**中软国际考题-笔试**

1. 单选题

1、Java中，枚举的构造方法默认是（C）

A.public B.protected C.private D.default

2、Java中，定义一个颜色的枚举类型Color，下面正确的是：（A）

A. public enum Color{red,green,yellow,blue;}

B. public enum Color{1:red,2:green,3:yellow,4:blue;}

C. public enum Color{1:red;2:green;3:yellow;4:blue;}

D. public enum Color{String red,String green,String yellow,String blue;}

3、Java中，关于枚举的说法错误的是（D）

A. 枚举类型也可以实现一个或多个接口

B. 枚举类型实现接口时也要实现该接口中全部方法

C. 枚举类型里定义抽象方法无须显示的使用abstract关键字

D. 枚举类型继承接口要使用关键字是extends

4、下列描述中正确的是（B）

A. 枚举类全部都以引用数据类型的形式来表示

B. 枚举类允许用常量来表示特定的数据片断

C. 有了枚举，可以把相关的类分组到一个枚举类型里

D. 枚举提供了比数据对象更多的方法

5、下列关于枚举类描述错误的是（B）

A. 枚举的更多用法还需要在开发中去研究创造

B. 枚举的本质是java流程控制

C. 枚举在Java家族中只占了很小的一块比重

D. 枚举（enum）类型是Java 5新增的特性。

6、java中枚举类描述正确的是（D）

A．在引用一个不存在的枚举值的时候，编译器不会报错

B. 在JDK1.6之前，我们定义常量都是：public static fianl...

C. 枚举已结在项目开发中大量使用

D. 枚举是用来构建常量数据结构的模板，这个模板可扩展

7、java中，Enum枚举类中的name()方法的作用是（C）

A. 比较此枚举与指定对象的顺序

B. 返回枚举常量的序数

C. 返回引枚举常量的名称，在其枚举声明中对其进行声明

D. 返回还指定名称的指定枚举类型的枚举常量

8、Java中定义一个颜色的枚举类型color，下面正确的是（A）

A.       public enumColor{red,green,yellow,blue;}

B.       public enumColor{1:red,2:green,3:yellow,4:blue;}

C.       public enumColor{1:red;2:green;3:yellow;4:blue;}

D.       public enum Color{Stringred,String String green,yellow,String blue;}

9、 Java的枚举，以下定义一个枚举类型color正确的是(A)

A. public enumColor{red,green,yellow,blue;}

B. public enumColor{1:red,2:green,3:yellow,4:blue;}

C. public enumColor{1:red;2:green;3:yellow;4:blue;}

D. public enum Color{Stringred,String String green,yellow,String blue;}

10. Java的枚举，是jdk哪个版本以后引入的（C）

A.jdk1.1 B.jdk1.2 C.jdk1.5 D.java8

11.Java的枚举，所有的枚举值之间用什么（A）分隔

A.逗号 B.分号 C.下划线 D.句号

12、Java中，定义一个颜色的枚举类型color，下面正确的是（A）.

