附录

综合测试

测试试卷一

1. 下面关于 Java 的特点不正确的一项是（　　）。
   A. Java 具备跨平台性，可以在任意的操作系统间进行移植
   B. Java 编写的程序可以直接解释执行，属于解释型的编程语言类型
   C. Java 中具备垃圾收集机制，这样在用户编写代码中无须处理手工处理内存空间的释放操作
   D. Java EE 企业级开发是在 Java SE 基础之上的扩展应用
2. 下面（　　）类型不属于 Java 的基本数据类型。
   A. byte　　B. int　　C. boolean　　D. String
3. 下面（　　）属性与 Java 解释程序有关。
   A. CLASSPATH　B. GC　　C. TMP　　D. CPU
4. 下面关于 Java 程序编写描述正确的一项是（　　）。
   A. Java 程序直接利用 javac.exe 命令就可以直接运行程序
   B. 一个 Java 文件中可以定义有多个 class 声明，并且类名称可以与文件名称同名
   C. 一个 Java 文件可以使用 public class 定义多个程序类
   D. Java 文件的后缀必须使用“*.javac”
5. 下面（　　）不属于 Java 语言。
   A. // 注释　　B. -- 注释　　C. /**注释..*/　　D. /* 注释..*/
6. 下面（　　）标识符不符合 Java 定义要求。
   A. String　　B. _Name　　C. Name123　　D. 100Book
7. 下面（　　）关键字（保留字）属于 Java 未被使用到的关键字（保留字）。
   A. final　　B. goto　　C. enum　　D. assert
8. 下面关于基本数据类型的描述正确的是（　　）。
   A. boolean 数据类型只有 true 和 false 两种取值
   B. 使用 long 可以保存小数
   C. float 数据类型可以保存的数据范围比 double 数据范围要大
   D. byte 数据类型可以正常保存 200 这个数字
9. main()方法的返回值类型是（　　）。
   A. void　　B. int　　C. public　　D. static

10. 现在有一个方法：public static int info(int x,double y)，下面（　　）方法是对本方法的正确重载。

A. public static int infos(int x,int y)　　B. public static void info(int x,double y)

C. public static int info(int x,int y)　　D. public static void infos(int x,int y)

11. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        long num = 100 ;
        int x = num + 2 ;
        System.out.println(x) ;
    }
}
```

最终程序的执行结果是（　　）。

A. 102　　B. 1002　　C. 100　　D. 程序错误

12. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int num = 2147483647 ;
        num += 2 ;
        System.out.println(num) ;
    }
}
```

以上程序最终的执行结果是（　　）。

A. -2147483648　B. 2147483649　C. -2147483647　D. 2

13. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int num = 2147483647 ;
        num += 2L ;
        System.out.println(num) ;
    }
}
```

以上程序最终的执行结果是（　　）。

A. -2147483648　B. 2147483649　C. -2147483647　D. 2

14. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int num = 2147483647 ;
        long temp = num + 2L ;
        System.out.println(num) ;
    }
}
```

以上程序最终的执行结果是（　　）。

A. -2147483648　B. 2147483649　C. 2147483647　D. 2

15. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int num = 68 ;
        char c = (char) num ;
        System.out.println(c) ;
    }
}
```

以上程序最终的执行结果是（　　）。

A. B　B. C　C. D　D. a

16. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int num = 50 ;
        num = num ++ * 2 ;
        System.out.println(num) ;
    }
}
```

以上程序最终的执行结果是（　　）。

A. 50　B. 102　C. 100　D. 101

17. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int sum = 0 ;
        int x = 10 ;
        while (x > 0) {
            sum += x ;
        }
        System.out.println(sum) ;
    }
}
```

以上程序的最终执行结果是（　　）。

A. 55　B. 10　C. 程序错误，死循环　D. 15

18. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int sum = 0 ;
        for (int x = 0 ; x < 10 ; x ++) {
            sum += x ;
        }
```

```
        System.out.println(sum) ;
    }
}
```

