1、

使用this调用类的构造方法，下面的说法不正确的是？

2分

* A、使用this调用构造方法的格式为this([参数1,参数2…])
* B、只能在构造方法中使用this调用其它的构造方法
* C、使用this调用其它构造方法的语句必须放在第一行.
* D、能在一个类的两个构造方法中使用this互相调用

2、

在以下什么情况下，构造方法会被调用？

2分

* A、类定义时
* B、创建对象时
* C、调用对象方法时
* D、使用对象的变量时

3、

要想保存具有映射关系的数据，可以使用以下哪些集合？

2分

* A、ArrayList
* B、TreeMap
* C、List
* D、TreeSet

4、

下面（  ）不是String类提供的合法的方法

2分

* A、equals()
* B、append()
* C、indexOf()
* D、trim()

5、

关于super关键字以下说法哪些是错误的？

2分

* A、super关键字可以调用父类的构造方法
* B、super关键字可以调用父类的普通方法
* C、super与this不能同时存在于同一个构造方法中
* D、super与this可以同时存在于同一个构造方法中

6、

阅读下面的程序：

public class test {

      public static void main(String args[]) {

         int i;

         float  f = 2.3f;

         double d = 2.7;

         i = ((int)Math.ceil(f)) \* ((int)Math.round(d));

        System.out.println(i);

        }

     }

2分

* A、9
* B、5
* C、6
* D、6.1

7、

类中的一个成员方法被下面哪个修饰符修饰，该方法只能在本类被访问？

2分

* A、public
* B、 protected
* C、 private
* D、default

8、

把当前时间转换为字符串“2021-12-27”的形式，下列选项正确的是

2分

* A、
* SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
* Date date = new Date();
* String dateStr = sdf.format(date);
* B、
* SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
* Date date = new Date();
* String dateStr = sdf.parse(date);
* C、
* SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
* Date date = new Date();
* String dateStr = sdf.format(date);
* D、
* SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
* Date date = new Date();
* String dateStr = sdf.parse(date);

9、

在Java中，以下变量赋值无法通过编译的是（）

2分

* A、.char c1 = "男";
* B、char c2 = '女';
* C、int f1 = 128;
* D、double d1 = 1.2;

10、

假若indexOf()方法未能找到所指定的子字符串，则返回以下选项中的哪个？

2分

* A、 false
* B、 0
* C、-1
* D、以上答案都不对

11、

String s ="hello";

String t = "hello";

char c[] ={'h','e','l','l','o'};

下列表达式错误的是（）

2分

* A、s.equals(t)
* B、s==t
* C、t.equals(new String("hello"))
* D、t.equals(c)

12、

以下说法哪些是错误的？

2分

* A、Java语言中允许一个类实现多个接口
* B、Java语言中不允许一个类继承多个类
* C、Java语言中允许一个类同时继承一个类并实现一个接口
* D、Java语言中不允许一个接口继承一个接口

13、

使用Iterator时，判断是否存在下一个元素可以使用以下哪个方法？

2分

* A、next()
* B、hash()
* C、hasPrevious()
* D、hasNext()

14、

Map集合添加元素的方式是（）

2分

* A、add()
* B、push()
* C、put()
* D、pop()

15、

已知类的继承关系如下：

class Employee;

class Manager extends Employee;

class Director extends Employee;

则以下语句能通过编译的有哪些

2分

* A、 Employee e=new Manager();
* B、Director d=new Manager();
* C、Director d=new Employee();
* D、Manager m=new Director();

16、

请看下列代码：

public static void main(String[] args) {

<插入代码>

set.add("D");

set.add("C");

set.add("A");

set.add("B");

System.out.println(set);

}

如果输出结果是[A, B, C, D]，那么<插入代码>出应填入的代码是：

2分

* A、Set set = new TreeSet();
* B、Set set = new HashSet();
* C、Set set = new SortedSet();
* D、List set = new SortedList();

17、

获取单列集合中元素的个数可以使用以下哪个方法？

2分

* A、length()
* B、size()
* C、get(int index)
* D、add(Object obj)

18、

在程序开发中，经常会使用以下哪个类来存储程序中所需的配置？

2分

* A、HashMap
* B、TreeSet
* C、Properties
* D、TreeMap

### **多选题**

19、

下面选项中，对Math类描述正确的是（）

2分

* A、Math类中所有的方法都是静态的
* B、Math类中的方法可以使用类名.方法名的方式调用
* C、Math类的构造函数是私有的
* D、
* 无法使用Math类创建对象

20、

下列关于封装的描述中，正确的是（ ）

2分

* A、方法和类都可以称为封装体
* B、封装隐藏了程序的实现细节，同时对外提供了特定的访问方式
* C、封装能提高代码的复用性
* D、以上说法均错误

21、

下面关于静态变量的描述正确的是（ ）

2分

* A、静态变量可以通过类名直接调用，因此也称类变量
* B、静态变量可以在非静态方法中使用
* C、静态变量不可以通过类名直接调用
* D、成员变量就是静态变量