A.public enum Color{red,green,yellow,blue;}

B.public enum Color{1:red,2:green,3:yellow,4:blue;}

C.public enum Color{1:red,2:green,3:yellow,4:blue;}

D.public enum Color{String red,String green, String yellow, String blue;}

13、Java中，枚举类的构造方法默认是是（C）

A.public B.protected C.private D.都可以

14、java中定义一个颜色的枚举类,下面正确的是：A

A. public enumColor{red,green,yellow,blue;}

B. public enumColor{1:red,2:green,3:yellow,4:blue;}

C. public enumColor{1:red;2:green;3:yellow;4:blue;}

D. public enum Color{Stringred,String String green,yellow,String blue;}

15、java中枚举类的默认构造方法是：C

A. public B. protected

C. private D. 都可以

16、下列说法正确的是（A）

A. 枚举不能再继承其它的类 B. 枚举不能实现接口

C. 枚举中可以有public的构造函数 D. 枚举值是可变的

1. 多选题
2. Java中，以下说法正确的有？（A、B、C）

A.枚举类型的取值默认为publicstatic final B. 枚举类型中可以定义属性和方法

C. 构造器必须私有 D . 枚举类型不可实现其他接口

2、以下关于匿名类说法错误的是？（A、C）

A. EnumSet保证集合中的元素可以重复存储数据

B. EnumSet保证集合中的元素不重复

C. EnumMap中的 key是enusrm类型

D. EnumMap中的value则可以是任意类型

3、枚举几个常用方法（A、B、C）

A、 ordinal B、compareTo

C、equals D、hsahCode

4、[枚举类型](http://www.51code.com/)的定义和使用正确的是（A、B、C）

A.       枚举类型的取值默认为publicstatic final

B.       枚举类型中可以定义属性和方法

C.       构造器必须私有

D.       枚举类型不可实现其他接口

5、下面哪些项编译会出错？（A、B、C）

A.枚举类型取值默认为public static final

B.枚举类可以定义属性和方法

C.枚举类的构造方法为私有

D.枚举不可以实现接口

6. 以下属于枚举的方法是（A、B、C）

A. ordinal（） B. equals（） C.compareTo() D.wait()

7、Java中，以下说法正确的是（ ABC ） 。

A.枚举类型的取值默认为public static final B.枚举类型中可以定义属性和方法

C.构造器必须私有 D.枚举类型不可实现其他接口

8、java中，以下说法正确的是：ABC

A. 枚举类型的取值默认为publicstatic final

B. 枚举类型中可以定义属性和方法

C. 构造器必须私有

D. 枚举类型不可实现其他接口

9、下面关于枚举的说法正确的是？（B、C、E）

A. 枚举类默认继承了Object类

B. 枚举类的所有实例必须放在第一行显示

C. 枚举类的默认修饰符为final，所以不可以被继承

D. 枚举类的构造方法可以是public修饰的

E. 枚举类中也可以定义属性和方法，可以是静态的，也可以是非静态的

1. 填空题（每空1分，共10分）
2. 枚举类的\_\_\_\_\_\_\_\_\_\_方法用于返回此枚举常量的名称。 (name)

2、\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_是两个枚举集合(请写出java包链接路径)。

(java.util.EnumSet, java.util.EnumMap)

3、两枚举常量可以用 == 比较

4、枚举有两个限制：\_\_\_\_\_\_\_\_\_\_，\_\_\_\_\_\_\_\_\_\_ 。(不能继承另一个类， 不能是超类不能被扩展)

5、使用enum定义的枚举类默认继承了\_\_\_\_\_类。（java.lang.Enum）

6、java.lang.Enum类实现了\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_两个接口。（java.lang.Serializable和java.lang.Comparable）

7、枚举类型使用\_\_\_\_\_\_\_\_\_\_\_\_\_关键字声明。（enum）

8、 Enum枚举类中的name()方法的作用是：\_\_\_\_\_\_\_\_\_\_。(返回引枚举常量的名称，在其枚举声明中对其进行声明)

9、jdk（1.5）以后引入关键字 enmu

10、所有枚举都默认继承的java类是\_\_\_\_\_\_\_\_\_\_。(java.lang.Enum)