以上程序的最终执行结果是（　　）。

A. 6　　B. 0　　C. 程序错误，死循环　D. 45

19. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int sum = 0 ;
        for (int x = 0 ; x < 10 ; x ++) {
            sum += x ;
            if (x % 3 == 0) {
                break ;
            }
        }
        System.out.println(sum) ;
    }
}
```

以上程序的最终执行结果是（　　）。

A. 6　　B. 0　　C. 程序错误，死循环　D. 45

20. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int sum = 0 ;
        for (int x = 1 ; x < 10 ; x ++) {
            sum += x ;
            if (x % 3 == 0) {
                continue ;
            }
        }
        System.out.println(sum) ;
    }
}
```

以上程序的最终执行结果是（　　）。

A. 6　　B. 0　　C. 程序错误，死循环　D. 45

21. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        char c = 'A' ;
        int num = 10 ;
        switch(c) {
            case 'B' :
                num ++ ;
```

```
            case 'A' :
                num ++ ;
            case 'Y' :
                num ++ ;
                break ;
            default :
                num -- ;
        }
        System.out.println(num) ;
    }
}
```

以上程序的最终执行结果是（　　）。

A. 11　　B. 13　　C. 12　　D. 10

22. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        System.out.println(inc(10) + inc(8) + inc(-10)) ;
    }
    public static int inc(int temp) {
        if (temp > 0) {
            return temp * 2 ;
        }
        return -1 ;
    }
}
```

以上程序的最终执行结果是（　　）。

A. 35　　B. 8　　C. 28　　D. 12

23. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        String str = "" ;
        for (int x = 0 ; x < 5 ; x ++) {
            str += x ;
        }
        System.out.println(str) ;
    }
}
```

以上程序最终的执行结果是（　　）。

A. 01234　　B. 10　　C. 14　　D. 25

24. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        int x = 10 ;
```

```
        double y = 20.2 ;
        long z = 10L;
        String str = "" + x + y * z ;
        System.out.println(str) ;
    }
}
```

以上程序的最终执行结果是（　　）。

A. 10202.0　　B. 0212.0　　C. 302.0　　D. 1020.210

25. 现在假设有如下程序：

```
public class Demo {
    public static void main(String args[]) {
        boolean flag = 10%2 == 1 && 10 / 3 == 0 && 1 / 0 == 0 ;
        System.out.println(flag ? "mldn" : "yootk") ;
    }
}
```

以上程序的最终执行结果是（　　）。

A. mldn　　B. yootk　　C. true　　D. 程序出错

26. 编译 Java 源程序文件产生的字节码文件的扩展名为（　　）。

A. java　　B. class　　C. html　　D. exe

27. 下面的数据声明及赋值，没有错误的是（　　）。

A. float f = 1.3;　　B. char c = "a";　　C. byte b = 257;　　D. int i = 10;

28. 现在假设有如下程序：

```
class Happy {
    public static void main(String args[])    {
        int i = 1 ;
        int j = i++ ;
        if((i==(++j))&&((i++)==j))       {
            i += j ;
        }
        System.out.println("i = "+i);
    }
}
```

