**Java面向对象程序设计 题库1**

**一、**单项选择题（从下列各题四个备选答案中选出一个正确答案，并将其代号写在答题纸相应位置处。答案错选或未选者，该题不得分。每小题1分，共15分。）

1.\_\_\_\_\_\_\_\_包含用于开发Java程序的预定义类和接口。

a. Java language specification b. Java API

c. Java JDK d. Java IDE

Key：b

2.程序中main方法头部写为（）

a. public static void main(string[] args)

b. public static void Main(String[] args)

c. public static void main(String[] args)

d. public static main(String[] args)

e. public void main(String[] args)

Key:c

3.要运行ByteCode.class应用程序，哪一个JDK命令是正确的？

a. java ByteCode

b. java ByteCode.class

c. javac ByteCode.java

d. javac ByteCode

e. JAVAC ByteCode

Key:a

4.

5.假设一个Scanner对象创建了，如下：

Scanner input = new Scanner(System.in);

下面哪一个方法可以读取一个实数（）

a. input.nextDouble();

b. input.nextdouble();

c. input.double();

d. input.Double();

Key:a

6.要定义一个值为99.99的常量MAX\_LENGTH，应该写成（）

a. final MAX\_LENGTH = 99.98;

b. final float MAX\_LENGTH = 99.98;

c. double MAX\_LENGTH = 99.98;

d. final double MAX\_LENGTH = 99.98;

Key:d

7.下面哪一个表达式的结果是1（）

a. 2 % 1

b. 15 % 4

c. 25 % 5

d. 37 % 6

Key:d

8. 24 % 5 is \_\_\_\_\_

a. 1

b. 2

c. 3

d. 4

e. 0

Key:d

9.在Java中如何表达2.5 ^ 3.1（）

a. 2.5 \* 3.1

b. Math.pow(2.5, 3.1)

c. Math.pow(3.1, 2.5)

d. 2.5 \*\* 3.1

e. 3.1 \*\* 2.5

Key:b

10.Math.pow(4, 1.0 / 2) returns \_\_\_\_\_\_\_\_\_\_.

a. 2

b. 2.0

c. 0

d. 1.0

e. 1

Key:b

11.表达式4 + 20 / (3 - 1) \* 2的值是（）

a. 4

b. 20

c. 24

d. 9

e. 25

Key:c

12.要将一个double变量d赋值给一个float变量x，可以写成（）

a. x = (long)d

b. x = (int)d;

c. x = d;

d. x = (float)d;

Key:d

13.语句System.out.println((int)Math.random() \* 4);的输出是（）

a. 0

b. 1

c. 2

d. 3

e. 4

Key:a

14.假设x=4，下面哪一个返回值是true（）

a. !(x == 4)

b. x != 4

c. x == 5

d. x != 5

Key:d

15.假设x和y都是10，表达式(y > 10) && (x-- > 10)的运算后，x的值是（）

a. 9

b. 10

c. 11

Key:b

16.假设x和y都是10，表达式(y > =10) || (x++ > 10)的运算后，x的值是（）

a. 9

b. 10

c. 11

Key:b

17.分析程序下面片段

int x;

double d = 1.5;

switch (d) {

case 1.0: x = 1;

case 1.5: x = 2;

case 2.0: x = 3;

}

1. 编译错误，因为缺少break语句
2. 编译错误，因为缺少default语句
3. switch控制变量不能为double类型
4. 没有编译错误

Key：c

18.下面代码的运算结果是（）

boolean even = false;

System.out.println((even ? "true" : "false"));

a. true

b. false

c. nothing

d. true false

Key:b

19.Math.round(3.6)返回?

a. 3.0

b. 3

c. 4

d. 4.0

Key:c

20.假设x是一个值为'b'的字符变量，语句System.out.println(++x)输出（）

a. a

b. b

c. c

d. d

Key:c

21.下面哪一个赋值语句是正确的?

a. char c = 'd';

b. char c = 100;

c. char c = "d";

d. char c = "100";

Key:ab

22.假设s是一个存储值为”java”的字符串变量，运行char x = s.charAt(4);后x的值是（）

a. 'a'

b. 'v'