1. 判断题
2. 两个枚举常量可以用“==”比较并且枚举值可以用于switch语句中（ T ）
3. **EnumList和EnumTable是两个枚举集合。（ N ）**
4. 枚举类内也可以定义属性和方法（ T ）
5. 使用关键字static创建枚举类型。（ F ）
6. 枚举是指一组固定常量组成合法值的类型（T ）
7. Java 枚举本质上是 int 值。只能通过公有的静态 final 域为枚举类导出实例（T ）
8. 枚举类的构造器只能使用private。（ T ）
9. Java中的枚举类是值类型。（ F）
10. Jdk1.5版本后，枚举类型不可以用在switch表达式中（F）
11. 枚举的方法可以重载（ T ）
12. 枚举能够定义类类型，但却不能使用new实例化一个枚举（ T ）
13. java枚举的values()方法返的是一个数组（T）
14. 枚举类的构造方法默认是public。 （F）
15. 枚举类对象的属性不应允许被改动。（T）
16. 枚举类必须在第一行声明枚举类对象。（T）
17. 枚举类若枚举只有一个成员, 则可以作为一种单例模式的实现方式（T）
18. 枚举类的使用 public final 修饰的属性应该在构造器中为其赋值（F）
19. 若枚举类显式的定义了带参数的构造器, 则在列出枚举值时也必须对应的传入参数（T）
20. JDK1.5之前需要自定义枚举类（T）
21. JDK 1.5 新增的 enum 关键字用于定义（T）
22. 枚举类型不可实现其他接口。（ F ）
23. Java中，定义一个颜色的枚举类型color，public enumColor{red,green,yellow,blue;}（T）
24. 枚举有两个限制 1不能继承另一个类。2 不能是超类不能被扩展(√)
25. 6:枚举支持重载(√)
26. 枚举可以继承其它的java类。（ F ）
27. 枚举值可以改变。（ F ）
28. 枚举中不能有public的构造函数。（ T ）
29. 枚举给出了将一个任意项同另一个项相比较的能力，并且可以在一个已定义项列表中进行迭代。（ T ）

五、简答题

1、枚举的主要作用是什么？

答案：枚举的本质是类，在没有枚举之前，仍然可以按照java最基本的编程手段来解决需要用到枚举的地方。枚举屏蔽了枚举值的类型信息，不像在用public static final定义变量必须指定类型。枚举是用来构建常量数据结构的模板，这个模板可扩展。枚举的使用增强了程序的健壮性，比如在引用一个不存在的枚举值的时候，编译器会报错。在JDK5中新引入的枚举完美地解决了之前通过常量来表示离散量所带来的问题，大大加强了程序的可读性、易用性和可维护性，并且在此基础之上又进行了扩展，使之可以像类一样去使用，更是为Java对离散量的表示上升了一个台阶。因此，如果在Java中需要表示诸如颜色、方式、类别、状态等等数目有限、形式离散、表达又极为明确的量，应当尽量舍弃常量表示的做法，而将枚举作为首要的选择。

2、列举枚举重要的内置方法和作用。

int compareTo(E o)           比较此枚举与指定对象的顺序。

Class<E> getDeclaringClass()           返回与此枚举常量的枚举类型相对应的 Class 对象。

String name()           返回此枚举常量的名称，在其枚举声明中对其进行声明。

int ordinal()           返回枚举常量的序数（它在枚举声明中的位置，其中初始常量序数为零）。

String toString()           返回枚举常量的名称，它包含在声明中。

static <T extends Enum<T>> T valueOf(Class<T> enumType, String name)           返回带指定名称的指定枚举类型的枚举常量。

static T[] values() 返回该枚举的所有值。

3、请简述关于枚举的总结（5分）

答案：（1）枚举的本质是类，在没有枚举之前，仍然可以按照java最基本的编程手段来解决需要用到枚举的地方。枚举屏蔽了枚举值的类型信息，不像在用public static final定义变量必须指定类型。枚举是用来构建常量数据结构的模板，这个模板可扩展。枚举的使用增强了程序的健壮性，比如在引用一个不存在的枚举值的时候，编译器会报错。

枚举的更多用法还需要在开发中去研究创造，Java5、Java6增加了不少新的特性，技术在升级，

（2）枚举在Java家族中只占了很小的一块比重，所以我在项目中用枚举的地方不是很多，毕竟，一个项目是很多人开发维护的，用一个陌生的东西，会给其他的同事造成阅读困难。所以常量大都是用public static final 来定义的

4、为什么不用静态常量来替代枚举类呢？

答案：枚举类更加直观，类型安全。使用常量会有以下几个缺陷：

　　1. 类型不安全。若一个方法中要求传入季节这个参数，用常量的话，形参就是int类型，开发者传入任意类型的int类型值就行，但是如果是枚举类型的话，就只能传入枚举类中包含的对象。

