# 初级：

## 单选：

1. ISO定义了网络的7层标准模型，以下选项中按自下而上的顺序排列正确的是：

物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

1. 关于HTTP和HTTPS的区别，以下说法种错误的是：D
   1. HTTPS比HTTP更安全
   2. HTTP是超文本传输，HTTPS是加密传输协议
   3. HPPTS需要申请CA证书
   4. HTTP和HTTPS使用的端口必须不能相同，前者使用80，后者使用443
2. 常见HTTP请求返回的状态码中，表示Forbidden资源不允许使用的是：D
   1. 302
   2. 500
   3. 404
   4. 403
3. 在各自最优的条件下，对N个数进行排序，以下算法中复杂度最低的是：C
   1. 快速排序
   2. 插入排序
   3. 冒泡排序
   4. 归并排序
4. 一个栈的入栈序列为ABCDEF，则不可能的出栈序列是：D
   1. DEFCBA
   2. DCEFBA
   3. FEDCBA
   4. FECDBA
5. 如下哪个不是JAVA的关键字：B
   1. const
   2. NULL
   3. false
   4. this
6. 以下代码运行输出是：C

public class Person{

private String name=”Person”;

int age=0;

}

public class Child extends Person{

public String grade;

public static void main(String[] args){

Person p = new Child();

System.out.println(p.name);

}

}

* 1. 输出：Person
  2. 没有输出
  3. 编译出错
  4. 运行出错

1. 关于垃圾收集的叙述正确的是：B
   1. 程序开发者必须自己创建一个线程进行内存释放的工作
   2. 垃圾收集将检查并释放不再使用的内存
   3. 垃圾收集允许程序开发者明确指定并立即释放内存
   4. 垃圾收集能在期望的时间释放被JAVA对象使用的内存
2. Java中的集合类包括ArrayList、LinkedList、HashMap等类，下列关于集合类描述错误的是：C
   1. ArrayList和LinkedList均实现了List接口
   2. ArrayList的访问速度比LinkedList快
   3. 添加和删除元素时，ArrayList的表现更佳
   4. HashMap实现Map接口，它允许任何类型的键和值对象，并允许将null用作键或值
3. Spring声明式事务中默认的传播属性是：A
   1. PROPAGATION\_REQUIRED
   2. PROPAGATION\_REQUIRES\_NEW
   3. PROPAGATION\_SUPPORTS
   4. PROPAGATION\_MANDATORY

## 多选：

1. 下列关于HTTP协议的描述正确的是：BC
   1. HTTP协议是有状态的
   2. HTTP是一个基于TCP/IP的应用层协议
   3. HTTP请求方法有GET、HEAD、POST、PUT等
   4. HTTP响应码200表示服务器异常
2. 事务的特性有：ACDE
   1. 原子性
   2. 可用性
   3. 一致性
   4. 隔离性
   5. 持久性
3. switch语句支持以下哪些基本类型：ABCD
   1. byte
   2. short
   3. int
   4. char
   5. long
4. 以下属于面向对象的特征的是：ACD
   1. 继承
   2. 重载
   3. 多态
   4. 封装
5. 以下关于final关键字说法错误的是：AC
   1. final是java中的修饰符，可以修饰类、接口、抽象类、方法和属性
   2. final修饰的类肯定不能被继承
   3. final修饰的方法不能被重载
   4. final修饰的变量不允许被再次赋值

## 简答题：

1. 抽象类（abstract class）和接口（interface）有什么区别？

抽象类可以有构造方法；接口中不能有构造方法。

抽象类中可以有普通成员变量或者常量或者静态变量；接口中没有普通成员变量和静态变量，只能是常量（默认修饰符为 publci static final）。

抽象类中可以包含非抽象的普通方法和抽象方法及静态方法；接口中的所有方法必须都是抽象的，不能有非抽象的普通方法和静态方法（默认修饰符为 public abstract）。

抽象类中的抽象方法访问类型可以是 public、protected 的；接口中的抽象方法只能是 public 的（默认修饰符为 public abstract）。一个子类可以实现多个接口，但只能继承一个抽象类。

1. 程序实现冒泡排序算法。

public class 冒泡排序 {

public static void main(String[] args) {

int[] a={49,38,65,97,76,13,27,49,78,34,12,64,1,8};

System.out.println("排序之前：");

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

System.out.print(a[i]+" ");

}

//冒泡排序

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

for(int j = 0; j<a.length-i-1; j++){

//这里-i主要是每遍历一次都把最大的i个数沉到最底下去了，没有必要再替换了

if(a[j]>a[j+1]){

int temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

System.out.println();

System.out.println("排序之后：");

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

System.out.print(a[i]+" ");

}

}

}

1. 哈

# 高级：

## 单选：

## 多选：

## 简答：

1. 程序实现归并排序算法。

public class 归并排序 {

public static void main(String[] args) {

int[] a={49,38,65,97,76,13,27,49,78,34,12,64,1,8};

System.out.println("排序之前：");

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

System.out.print(a[i]+" ");

}

//归并排序

mergeSort(a,0,a.length-1);

System.out.println();

System.out.println("排序之后：");

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

System.out.print(a[i]+" ");

}

}

private static void mergeSort(int[] a, int left, int right) {

if(left<right){

int middle = (left+right)/2;

//对左边进行递归

mergeSort(a, left, middle);

//对右边进行递归

mergeSort(a, middle+1, right);

//合并

merge(a,left,middle,right);

}

}

private static void merge(int[] a, int left, int middle, int right) {

int[] tmpArr = new int[a.length];

int mid = middle+1; //右边的起始位置

int tmp = left;

int third = left;

while(left<=middle && mid<=right){

//从两个数组中选取较小的数放入中间数组

if(a[left]<=a[mid]){

tmpArr[third++] = a[left++];

}else{

tmpArr[third++] = a[mid++];

}

}

//将剩余的部分放入中间数组

while(left<=middle){

tmpArr[third++] = a[left++];

}

while(mid<=right){

tmpArr[third++] = a[mid++];

}

//将中间数组复制回原数组

while(tmp<=right){

a[tmp] = tmpArr[tmp++];

}

}

}

1. 哈哈