1. ×

在Java中任何一个子类的构造方法可以使用super调用其父类的构造方法（包括隐式调用），也可以使用this调用当前类的其他构造方法，并且都必须是构造方法的第一条语句。

1. √
2. √
3. ×

分析:父类的私有方法，不能被子类重写。

1. ×

分析：接口中的方法也是抽象方法。

1. ×

分析：构造方法没有返回值，也不可以写void。

1. ×

分析：因为没有使用new在堆内存中开辟空间，所以不会产生新对象，而是指向了同一个引用。

1. √
2. ×

分析:Exception是异常的父类，放到最后一个catch子句中。

1. √
2. ×

分析:泛型是JDK1.5的特性。

1. √
2. ×

分析:不会产生异常，如果文件不存在，则delete方法的返回值为false。

1. ×

分析：当一个线程进入一个对象的一个synchronized方法后，其它线程不可以再进入该对象的其它synchronized方法执行，但可以进入其他非synchronized方法执行。

1. ×

分析: TCP/IP网络参考模型分四层，分别是应用层，传输层，网际互联层，网络接入层。

1. √
2. √
3. **简答题（共5个题目，总计25分）**
4. **在Java的异常处理机制中，关键字try、catch、finally、throw、throws分别代表什么含义？（5分）**

**答:** try：用来指定一块预防所有“异常”的程序。（1分）

catch:语句紧跟在try块后面，用于指定你想要捕捉的“异常”的类型。（1分）

finally:为确保一段代码不管发生什么“异常”都被执行一段代码。（1分）

throw:语句用来明确地抛出一个“异常”，在方法体中。（1分）

throws:用来标明一个方法可以抛出的各种“异常”，写在方法名的后面。（1分）

1. **请你简述sleep()和wait()有什么区别？（5分）**

答:

sleep是线程类(Thread)的方法；（0.5分）作用是导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复；（1分）调用sleep()不会释放对象锁。（1分）

wait是Object类的方法；（0.5分）对此对象调用wait方法导致本线程放弃对象锁，进入等 待此对象的等待锁定池。（1分）只有针对此对象发出notify方法(或notifyAll)后本线程才进入对象锁定池，准备获得对象锁进行运行状态。（1分）

1. **请你简述HashMap和Hashtable的联系和区别？（5分）**

答：共同点：

1. 二者都实现了Map接口，是将惟一键映射到特定的值上。（1分）
2. 底层都采用哈希表作为存储结构，查询速度快，在很多情况下可互用。（1分）

主要区别：

1. Hashtable继承自Dictionary类，HashMap是Java1.2引进的Map接口的实现。（1分）
2. Hashtable的线程安全的，而HashMap不是。在多个线程访问Hashtable时，不需要自己为它的方法实现同步，而HashMap就必须为之提供外同步。（1分）
3. HashMap没有排序，允许一个null键和多个null值，而Hashtable不允许。（0.5分）
4. HashMap把Hashtable的contains( )去掉了，改成containsValue和containsKey，因为contains方法容易让人引起误解。（0.5分）
5. **什么是设计模式？设计模式的作用？（5分）**

答：

设计模式是一套被反复使用的、多数人知晓、经过分类编目的优秀代码设计经验的总结。每个设计模式都是特定环境下特定问题的处理方法。（1分）

设计模式主要作用如下：

1. 重用设计和代码。重用设计比重用代码更有意义，自动带来代码重用。（1分）
2. 提高扩展性。大量使用面向接口编程，预留扩展插槽，新的功能或特性很容易加入到系统中来。（1分）
3. 提高灵活性。通过组合提高灵活性，可以允许代码修改平稳发生，对一处的修改不会波及到很多其他模块。（1分）
4. 提高开发效率。正确使用设计模式，可以节省大量的时间。（1分）
5. **TCP/IP协议栈中，TCP协议和UDP协议的联系和区别（5分）**

答：联系：

TCP和UDP是TCP/IP协议栈中传输层的两个协议，它们使用网络层功能把数据包发送到目的地，从而为应用层提供网络服务。（1分）

区别：

1. TCP是面向连接的传输。UDP是无连接的传输。（1分）

2. TCP保证数据安装发送顺序到达，UDP无法保证。（1分）

4. TCP是可靠性传输，而UDP则是不可靠传输。（1分）

5. UDP因为少了很多控制信息，所以传输速度比TCP速度快。（0.5分）

6. TCP适合用于传输大量数据，UDP适合用于传输小量数据。（0.5分）

比喻：

TCP的server和client之间通信就好比两个人打电话。UDP的server和client之间的通信就像两个人发电报或者发短信。

通过生活案例正确说明TCP和UDP区别，可加分1分，但总分不超过5分。

1. **编码题（共3个题目，总计20分）**
2. **编写两个线程,一个线程打印1-52的整数，另一个线程打印字母A-Z。打印顺序为12A34B56C….5152Z。即按照整数和字母的顺序从小到大打印，并且每打印两个整数后，打印一个字母，交替循环打印，直到打印到整数52和字母Z结束（8分）**

**打印类（共享资源）**

|  |
| --- |
| **public** **class** Printer {  **private** **int** index=1;//设为1，方便计算3的倍数  //打印数字的方法，每打印两个数字，等待打印一个字母  **public** **synchronized** **void** print(**int** i){  **while**(index%3==0){  **try** {  wait();  } **catch** (InterruptedException e) {  e.printStackTrace();  }  }  System.*out*.print(""+i);  index++;  notifyAll();  }  //打印字母，每打印一个字母，等待打印两个数字  **public** **synchronized** **void** print(**char** c){  **while**(index%3!=0){  **try** {  wait();  } **catch** (InterruptedException e) {  e.printStackTrace();  }  }  System.*out*.print(""+c);  index++;  notifyAll();  }  } |