运行完上面代码之后输出 i 的值是（　　）。

A. 4　　B. 5　　C. 3　　D. 6

测试试卷二

1. 下面（　　）不属于面向对象的特点。

A. 封装　　B. 转型　　C. 继承　　D. 多态

2. 下面关于类与对象的描述正确的是（　　）。

A. 任何情况下必须先有类再有对象，对象只能够调用类中定义的方法，不能够调用属性

B. “class”关键字可以定义类，并且要求文件名称与类名称完全一致，否则程序将无法编译通过

C. 一个类可以产生多个对象，通过关键字 new 实例化的每个对象都将拥有属于自己的堆内存空间

D. 对象一旦开辟之后即使不再使用了，也会一直占据内存空间不释放

3. 下面（　　）权限定义不属于 Java。

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

4. 关于构造方法的描述正确的是（　　）。

A. 构造方法；在使用关键字 new 实例化对象时会自动进行调用

B. 一个类中可以没有任何构造方法的定义

C. 构造方法不会有返回值，所以需要使用 void 进行声明

D. 构造方法在进行重载时，方法名称可以不同

5. 下面关于 String 类的特点描述正确的一项是（　　）。

A. String 类在需要时可以定义子类

B. String 类的对象内容一旦声明则不可改变

C. String 类可以直接利用“==”进行字符串内容的比较

D. String 类对象实例化后都会自动存入字符串对象池

6. 下列（　　）不属于面向对象程序设计的基本要素。

A. 类　　B. 对象　　C. 方法　　D. 安全

7. 下列程序的执行结果是（　　）。

```
public class TestDemo {
    public void fun() {
        static int i = 0;
        i++;
        System.out.println(i);
    }
    public static void main(String args[]) {
        Demo d = new Demo();
        d.fun();
    }
}
```

A. 编译错误　　B. 0

C. 1　　D. 运行成功，但不输出

8. 顺序执行下列程序语句后，则 b 的值是（　　）。

```
String str = "Hello" ;
String b = str.substring(0,2) ;
```

A. Hello　　B. hello　　C. He　　D. null

9. 不能直接使用 new 创建对象的类是（　　）。

A. 静态类　　B. 抽象类　　C. 最终类　　D. 公有类

10. 为类定义多个名称相同、但参数的类型或个数不同的方法的做法称为（　　）。

A. 方法重载　　B. 方法覆写　　C. 方法继承　　D. 方法重用

11. 定义接口的关键字是（　　）。

A. extends　　B. class　　C. interface　　D. public

12. 现在有两个类 A、B，以下描述中表示 B 继承自 A 的是（　　）。

A. class A extends B　　B. class B implements A

C. class A implements　　D. class B extends A

13. 下面关于子类调用父类构造方法的描述正确的是（　　）。

A. 子类定义了自己的构造方法，就不会调用父类的构造方法。

B. 子类必须通过 super 关键字调用父类有参的构造方法。

C. 如果子类的构造方法没有通过 super 调用父类的构造方法，那么子类会先调用父类中无参构造方法，之后再调用子类自己的构造方法。

D. 创建子类对象时，先调用子类自己的构造方法，让后再调用父类的构造方法。

14. 假设类 X 是类 Y 的父类，下列声明对象 x 的语句中不正确的是（　　）。

A. X x = new X() ;　　B. X x = new Y() ;

C. Y x = new Y() ;　　D. Y x = new X() ;

15. 编译并运行下面的程序，程序的执行结果是（　　）。

```
public class A {
    public static void main(String args[]) {
        B b = new B();
        b.test();
    }
    void test() {
        System.out.print("A");
    }
}
class B extends A {
    void test() {
        super.test();
        System.out.println("B");
    }
}
```

A. 产生编译错误　　B. 代码可以编译运行，并输出结果：AB

C. 代码可以编译运行，但没有输出　　D. 编译没有错误，但会运行时会产生异常

16. 编译运行下面的程序，程序的运行结果是（　　）。

```
public class A {
    public static void main(String args[]) {
        B b = new B();
        b.test();
    }
    public void test() {
        System.out.print("A");
    }
```

```
}
class B extends A {
    void test() {
        super.test();
        System.out.println("B");
    }
}
```

A. 产生编译错误，因为类 B 覆盖类 A 的方法 test()时，降低了其访问控制的级别

B. 代码可以编译运行，并输出结果：AB

C. 代码可以编译运行，但没有输出

D. 代码可以编译运行，并输出结果：A

17. 下面（　　）修饰符所定义的方法必须被子类所覆写。

A. final　　B. abstract　　C. static　　D. interface

18. 下面（　　）修饰符所定义的方法不能被子类所覆写。

A. final　　B. abstract　　C. static　　D. interface

19. 下面的程序编译运行的结果是（　　）。

```
public class A implements B {
    public static void main(String args[]) {
        int m, n;
        A a = new A();
        m = a.K;
        n = B.K;
        System.out.println(m + ", " + n);
    }
}
interface B {
    int K = 5;
}
```

