

## Java 就业班入学测试题

### 第一部分：选择题（共计 45 题）

1、阅读以下代码，并给出正确答案（ ）

```
public static void main(String[] args) {  
    List<String> list = new ArrayList<>();  
    list.add("sh");  
    list.add("itcast");  
    for (String str : list)  
        if (str.equalsIgnoreCase("SH")) {  
            list.add("传智播客");  
        }  
    list.add("cn");  
}
```

- A、执行完后，list 大小为 4
- B、程序运行报错
- C、执行完后，list 大小为 3
- D、执行完后，list 大小为 5

2、阅读以下代码，并给出正确答案（ ）

```
class Father{  
    public void method(int num) {  
        num++;  
    }  
}  
class Child extends Father{  
    protected void method(int num) {  
        super.method(num);  
        System.out.println(num++);  
    }  
}  
class Test{  
    public static void main(String[] args){  
        Child c = new Child();  
        c.method(10);  
    }  
}
```

- A、程序编译报错
- B、程序输出 num 的结果为：10
- C、程序输出 num 的结果为：12
- D、程序输出 num 的结果为：11

3、以下程序的执行结果是（ ）

```
class Father{
    public Father()
    {
        System.out.println("父类无参构造函数");
    }
    public Father(String name)
    {
        System.out.println("父类有参构造函数");
    }
}
class Child extends Father{
    public Child()
    {
        this("dd");
        System.out.println("子类无参构造函数");
    }
    public Child(String name){
        super("dd");
        System.out.println("子类有参构造函数");
    }
}
public class Test{
    public static void main(String [] args)
    {
        Child c = new Child();
    }
}
```

- A、子类无参构造函数 子类有参构造函数 父类无参构造函数
- B、子类有参构造函数 子类无参构造函数 父类无参构造函数
- C、父类无参构造函数 子类有参构造函数 子类无参构造函数
- D、父类有参构造函数 子类有参构造函数 子类无参构造函数

4、以下程序运行后的结果是（ ）

```
public class ArrayDemo{
    public static void main(String[] args)
    {
        int[] array={110,130,133,150};
        List list=Arrays.asList(array);
        list.add(180);
        System.out.println(list);
    }
}
```

- A、[110,130,133,150,180]
- B、{110,130,133,150,180}
- C、程序运行报错
- D、110 130 133 150 180

5、以下程序说法正确的是（ ）

```
public static void main(String[] args){
    ArrayList al=new ArrayList();
    al.add("sh");
    al.add(123);
    al.add(1,"itcast");
    al.add(4,"cn");
    System.out.println(al);
}
```

- A、程序执行完后输出容器中的元素
- B、容器中只能添加相同类型的元素
- C、容器中只能添加引用类型的元素，添加 123 时会报错
- D、程序运行会报错不会输出容器中的元素

6、阅读以下代码，在空缺位置 1 可以填补的代码有（ ）[选两项]

```
public static void main(String[] args) throws ParseException {
    //位置1
    String strDate="2015年2月27日";
    Date date=df.parse(strDate);
    System.out.println(date);
}
```

- A、DateFormat df = new DateFormat(DateFormat.LONG);
- B、DateFormat df = DateFormat.getDateInstance(DateFormat.LONG);
- C、DateFormat df = new SimpleDateFormat(DateFormat.LONG);
- D、DateFormat df = new SimpleDateFormat("yyyy 年 MM 月 dd 日");

7、程序出现以下错误，说法正确的是（ ）[选两项]

```
E:\JavaSE0515\code\day08_code>javac PersonDemo2.java
PersonDemo2.java:47: 错误: 无法将类 Person 中的构造器 Person 应用到给定类型;
    Person p = new Person<>;
                  ^
需要: String,int
找到: 没有参数
原因: 实际参数列表和形式参数列表长度不同
1 个错误
```

- A、缺少 Person 类，所以编译时报错
- B、Person 类是私有的类，PersonDemo2 类中不能创建 Person 类的对象
- C、Person 类中没有无参数的构造器
- D、在创建 Person 类对象时，需要给构造器中指定实参

8、以下程序运行后的结果是（ ）

```
public static void main(String[] args)
{
    String s1="a";
    String s2="b";
    String s3="ab";
    String s4="ab";
    System.out.println(s3==s4);
    String s5="a"+"b";
    System.out.println(s3==s5);
    String s6=s1+s2;
    System.out.println(s3==s6);
    String s7=new String("ab");
    System.out.println(s3==s7);
    final String s8="a";
    final String s9="b";
    String s10=s8+s9;
    System.out.println(s3==s10);
}
```