　　2. 没有命名空间。开发者要在命名的时候以SEASON\_开头，这样另外一个开发者再看这段代码的时候，才知道这四个常量分别代表季节。

5、枚举类与普通类的区别

答案：主要区别：使用enum定义的枚举类默认继承了java.lang.Enum类

枚举类的构造器只能使用private

枚举类的每个实例必须在枚举类中显示的列出（,分隔   ；结尾） 列出的实例系统会自动添加public static final修饰

所有的枚举类都定义了一个values方法，该方法可以很方便的遍历所有的枚举值

可以在switch表达式使用枚举类对象作为表达式，case子句可以直接使用枚举的名字，无需添加枚举类作为限定

枚举类对象的属性不能更改，所以要用private final修饰

枚举类对象要在构造器中被赋值

6、请试说明枚举类型的意义，即如果有些情况用类代替枚举存在什么问题，（5分）

答案：类型不安全

没有命名空间

意义不明确

7、以下代码运行结果：

public class EnumDemo{  
public static void main(String[] args) {  
　　doAction(Action.TURN\_RIGHT);  
　　}  
  
　　public static void doAction(Action action){  
　　switch(action){  
　　case TURN\_LEFT:  
　　System.out.println("向左转");  
　　break;  
　　case TURN\_RIGHT:  
　　System.out.println("向右转");  
　　break;  
　　case SHOOT:  
　　System.out.println("射击");  
　　break;  
　　}  
　　}  
　　}

答案：向右转

8、枚举类的特点阐述一下（5分）

答案：

使用enum定义的枚举类默认继承了java.lang.Enum类

枚举类的构造器只能使用private

枚举类的每个实例必须在枚举类中显示的列出（,分隔   ；结尾） 列出的实例系统会自动添加public static final修饰

所有的枚举类都定义了一个values方法，该方法可以很方便的遍历所有的枚举值

可以在[switch](https://www.baidu.com/s?wd=switch&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)表达式使用枚举类对象作为表达式，case子句可以直接使用枚举的名字，无需添加枚举类作为限定

枚举类对象的属性不能更改，所以要用private final修饰

枚举类对象要在构造器中被赋值

9、两种方法values(),value of()的区别：

values() 返回包含一列枚举常量的数组 ， valueOf() 返回带指定名称的指定枚举类型的枚举常量。

10、请列举枚举类型几个特点。(不少于5条)（5分）

答案：1.所有的枚举都继承自java.lang.Enum类。由于Java 不支持多继承，所以枚举对象不能再继承其他类

2.不能有public的构造函数

3.所有枚举值都是public , static , final的

4.Enum默认实现了java.lang.Comparable接口

5.Enum覆载了了toString方法

6. 枚举类的所有实例必须放在第一行显示

六、程序实现题

1、创建一个表示计算机文件类型的枚举类型，该枚举需要包含一个数字型（编号）和一个字符串型（名称）的属性，并为该枚举实现一个根据整数值生成枚举值的方法

|  |
| --- |
| public enum TypeEnum {  VIDEO(1, "视频"), AUDIO(2, "音频"), TEXT(3, "文本"), IMAGE(4, "图像");    int value;  String name;    TypeEnum(int value, String name) {  this.value = value;  this.name = name;  }    public int getValue() {  return value;  }    public String getName() {  return name;  }    public static TypeEnum getByValue(int value) {  for(TypeEnum typeEnum : TypeEnum.values()) {  if(typeEnum.value == value) {  return typeEnum;  }  }  throw new IllegalArgumentException("No element matches " + value);  }  } |

2、枚举实现： 实现java代码中不同颜色的枚举常量选择（10分）

|  |
| --- |
| public class B {  public static void main(String[] args) {  showColor( Color.RED );  }  static void showColor(Color color){  switch ( color ) {  case BLANK:  System.out.println( color );  break;  case RED :  System.out.println( color );  break;  default:  System.out.println( color );  break;  }  }  } |

3、一个星期week的枚举，枚举值为英文名称的星期一到星期天，要求要用两种不同的方法来打印中文的格式的星期名称.