A. 5，5　　B. 0，5

C. 0，0　　D. 编译程序产生编译结果

20. 下面关于接口的说法中不正确的是（　　）。

A. 接口所有的方法都是抽象的

B. 接口所有的方法一定都是 public 类型

C. 用于定义接口的关键字是 implements

D. 接口是 Java 中的特殊类，包含全局常量和抽象方法

21. 下面关于 Java 的说法不正确的是（　　）。

A. abstract 和 final 能同时修饰一个类

B. 抽象类不光可以做父类，也可以做子类

C. 抽象方法不一定声明在抽象类中，也可以在接口中

D. 声明为 final 的方法不能在子类中覆写

22. 下面关于 this 与 super 的区别描述错误的是（　　）。
A. this 和 super 都可以调用类中的属性、方法、构造方法
B. this 表示本类实例化对象，而 super 表示父类实例化对象
C. 使用“this.属性”或者“this.方法()”时都会先从本类查找方法，如果本类没有定义，则通过父类查找
D. 子类可以利用“super.方法()”调用父类方法，这样可以避免覆写父类方法时所产生的递归调用问题

23. 使用（　　）关键字可以在程序中手工抛出异常。
A. throws　　B. throw　　C. assert　　D. class

24. 下面（　　）关键字可以用在方法的声明处。
A. throws　　B. assert　　C. class　　D. interface

25. 为了捕获一个异常，代码必须放在下面（　　）语句块中。
A. try　　B. catch　　C. throws　　D. finally

26. 下面关于 try 块的描述正确的一项是（　　）。
A. try 块后至少应有一个 catch 块　　B. try 块后必须有 finally 块
C. 可能抛出异常的方法应放在 try 块中　　D. 对抛出的异常的处理应放在 try 块中

27. 下面关于 finally 块中的代码的执行，描述正确的是（　　）。
A. 总是被执行
B. 如果 try 块后面没有 catch 块时，finally 块中的代码才会执行
C. 异常发生时才被执行
D. 异常没有发生时才执行

28. 一个异常将终止（　　）。
A. 整个程序　　B. 只终止抛出异常的方法
C. 产生异常的 try 块　　D. 上面的说法都不对

29. 所有程序可处理异常的共同父类是（　　）。
A. Error　　B. Exception　　C. Throwable　　D. RuntimeException

30. String 和 Object 类在（　　）包中定义的。
A. java.lang　　B. java.util　　C. java.net　　D. java.sql

31. 下面（　　）权限是同一包可以访问，不同包的子类可以访问，不同包的非子类不可以访问。
A. private　　B. default　　C. protected　　D. public

32. 下列说法正确的一项是（　　）。
A. java.lang.Integer 是接口　　B. String 定义在 java.util 包中
C. Double 类在 java.lang 包中　　D. Double 类在 java.lang.Object 包中

33. 下列关于包、类和源文件的描述中，不正确的一项是（　　）。
A. 一个包可以包含多个类
B. 一个源文件中，只能有一个 public class
C. 属于同一个包的类在默认情况不可以互相访问，必须使用 import 导入
D. 系统不会为源文件创建默认的包