- A、 false true true false false
- B、 true false false true false
- C、 true true false false true
- D、 false false true false true

9、阅读以下代码，并给出正确答案（ ）

```
class Demo
{
    public static void main(String[] args)
    {
        int x = 3;
        int y = 4;
        int sum = getSum(x,y);
        System.out.println(sum);
    }
    public static double getSum(byte x , byte y)
    {
        return x+y;
    }
}
```

- A、 编译报错
- B、 程序运行并输出 7
- C、 程序运行并输出 0
- D、 程序运行并输出 7.0

10、阅读以下程序，说法正确的是（ ）

```
class Demo
{
    Demo d = new Demo();
    public static void main(String[] args)
    {
        Demo d2 = new Demo();
    }
}
```

- A、程序正常运行，但没有结果
- B、编译报错
- C、程序运行后出现内存错误
- D、程序运行会死循环

11、阅读以下代码，在 main 方法中可以执行 show 方法的代码有（ ）[选两项]

```
class Demo
{
    int num = 111;

    public void show()
    {
        System.out.println("num="+num);
    }

    public static void main(String[] args)
    {
    }
}
```

- A、Demo d=new Demo(); d.show();
- B、show();
- C、Demo.show();
- D、new Demo().show();

12、java 中类的加载过程，按照加载的顺序正确的是（ ）

- ① 默认初始化静态成员变量
- ② 执行静态代码块
- ③ 加载非静态的成员到非静态区域，加载静态成员到静态区域
- ④ 对静态成员变量进行显式初始化

- A、1, 2, 4, 3
- B、4, 3, 2, 1
- C、3, 1, 4, 2
- D、3, 4, 1, 2

13、下面哪些函数是 public void aMethod(){...}的重载函数（ ）

- A、void aMethod(){...}
- B、public int aMethod(int a, float b){...}
- C、public void aMethod (){...}

D、private float aMethod () {...}

14、阅读以下代码，并给出正确的输出结果 ( )

```
public static void main(String[] args) {  
  
    int x = 4;  
    int y = 5;  
    if(x++>4 & ++y>5) {  
        x++;  
    }  
    System.out.println("x=" + x + ",y=" + y);  
}
```

- A、x=5,y=6
- B、x=4,y=6
- C、x=4,y=5
- D、x=6,y=6

15、在 Java 中，下列语句不能通过编译的有( )

- A、String s= "join" + 3;
- B、int i= "a" +3;
- C、short a = 'a' +5;
- D、float f = 5+5.5f;

16、阅读以下代码，并给出正确答案 ( )

```
public static void main(String args[]){  
    char digit = 'a';  
    for (int i = 0; i < 10; i++){  
        switch (digit)  
        {  
            case 'x' :  
            {  
                int j = 0;  
                System.out.println(j);  
            }  
            default :  
            {  
                int j = 100;  
                System.out.println(j);  
            }  
        }  
    }  
    int i = j;  
    System.out.println(i);  
}
```

- A、输出 11 次 100
- B、输出 10 次 100，然后抛出运行期异常
- C、编译错误，因为变量 i 不能在 main() 方法中被声明 2 次
- D、编译错误，因为变量 j 不能在 main() 方法中被声明 2 次

E、以上都不对

17、阅读以下代码，并给出正确答案（ ）

```
Set set= new HashSet();  
set.add("aaa");  
set.add("bbb");  
set.add("aaa");  
System.out.println(set.size());
```

- A、编译不通过
- B、运行时出错
- C、输出 2
- D、输出 3

18、程序出现以下错误，说法正确的是（ ）

```
E:\JavaSE0515\code\day08_code>javac PersonDemo.java  
PersonDemo.java:45: 错误: age可以在Person中访问private  
    p.age = -22;  
      ^  
1 个错误
```

- A、Person 类中没有 age 成员
- B、不能给 Person 类中的成员赋值为负数
- C、应该把 p.age=-22 修改为 age=-22
- D、Person 类中的 age 是私有的，不能对 age 赋值