22、

下列关于String类和StringBuffer类的说法中，正确的是（ ）

2分

* A、String类表示的字符串是常量，一旦创建后，内容和长度都是无法改变的。
* B、StringBuffer表示字符容器，其内容和长度都可以随时修改
* C、String类对象可以用操作符“+”进行连接。
* D、StringBuffer类对象之间能用"+"进行连接

23、

以下关于foreach说法中，哪些是正确的（ ）

2分

* A、foreach是一种更加简洁的for循环，也称增强for循环
* B、foreach可以对List
* C、Set数组进行遍历
* D、foreach的出现，简化了书写

24、

下面数据类型属于基本类型的有（）

2分

* A、int
* B、short
* C、byte
* D、
* char

25、

下面属于集合的有()

2分

* A、TreeSet
* B、ArrayList
* C、
* HashSet
* D、
* LinkedList

### **判断题**

26、

在定义一个类的时候，如果类的成员被private所修饰，该成员不能在类的外部被直接访问。

1分

* 
* 

27、

类中static修饰的变量或方法，可以使用类名或对象的引用变量访问。

1分

* 
* 

28、

声明构造方法时，不能使用private关键字修饰。

1分

* 
* 

29、

Java中的每个类都至少有一个构造方法，一个类中如果没有定义构造方法，系统会自动为这个类创建一个默认的构造方法。

1分

* 
* 

30、

Set集合是通过键值对的方式来存储对象的。

1分

* 
* 

31、

抽象方法必须定义在抽象类中，所以抽象类中的方法都是抽象方法。

1分

* 
* 

32、

使用Iterator迭代集合元素时，可以调用集合对象的方法增删元素

1分

* 
* 

33、

集合中不能存放基本数据类型，而只能存放引用数据类型。

1分

* 
* 

### **填空题**

34、

在一个类中，除了可以定义属性、方法，还可以定义类，这样的类被称为\_\_\_\_\_\_。



1分

35、

定义一个Java类时，如果前面加使用\_\_\_\_\_\_关键字修饰，那么该类不可以被继承。



1分

36、

JDK中提供了一系列可以存储任意对象的类，统称为      。



1分

37、

所谓类的封装是指在定义一个类时，将类中的属性私有化，即使用\_\_\_\_\_\_关键字来修饰。



1分

38、

在Java中，要想让一个类继承另一个类，可以使用关键字\_\_\_\_



1分

39、

被static关键字修饰的成员变量被称为\_\_\_\_\_\_，它可以被该类所有的实例对象共享。



1分

40、

Java的跨平台性，是通过不同系统都有对应的实现的。

1分

41、

java.exe程序是用来解析文件的。

1分

42、

System.out.println(“5 + 5 = ” + 5 + 5);语句的输出结果是。

1分

43、

 在非静态成员方法中，可以使用关键字\_\_\_\_\_\_访问类的其它非静态成员。



1分

44、

如果子类想使用父类中的成员，可以通过关键字\_\_\_\_\_\_引用父类的成员。



1分

45、

 在Java中，可以使用关键字\_\_\_\_\_\_来创建类的实例对象。



1分

### **简答题**

46、

**阅读下面的程序，分析代码是否能编译通过，如果能编译通过，请列出运行的结果。如果不能编译通过，请说明原因。**

**import java.util.\*;**

**import java.util.Map.\*;**

**public class Test04 {**

**public static void main(String[] args) {**

**Map map = new HashMap();**

**map.put(1, "Tom");**

**map.put(2, "Lucy");**

**map.put(3,"Annie");**

**Set keySet = map.keySet();**

**Iterator it = keySet.iterator();**

**while (it.hasNext()) {**

**Object key = it.next();**

**System.out.println(key);**

**map.remove(key);**

**}**

**}**

**}**

5分

我的答案：

47、

# int和Integer有什么区别？如何相互转换呢？

5分

48、

用LinkedList实现一个MyStack，实现以下功能：

（1）方法：push(),pop（），isEmpty()方法;其中push()实现向栈中加入一个元素，pop()弹出最顶元素, isEmpty()判断栈是否为空。

（2）在自定义的栈中用push()方法添加依次添加”hello”,”world”,”java”元素，按照栈（后进先出）的方式在控制台中输出元素。

5分

我的答案：

      49、

阅读下面的程序，分析代码是否能编译通过，如果能编译通过，请列出运行的结果。如果不能编译通过，请说明原因。

interface Animal {

    void breathe();

    void run();

    void eat(){};

}

class Dog implements Animal {

    public void breathe() {

    System.out.println("I’m breathing")

}

    public void eat() {

    System.out.println("I’m eathing")

}

}

public class test04 {

    public static void main(String [] args) {

      Dog dog = new Dog();

      dog.breathe();

      dog.eat();

}

}

5分

我的答案：

50、

Java中的泛型是什么 ? 使用泛型的好处是什么? 并用泛型创建存储String类型ArrayList集合。

5分

我的答案：

51、

简述Java的多态性，并举例说明。

5分

我的答案：