34. 定义类时不可能用到的关键字是（　　）。

A. final　　B. public　　C. protected　　D. static

35. 下面关于泛型的描述中错误的一项是（　　）。

A. “? extends 类”表示设置泛型上限

B. “? super 类”表示设置泛型下限

C. 利用“?”通配符可以接收全部的泛型类型实例，但却不可修改泛型属性内容

D. 如果类在定义时使用了泛型，则在实例化类对象时需要设置相应的泛型类型，否则程序将无法编译通过

36. 下面关于枚举的描述正确的一项是（　　）。

A. 枚举中定义的每一个枚举项其类型都是 String

B. 在 Java 中可以直接继承 java.util.Enum 类实现枚举类的定义

C. 利用枚举类中的 values()方法可以取得全部的枚举项

D. 枚举中定义的构造方法只能够使用 private 权限声明

37. 下面（　　）Annotation 不是 Java 内建的 Annotation。

A. @Override　　B. @Deprecated

C. @SuppressWarning　　D. @FunctionalInterface

38. 关于 Java8 中提供的四个核心函数式接口的描述，正确的一项是（　　）。

A. Predicate 接口中的方法不能够返回数据，只能够接收并操作数据

B. Consumer 接口中的方法可以对数据进行判断，并且可以返回判断结果

C. Function 接口中的方法可以接收参数，并且将数据处理后返回

D. Supplier 接口中的方法可以接收基本数据类型参数，但是没有返回值

39. 关于 Java 的异常处理中，错误的是（　　）。

A. Java 中用户可以处理的异常都是 Exception 的子类

B. Java 中出现异常时，可以利用 try 进行捕获

C. Java 中产生异常代码时，如果没有异常处理，则会由系统处理异常，而后让程序正常执行完毕

D. 一个 try 语句后面可以跟多个 catch 块，也可以只跟一个 finally 语句块

40. 下面对于多态性的描述，错误的一项是（　　）。

A. 面向对象多态性描述的就是对象转型的操作

B. 对象可以自动实现向上转型

C. 对象的向下转型需要强制转型

D. 可以利用 instanceof 方法判断某一个对象是否属于某个类的实例

41. 为 Demo 类的一个无形式参数无返回值的方法 method 书写方法头，使得使用类名 Demo 作为前缀就可以调用它，该方法头的形式为（　　）。

A. static void method()　　B. public void method()

C. final void method()　　D. abstract void method()

42. 下面代码会存在什么问题？（　　）

```
public class MyClass {
    public static void main(String arguments[])  {
```

```
        amethod(arguments);
    }
    public void amethod(String[] arguments){
        System.out.println(arguments);
        System.out.println(arguments[1]);
    }
}
```

A. 错误，void amethod()不是 static 类型

B. 错误，main()方法不正确

C. 错误，数组必须导入参数

D. 方法 amethod()必须用 String 类型描述

43. 当编译下列代码可能会输出（　　）。

```
class Test {
    static int i ;
    public static void main(String args[]) {
        System.out.println(i);
}
}
```

A. Error Variable i may not have been initialized

B. null

C. 1

D. 0

44. 如果试图编译并运行下列代码可能会打印输出（　　）。

```
int i = 9 ;
switch(i) {
     default:
          System.out.println("default");
     case 0 :
          System.out.println("zero");
          break ;
     case 1 : System.out.println("one");
     case 2 : System.out.println("two");
}
```

A. default　　B. default , zero

C. error default clause not defined　　D. no output displayed

45. 在一个类文件中，导入包、类和打包的排列顺序是（　　）。

A. package、import、class　　B. class、import、package

C. import、package、class　　D. package、class、import

46. 现在有如下一段程序：

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

```
        float [][] f1 = {{1.2f,2.3f},{4.5f,5.6f}} ;
        Object oo = f1 ;
        f1[1] = oo ;
        System.out.println("Best Wishes "+f1[1]);
    }
}
```

该程序会出现的效果是（　　）。

A. {4.5,5.6}　　B. 4.5

C. compilation error in line NO.5　　D. exception

47. 现在有如下一段程序：

```
class super {
    String name ;
    public super(String name) {
        this.name = name ;
    }
    public void fun1()  {
        System.out.println("this is class super !"+name);
    }
}
class sub extends super {
    public void fun1()  {
        System.out.println("this is class sub !"+name);
    }
}
class Test {
    public static void main(String args[]) {
        super s = new sub();
    }
}
```

运行上面的程序可能会出现的结果是（　　）。

A. this is class super !　　B. this is class sub !

C. 编译时出错　　D. 运行时出错

48. 当试图编译和运行下面代码可能会发生（　　）。

```
class Base {
    private void amethod(int iBase) {
        System.out.println("Base.amethod");
    }
}
class Over extends Base {
    public static void main(String args[]) {
        Over o = new Over();
        int iBase = 0 ;
        o.amethod(iBase) ;
```

```
    }
    public void amethod(int iOver) {
        System.out.println("Over.amethod");
    }
}
```