19、下列关于 java 中抽象类的说法正确的有（ ）

- A、含有抽象方法的类必须声明为抽象类
- B、抽象类中不可以有非抽象方法
- C、抽象类中至少需要包含一个抽象方法
- D、抽象类无法实例化

20、对象的创建流程，正确的是（ ）

- ① 调用构造方法
  - ② 显式初始化所有成员变量
  - ③ 开辟堆中的空间
  - ④ 调用父类构造方法
  - ⑤ 默认初始化所有成员变量
  - ⑥ 执行构造代码块
- A、3, 1, 5, 2, 6, 4
  - B、4, 3, 1, 5, 2, 6
  - C、3, 4, 1, 6, 5, 2
  - D、3, 5, 1, 4, 2, 6

E、4, 1, 3, 5, 6, 2

21、编译并运行以下代码的输出结果是 ( )

```
String s1 = new String("amit");
System.out.println(s1.replace('m', 'r'));
System.out.println(s1);
String s3 = "arit";
String s4 = "arit";
String s2 = s1.replace('m', 'r');
System.out.println(s2 == s3);
System.out.println(s3 == s4);
```

- |        |        |        |        |
|--------|--------|--------|--------|
| A、arit | B、arit | C、amit | D、arit |
| amit   | arit   | amit   | amit   |
| false  | false  | false  | true   |
| true   | true   | true   | true   |

22、编译并运行以下代码将发生什么 ( )

```
class MyClass {
    int x;
    MyClass(int i) {
        x = i;
    }
    public static void main(String args[]) {
        MyClass m1 = new MyClass(100);
        MyClass m2 = new MyClass(100);
        if (m1.equals(m2)) {
            System.out.println("Both are equal");
        } else {
            System.out.println("Both are not equal");
        }
    }
}
```

- A、代码编译时报出错误提示信息“equals() 方法未定义”
- B、编译通过，抛出运行期异常
- C、输出 Both are equal
- D、输出 Both are not equal

23、Socket 类哪个方法返回 Socket 对象绑定的本地端口 ( )

- A、getPort()
- B、getLocalPort()
- C、getRemotePort()
- D、不存在这个样的方法，因为 SocketChannel 对象绑定的本地端口对程序是透明的

24、阅读以下代码，在空缺位置填补可以实现懒汉式单例的代码是 ( )



```
public class Single{
    private Single(){
    }
    //位置1
    public static Single getInstance(){
        //位置2
        return instance;
    }
}
```

- A、位置 1 : Single instance=new Single();  
位置 2 : if(instance==null) instance = new Single();
- B、位置 1 : public Single instance=new Single();  
位置 2 : 不书写代码;
- C、位置 1 : private static Single instance=null;  
位置 2 : if(instance ==null) instance = new Single();
- D、位置 1 : Single instance=null;  
位置 2 : instance = new Single();

25、想要把获取到的数据有序的使用集合存储起来，在存储数据都存储在集合的末尾，并在遍历集合的过程中对最先存储的数据逐一获取并删除，这时需要使用以下哪个集合进行存储比较适合（ ）

- A、Properties
- B、LinkedList
- C、ArrayList
- D、LinkedHashSet

26、阅读以下代码，要实现队列结构在空缺位置可以填补的代码有（ ） [选两项]

```
public class Queue
{
    private LinkedList element=null;
    public Queue(){
        element = new LinkedList();
    }
    public void addElement(Object obj){
        //位置1
    }
    public Object getElement(){
        //位置2
    }
    public boolean isNull(){
        //位置3
    }
}
```

- A、位置 1 : element.add(obj);  
位置 2 : return element;  
位置 3 : return element.isEmpty;
- B、位置 1 : element.addFirst(obj);

位置 2 : return element.getLast();

位置 3 : return element.size==0;

C、位置 1 : element.addLast(obj);

位置 2 : return element.getFirst();

位置 3 : return element.isEmpty

D、位置 1 : element.addFirst(obj);

位置 2 : return element.getFirst();

位置 3 : return element.size==0;

27、想要使用 IO 流来读取 UTF-8 格式的字符文件需要使用以下哪个流实现 ( )

A、BufferedInputStream

B、FileReader

C、FileInputStream

D、InputStreamReader

28、阅读以下代码，在空缺位置填补 ( ) 代码可以实现字符缓冲区读取文件的功能

```
FileInputStream fis=new FileInputStream("d:\\temp.txt");  
//位置1  
BufferedReader br=new BufferedReader (/*位置2*/);  
String line=null;  
while ((line=br.readLine())!=null) {  
    System.out.println(line);  
}  
br.close();
```

A、位置 1 : FileReader fr=new FileReader(fis);

位置 2 : fr

B、位置 1 : InputStreamReader isr=new InputStreamReader(fis);