**打印字母线程**

|  |
| --- |
| **public** **class** LetterPrinter **extends** Thread {  **private** Printer p;  **public** LetterPrinter(Printer p){  **this**.p=p;  }  **public** **void** run(){  **for**(**char** c='A';c<='Z';c++){  p.print(c);  }  }  } |

**打印数字线程**

|  |
| --- |
| **public** **class** NumberPrinter **extends** Thread {  **private** Printer p;  **public** NumberPrinter(Printer p){  **this**.p=p;  }  **public** **void** run(){  **for**(**int** i=1;i<=52;i++){  p.print(i);  }  }  } |

**测试类**

|  |
| --- |
| **public** **class** Test {  **public** **static** **void** main(String[] args) {  Printer p=**new** Printer(); //创建打印机对象  Thread t1=**new** NumberPrinter(p); //创建线程对象  Thread t2=**new** LetterPrinter(p); //创建线程对象  t1.start(); //启动线程  t2.start(); //启动线程  }  } |

评分标准：

1. 编写打印类Printer，声明私有属性index，初始值1，表示第几次打印。（1分）
2. 在打印类Printer中编写打印数字的方法print(int i)，3的倍数就使用wait()方法等待，否则就输出i，使用notifyAll()进行唤醒其它线程。（1.5分）
3. 在打印类Printer中编写打印字母的方法print(char c)，不是3的倍数就等待，否则就打印输出字母c，使用notifyAll()进行唤醒其它线程。（1.5分）
4. 编写打印数字的线程NumberPrinter继承Thread类，声明私有属性private Printer p;在构造方法中进行赋值。（0.5分）实现父类的run方法，调用Printer类中的输出数字的方法。（1分）
5. 编写打印字母的线程LetterPrinter继承Thread类，声明私有属性private Printer p;在构造方法中进行赋值。（0.5分）实现父类的run方法，调用Printer类中的输出字母的方法。（1分）
6. 编写测试类Test，创建打印类对象，创建两个线程类对象，启动线程。（1分）
7. **使用IO包中的类读取D盘上exam.txt文本文件的内容，每次读取一行内容，将每行作为一个输入放入ArrayList的泛型集合中并将集合中的内容使用加强for进行输出显示。（5分）**

|  |
| --- |
| **public** **class** Test {  **public** **static** **void** main(String[] args) **throws** IOException {  String path="D:\\exam.txt";  *outputMethod*(path);  }  **public** **static** **void** outputMethod(String path) **throws** IOException {  List<String> list = **new** ArrayList<String>(); // 创建集合对象  // 创建缓冲区对象  BufferedReader br = **new** BufferedReader(**new** FileReader(path));  String line = br.readLine(); // 读取数据每次读一行  **while** (line != **null**) {  list.add(line);  line = br.readLine();  }  br.close(); //关闭  **for**(String s:list){  System.*out*.println(s);  }  }  } |

评分标准：

1. 创建缓冲字符输入流读取D盘上exam.txt文本文件的内容。（1分）
2. 每次读取一行内容，将每行作为一个输入放入ArrayList的泛型集合中。（2分）
3. 关闭输入流。（1分）
4. 将集合中的内容使用加强for进行输出显示。（1分）
5. **使用反射机制完成学生对象的创建并输出学生信息。（7分）**

**学生类**

|  |
| --- |
| **public** **class** Student {  **private** String name;//姓名  **private** **int** age;//年龄  **private** **int** score;//成绩  //…此处省略各个属性的getter和setter方法  **public** Student(String name, **int** age, **int** score) {  **super**();  **this**.name = name;  **this**.age = age;  **this**.score = score;  }  **public** Student() {  **super**();  }  @Override  **public** String toString() {  **return** name+"\t"+age+"\t"+score;  }  } |

**测试类**

|  |
| --- |
| **public** **class** TestStudent {  **public** **static** **void** main(String[] args) **throws** Exception {  Scanner input = **new** Scanner(System.*in*);  System.*out*.println("请输入学员的信息:(如:mary:20:90)");  String info = input.next();  // 使用split方法进行分割  String[] spinfo = info.split(":");  // 调用Constructor的newInstance方法创建对象  Class<Student> clazz =  (Class<Student>) Class.*forName*("com.bjsxt.bean.Student");  Constructor<?> cons =  clazz.getConstructor(String.**class**, **int**.**class**,**int**.**class**);  String name = spinfo[0];  **int** age = Integer.*parseInt*(spinfo[1]);  **int** score = Integer.*parseInt*(spinfo[2]);  Student stu = (Student) cons.newInstance(name,age,score);  System.*out*.println(stu);  }  } |

评分标准

1. 定义一个学生类Student，其中包含姓名、年龄、成绩的属性。（0.5分）
2. 编写带参与无参构造方法。（0.5分）
3. 重写父类的toString()方法用于输出学生的信息。（0.5分）
4. 编写测试类TestStudent，从键盘录入学生的信息格式为(姓名:年龄:成绩)一次性录入使用“:”分隔，举例(mary:20:90) 。（1分）
5. 使用String类的split方法按照“:”进行分隔。（1分）
6. 使用Class.forName( )获取Student类的Class信息。（1分）
7. 调用Constructor的newInstance()方法并用分隔后信息初始化学生对象。（2分）
8. 调用重写父类的toString()方法将学生信息进入输出显示。（0.5分）