A. Compile time error complaining that Base.amethod is private

B. Runntime error complaining that Base.amethod is private

C. Output of Base.amethod

D. Output of Over.amethod

49. 如要在字符串 s（内容为“welcome to mldn !! ”），中，发现字符't'的位置，应该使用下面（　　）方法。

A. mid(2,s)　　B. charAt(2)　　C. s.indexOf('t')　　D. indexOf(s,'v')

50. 现在有如下一段代码：

```
public class Test {
    public int aMethod() {
        static int i=0;
        i++;
        return i;
    }
    public static void main(String args[]) {
        Test test = new Test();
        test.aMethod();
        int j = test.aMethod();
        System.out.println(j);
    }
}
```

将产生的结果是（　　）。

A. Compilation will fail

B. Compilation will succeed and the program will print “0”.

C. Compilation will succeed and the program will print “1”.

D. Compilation will succeed and the program will print “2”.

测试试卷三

1. 线程的启动方法是（　　）。

A. run()　　B. start()　　C. begin()　　D. accept()

2. Thread 类提供表示线程优先级的静态常量，代表普通优先级的静态常量是（　　）。

A. MAX_PRIORITY　　B. MIN_PRIORITY

C. NORMAL_PRIORITY　　D. NORM_PRIORITY

3. 设置线程优先级的方法是（　　）。

A. setPriority()　　B. getPriority()　　C. getName()　　D. setName()

4. 下面（　　）方法是 Thread 类中不建议使用的。
   A. stop()　　B. suspend()　　C. resume()　　D. 全部都是
5. 下列（　　）关键字通常用来为对象加锁，从而使得对对象的访问是排他的。
   A. serialize　　B. transient　　C. synchronized　　D. static
6. 如果要实现多线程编程下面错误的是（　　）。
   A. 多线程处理类可以继承 Thread 类，同时覆写 run()方法
   B. 多线程处理类可以实现 Runnable 接口，同时覆写 run()方法
   C. 多线程处理类可以实现 java.util.concurrent.Callable 接口，同时覆写 apply()方法
   D. 多线程处理类可以继承 Synchronized 类，同时覆写 run()方法
7. 下面（　　）方法不是 Object 类所提供的线程操作方法。
   A. public final void wait() throws InterruptedException
   B. public final void notify()
   C. public final void notifyAll()
   D. public String toString()
8. 下面（　　）父类或父接口是无法实现多线程子类定义的。
   A. Serializable　　B. Thread　　C. Runnable　　D. Callable
9. 使用 Runtime 类的（　　）方法，可以释放垃圾内存。
   A. exec()　　B. run()　　C. invoke()　　D. gc()
10. Object 类中的（　　）方法不能被覆写。
    A. toString()　　B. getClass()　　C. clone()　　D. finalize()
11. 如果要为对象回收做收尾操作，则应该覆写 Object 类中（　　）方法。
    A. toString()　　B. getClass()　　C. clone()　　D. finalize()
12. 下面关于数组排序的说明错误的是（　　）。
    A. java.util.Arrays 类提供有数组排序的支持方法：sort()
    B. 通过 java.util.Arrays 类排序的对象所在类需要实现 Comparable 或 Comparator 接口
    C. String 数组可以进行排序，是因为 String 类实现了 Comparable 接口
    D. Comparator 接口中提供有 compare()方法实现数组的排序操作
13. 当执行“Math.round(-15.01)”程序后的计算结果是（　　）。
    A. -15　　B. -14　　C. -16　　D. 15
14. 下面关于 Date 类的描述错误的是（　　）。
    A. java.util.Date 类下有三个子类：java.sql.Date、java.sql.Timestamp、java.sql.Time
    B. 利用 SimpleDateFormat 类可以对 java.util.Date 类进行格式化显示
    C. 直接输出 Date 类对象就可以取得日期时间数据，但是取得的月数是从 0 开始计算的
    D. java.util.Date 类可以直接将 long 变量的数字转换为本类对象
15. 判断某一个字符串是否是小数或者是整数，以下列出的正则表达式正确的是（　　）。
    A. \d+　　B. \d+(\.\d+)?　　C. \d+\.\d+　　D. \d{1,}
16. 下面关于 Class 类对象的实例化对象取得，错误的一项是（　　）。
    A. 利用 Object 类中的 getClass()方法取得 Class 类的实例化对象