位置 2 : isr

C、位置 1 : BufferedInputStream bis=new BufferedInputStream(fis);

位置 2 : bis

D、位置 1 : OutputStreamReader osr=new OutputStreamReader(fis);

位置 2 : osr

29、以下字符串格式中的信息，如果想要使用集合来存储，哪个集合比较适合 ( )

```
"[{name="张三",age=20,address="上海"},{name="李四",age=22,address="北京"}]"
```

A、LinkedList

B、TreeSet

C、ArrayList

D、HashMap

30、下列哪个关键字可以用于实现同步 ( )

- A、native
- B、static
- C、synchronized
- D、finalize

31、下列哪些方法是 ArrayList 和 LinkedList 集合中都定义的 ( ) [选两项]

- A、add(Object o)
- B、removeFirst()
- C、element()
- D、add(int index, Object o)

32、阅读以下代码，并给出正确答案 ( )

```
public static void main(String[] args) {  
    Set set=new HashSet();  
    set.add("上海");  
    set.add("北京");  
    set.add("广州");  
    for (ListIterator it = set.listIterator(); it.hasNext();) {  
        String str=(String)it.next();  
        if(str.equals("北京")){  
            set.remove("北京");  
        }  
    }  
}
```

- A、程序正常执行，从 set 集合中删除元素“北京”
- B、程序运行报错，在迭代过程中不能对集合进行删除
- C、程序编译报错，set 集合不能使用 ListIterator 迭代器
- D、以上说法都不正确

33、下列哪些集合属于 Collection 的子类 ( ) [选两项]

- A、TreeMap
- B、Hashtable
- C、ArrayList
- D、HashSet

34、阅读以下代码，并给出正确答案 ( )

```
public class Test{
    public static int k = 0;
    {
        print("c_code");
    }
    static {
        print("s_code");
    }
    public static int print(String s){
        System.out.println("s="+s+", k="+k);
        ++k;
        return k;
    }
    public Test(String string){
        print(string);
    }
    public static void main(String[] args){
        Test d=new Test("T");
    }
}
```

- A、s=T,k=0 s=s\_code,k=1 s=c\_code,k=2
- B、s=s\_code,k=0 s=c\_code,k=1 s=T,k=2
- C、s=c\_code,k=0 s=s\_code,k=1 s=T,k=2
- D、s=T,k=0 s=c\_code,k=2 s=s\_code,k=2

35、HashSet 集合不能存放重复元素是通过下面那种方式判断的 ( )

- A、使用 hashCode 方法
- B、使用 equals 方法
- C、使用 compareTo 方法
- D、使用 compare 方法

36、下面关于 Java 中的 String 类说法正确的是 ( )

- A、使用 deleted 方法可以删除字符串中的指定字符
- B、字符串一旦初始化就不可以被改变
- C、调用 String 类的 reverse 方法可以将字符串数据进行反转
- D、创建 String 对象的时，可以直接传给构造器一个 Collection 集合，从而将集合转成字符串
- E、可以自定义个类继承 String 类，增加特有方法

37、下面关于线程的说法错误的是 ( )

- A、线程其实就是进程中的一个控制单元，它负责就是程序的执行。一个进程中至少有一个线程
- B、当一个类实现了 Runnable 接口后，并实现了其 run 方法，就可以直接调用这个类的 start 方法开启线程
- C、继承 Thread 类或者实现 Runnable 接口都可以封装线程要执行的任务
- D、Thread 类本身就是一个线程类，可以直接创建 Thread 类对象，开启线程

38、下面关于 map 集合说法正确的是 ( )

- A、map 集合中不能直接使用 Iterator 进行迭代
- B、对 map 集合使用 keySet 方法，会得到所有 value 的值组成一个 list 集合
- C、使用 map 的 append 方法可以向 map 集合中添加元素
- D、使用 removeAll 方法可以将 map 集合中的元素清空

39、下面的程序执行后 count 的结果是 ( )

```
class Client{
    public static void main(String[] args){
        int count=0;
        for(int i=0;i<10;i++) {
            count=count++;
        }
    }
}
```

- A、0                      B、10    C、9                      D、11

40、阅读以下代码，并给出正确答案 ( )

```
public class Test {
    public static void main(String argv[]) {
        ThreadChild t1 = new ThreadChild("one");
        t1.run();
        ThreadChild t2 = new ThreadChild("two");
        t2.run();
    }
}
class ThreadChild extends Thread {
    private String sname = "";
    ThreadChild(String s) {
        sname = s;
    }
    public void run() {
        for (int i = 0; i < 2; i++) {
            try {
                sleep(1000);
            } catch (InterruptedException e) {
            }
            yield();
            System.out.println(sname);
        }
    }
}
```