|  |
| --- |
| 第一种：  package com.java.xiong.test0301;  import org.junit.Test;  public class Weeks {  @Test  public void println(){  System.out.print(WeekDay.MONAY.getChinaName());  }  }  enum WeekDay{  MONAY {  @Override  public String getChinaName() {  return "星期一";  }  },TUESDAY {  @Override  public String getChinaName() {  return "星期二";  }  },WBDNESDAY {  @Override  public String getChinaName() {  return "星期三那";  }  },THURSDAY {  @Override  public String getChinaName() {  return "星期四";  }  },FRIDAY {  @Override  public String getChinaName() {  return "星期五";    }  },SATURDAY {  @Override  public String getChinaName() {  return "星期六";  }  };  public abstract String getChinaName();  }  第二种：  package com.java.xiong.test0301;  import org.junit.Test;  public class WeeksTow {  @Test  public void print() {  System.out.println(Week.FRIDAY);  }  }  enum Week {  MONDAY("星期一"), TUESDAY("星期二"), WEDEDAY("星期三"), THURSDAY("星期四"), FRIDAY(  "星期五"), SATURADAY("星期六"), SUNDAY("星期天");  private Week(String name) {  }  } |

4、定义义一个交通灯枚举类，包含红灯、绿灯、黄灯，需要有获得下一个灯的方法，并实现红灯出现5秒之后变成绿灯，绿灯3秒之后变成黄灯，黄灯2秒之后变成红灯,如此循环

public class Test5 {

/\*\*

\* @throws InterruptedException

\*/

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

Deng[] deng = Deng. values();

for( int i = 0;i < 3;i++){

Deng d = deng[i];

if(d == Deng. red){

System. out.println(d);

Thread. sleep(5000);

System. out.println(d.next());

} else if(d == Deng. green){

Thread. sleep(3000);

System. out.println(d.next());

} else if(d == Deng. yellow){

Thread. sleep(2000);

// System.out.println(d.next());

i = -1;

}

}

}

}

enum Deng{

red("红灯"){

public Deng next(){

return green;

}

},green("绿灯"){

public Deng next(){

return yellow;

}

},yellow("黄灯"){

public Deng next(){

return red;

}

};

public abstract Deng next();

private String name;

private Deng(String name){

this. name = name;

}

@Override

public String toString() {

return name;

}

}

5、编写一个根据控制台输入的大小类型，打印出对应的类型名和简称

package enums;

import java.util.\*;

public class EnumTest {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);//从键盘输入

System.out.println("Enter a size:(SMALL,MEDUIM, LARGE, EXTRA\_LARGE)");

String input = in.next().toUpperCase();

Size size = Enum.valueOf(Size.class, input);

System.out.println("size=" + size);

System.out.println("abbreviation=" + size.getAbbreviation());

if (size == Size.EXTRA\_LARGE)

System.out.println("Good job--you paid attention to the \_.");

}

}

enum Size {

SMALL("S"), MEDUIM("M"), LARGE("L"), EXTRA\_LARGE("XL");

private Size(String abbreviation) {

this.abbreviation = abbreviation;

}

public String getAbbreviation() {

return abbreviation;

}

private String abbreviation;

}

6、写一个颜色枚举，包含红色，绿色，白色，黄色，包含名称name和索引index两个属性。构造方法，set，get方法，以及一个通过索引index获取名称的方法。（10分）

|  |
| --- |
| **public** **enum** Color {  ***RED***("红色",1),***GREEN***("绿色",2),***BLANK***("白色",3),***YELLOW***("黄色",4);    **private** String name;  **private** **int** index;    **private** Color(String name, **int** index) {  **this**.name = name;  **this**.index = index;  }    **public** String getName(**int** index) {  **for** (Color c : Color.*values*()) {  **if**(c.index==index) {  return c.name;  }  }  return null;  }    **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** **int** getIndex() {  **return** index;  }  **public** **void** setIndex(**int** index) {  **this**.index = index;  }    } |

7、写一个季节枚举，含有属性name

|  |
| --- |
| public enum SeasonEnum {  SPRING("春天"),SUMMER("夏天"),FALL("秋天"),WINTER("冬天");    Private String name;    private SeasonEnum(String name){  this.name = name;  }  public String getName() {  return name;  }  } |