B. 利用 Class 类的构造方法取得 Class 类的实例化对象

C. 利用“类.class”格式取得 Class 类的实例化对象

D. 通过 Class.forName()方法根据类名称取得 Class 类的实例化对象

17. 下面列出的 Class 类的方法中，(　　) 可以取得指定类型中全部方法的定义 (　　)。

A. public Method [] getMethods()　　B. public Field [] getFields()

C. public Field [] getDeclaredFields()　　D. public Constructor [] getConstrutors()

18. 下面关于国际化程序实现的过程，描述错误的是 (　　)。

A. 国际化程序可以利用 Local 类来设置要显示文字的城市及语言编码

B. 国际化程序主要依靠*.properties 文件实现文字资源的定义

C. 国际化程序中必须依靠 ResourceBundle 才能够进行资源文件的读取

D. 国际化程序中可以使用 MessageFormat 类进行数据的转换，该类是 SimpleDateFormat 的子类

19. 下面 (　　) String 类方法不能够使用正则表达式。

A. substring()　　B. replaceFirst()　　C. split()　　D. matches()

20. 下面 (　　) 类不属于 CharSequence 接口的子类。

A. String　　B. StringBuffer　　C. StringBuilder　　D. StringUtils

21. 下面 (　　) 类不属于 Accessible 的子类。

A. Field　　B. Constructor　　C. Method　　D. Annotation

22. File 类提供了许多管理磁盘的方法。其中，建立目录的方法是 (　　)。

A. delete()　　B. mkdirs()　　C. makedir()　　D. exists()

23. 提供 println()方法和 print()方法的类是 (　　)。

A. PrintStream　　B. System

C. InputStream　　D. DataOutputStream

24. 不同的操作系统使用不同的路径分隔符。静态常量 separator 表示路径分隔符，它属于的类是 (　　)。

A. FileInputStream　　B. FileOutputStream

C. File　　D. InputStream

25. 下面的说法不正确的是 (　　)。

A. InputStream 与 OutputStream 类通常是用来处理字节流，也就是二进制文件

B. Reader 与 Writer 类则是用来处理字符流，也就是纯文本文件

C. Java 中 IO 流的处理通常分为输入和输出两个部分

D. File 类是输入/输出流类的子类

26. 下面的说法正确的是 (　　)。

A. InputStream 与 OutputStream 都是抽象类

B. Reader 与 Writer 不是抽象类

C. RandomAccessFile 是抽象类

D. File 类是抽象类

27. 与 InputStream 相对应的 Java 系统的标准输入对象是（　　）。

A. System.in　　B. System.out　　C. System.err　　D. System.exit()

28. FileOutputStream 类的父类是（　　）。

A. File　　B. FileOutput　　C. OutputStream　　D. InputStream

29. InputStreamReader 类提供的功能是（　　）。

A. 数据校验　　B. 文本行计数　　C. 压缩　　D. 将字节流变为字符流

30. Socket 的工作流程是（　　）。

①打开连接到 Socket 的输入/输出

②按照某个协议对 Socket 进行的读/写操作

③创建 Socket

④关闭 Socket

A. ①③②④　　B. ②①③④　　C. ③①②④　　D. ①②③④

31. 下面（　　）类不是 Collection 的子类。

A. ArrayList　　B. Vector　　C. HashMap　　D. TreeSet

32. HashSet 子类依靠（　　）方法区分重复元素。

A. toString()、equals()　　B. clone()、equals()

C. hashCode()、equals()　　D. getClass()、clone()

33. 下列（　　）不是 getConnection()方法的参数。

A. 数据库用户名　　B. 数据库的访问密码

C. JDBC 驱动器的版本　　D. 连接数据库的 URL

34. Statement 接口中的 executeQuery(String sql)方法返回的数据类型是（　　）。

A. Statement 接口实例　　B. Connection 接口实例

C. DatabaseMetaData 类的对象　　D. ResultSet 接口对象

35. 下列不属于更新数据库操作的步骤的一项是（　　）。

A. 加载 JDBC 驱动程序　　B. 定义连接的 URL

C. 执行查询操作　　D. 执行更新操作