- A、编译错误
- B、输出 One One Two Two
- C、输出 One Two One Two
- D、输出 One Two One Two

41、阅读以下代码，并给出正确答案 ( ) [选两项]

```

1 class Test{
2     public static void main(String[] args)
3     {
4         int[] arr=new int[3];
5         for(int i=0;i<arr.length;i++){
6             arr[i]=i+1;
7         }
8         System.out.println("arr[4]="+arr[4]);
9         arr=null;
10        System.out.println("arr="+arr);
11        System.out.println("arr[0]"+arr[0]);
12    }
13 }

```

- A、在程序第 8 行出现运行时异常 ArrayIndexOutOfBoundsException
- B、程序正常执行
- C、在程序第 9 行不能把 null 赋值给 arr 数组
- D、在程序第 11 行出现运行时异常 NullPointerException

42、阅读以下代码，在空缺位置填补什么异常可以实现方法的重写 [选两项]

```

class Father{
    public void fun(String str) throws NullPointerException,IOException,ParseException{
        System.out.println(str);
    }
}
class Child extends Father{
    public void fun(String str) throws /*空缺位置*/{
        System.out.println("Child run...");
    }
}

```

- A、DataFormatException
- B、Exception
- C、FileNotFoundException
- D、NullPointerException

43、当已经创建了 Socket 对象 s 来连接到远程服务器，下列哪个是最合适的方法来立即从 socket 读取 ASCII 数据 ( )

- A、InputStream in = s.getInputStream();
- B、DataInputStream in = new DataInputStream(s.getInputStream());
- C、ByteArrayInputStream in = new ByteArrayInputStream(s.getInputStream());
- D、BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()), "ios-8859-1" );
- E、BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));

44、关于被私有访问控制符 private 修饰的成员变量，以下说法正确的是 ( )

- A、可以被三种类所引用：该类自身、与它在同一个包中的其他类、在其他包中的该类的子类

- B、可以被两种类访问和引用：该类本身、该类的所有子类
- C、只能被该类自身所访问和修改
- D、只能被同一个包中的类访问

45、设有定义 `int i = 6` ;则执行以下语句后 `i` 的值为 ( )

`i+=i-1;`

- A、10            B、9        C、11            D、12

## 第二部分：代码补全题（共计 3 题）

1、阅读以下程序，要实现 IO 流读写图片文件，在空缺位置补全代码

```
public static void main(String[] args) throws IOException {
    FileInputStream fis=new FileInputStream("d:\\图片.jpg");
    FileOutputStream fos=new FileOutputStream("E:\\image.jpg");
    int len=0;
    byte[] buf=new byte[1024];
    while(/*填补位置1*/) {
        //填补位置2
    }
    fis.close();
    fos.close();
}
```

填补位置 1：\_\_\_\_\_

填补位置 2：\_\_\_\_\_

2、阅读以下程序，在空缺位置补全代码 [提示：Child 类中只需要 InterA 接口中的其中两个函数功能]

```
interface InterA{
    void method_1();
    void method_2();
    void method_3();
    void method_4();
}
/*填补位置1*/ AbsInterA implements InterA{
    //填补位置2
}
class Child extends AbsInterA
{
    public void method_2(){
        System.out.println("method_2 run...");
    }
    public void method_4(){
        System.out.println("method_4 run...");
    }
}
```

填补位置 1：\_\_\_\_\_

填补位置 2：\_\_\_\_\_

---

---

---

### 3、阅读以下程序，根据需求在空缺位置补全代码

```
class Person{
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
}
class Student extends Person{
    public Student(String name, int age) {
        super(name, age);
    }
}
class Worker extends Person{
    public Worker(String name, int age) {
        super(name, age);
    }
}
class Test{
    public static void print(/*填补位置1*/< /*填补位置2*/> c){
        System.out.println(c);
    }
}
```

需求 1：print 方法中的参数只能是 List 集合，而 List 集合中只能存储 Person 的子类对象

填补位置 1：\_\_\_\_\_

填补位置 2：\_\_\_\_\_

需求 2：print 方法中的参数可以是 List 集合也可以是 Set 集合，而集合中存储的对象却只能是 Student 或 Person 对象

填补位置 1：\_\_\_\_\_

填补位置 2：\_\_\_\_\_