C、运行时产生StringIndexOutofBoundsException型异常

Key：C

23.字符A的Unicode编码为65，请问表达式’A’ + 1的值是（）

a. 66

b. B

c. A1

d. 非法表达式

key:a

24."AbA".compareToIgnoreCase("abC") 返回\_\_\_\_\_\_\_\_\_\_\_.

a. 1

b. 2

c. -1

d. -2

e. 0

Key:d

25.下面的循环输出（）

for (int i = 1; i <= 10; i++) {

System.out.print(i + " ");

i++;

}

a. 1 2 3 4 5 6 7 8 9

b. 1 2 3 4 5 6 7 8 9 10

c. 1 2 3 4 5

d. 1 3 5 7 9

e. 2 4 6 8 10

Key:d

26.给定一个方法

static void nPrint(String message, int n) {

while (n > 0) {

System.out.print(message);

n--;

}

}

调用nPrint('a', 4)的输出结果是?

a. aaaaa

b. aaaa

c. aaa

d. invalid call

Key:d

27.(int)(Math.random() \* (65535 + 1))返回值范围\_\_\_\_\_\_\_\_\_\_.

a. [1, 65536]

b. [1, 65536)

c. [0, 65536)

d. [0, 65536]

key:c

28.当你将一个数组作为参数传给方法时，这个方法接收的是 \_\_\_\_\_\_\_\_\_\_.

a. 数组的拷贝

b. 数组第一个元素的拷贝

c. 数组的引用

d. 数组的长度

Key:c

29.下面那条语句是正确的?

a. char[][] charArray = {'a', 'b'};

b. char[2][2] charArray = {{'a', 'b'}, {'c', 'd'}};

c. char[2][] charArray = {{'a', 'b'}, {'c', 'd'}};

d. char[][] charArray = {{'a', 'b'}, {'c', 'd'}};

Key:d

30.\_\_\_\_\_\_\_\_\_\_ 表示一个现实世界中可以标识的实体

a. 一个类

b. 一个对象

c. 一个方法

d. 一个数据域

Key:b

31.如下定义Circle x = new Circle(), 下面哪一句最正确.

a.x是一个int值

b. x 是一个Circle类型的对象

c. x 是一个Circle类型对象的引用

d. 你可以将一个int值赋给x

Key:c

32.一个只和单个的实例对象相关联的方法称为\_\_\_\_\_\_\_\_\_.

a. 静态方法a static method

b. 类方法a class method

c. 实例方法an instance method

d. 以上都不对

Key:c

33.假设xMethod()在类MyClass的构造器中被调用, 请问xMethod() is 类的\_\_\_\_\_\_\_\_\_

public MyClass() {

xMethod();

}

a. 静态方法

b. 实例方法

c. 静态方法或实例方法

Key:c

34.假设Circle[] x = new Circle[10], 下面哪一个是最准确的?

a. x是10个int类型的数组

b. x是10个Circle类型的数组

c. x是一个数组的引用，并且该数组的每个元素都是一个Circle对象的引用

d. x是一个数组的引用，并且该数组的每个元素都是一个Circle对象

Key:c

35.下面哪一句是创建字符串"Welcome to Java"的引用的最佳方法?

a. String s = "Welcome to Java";

b. String s = new String("Welcome to Java");

c. String s; s = "Welcome to Java";

d. String s; s = new String("Welcome to Java");

Key:a

36.分析程序

class Test {

public static void main(String[] args) {

String s;

System.out.println("s is " + s);

}

}

a. 编译错误，因为s没有初始化，但在println语句使用了

b. 运行错误，因为s没有初始化，但在println语句使用了

c. 运行错误，因为s是null，但在println语句使用了

d. 编译运行良好

Key:a

37.假设Cylinder是Circle的子类.分析代码

Cylinder cy = new Cylinder(1, 1);

Circle c = cy;

a. 编译错误

b. 运行错误

c. 编译运行良好

Key:c

38.创建一个ArrayList对象，使用 \_\_\_\_\_\_\_\_\_.

a. new ArrayList[]

b. new ArrayList[100]

c. new ArrayList<>()

d. ArrayList()

Key:c

39.封装意味着\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

a. 数据域应该定义为private

b. 一个类可以继承另一个类

c. 父类的变量可以引用子类对象

d. 一个类可以包含另一个类的对象

Key:a

40.组合意味着\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

a. 数据域应该定义为private

b. T一个类可以继承另一个类

c. 父类的变量可以引用子类对象

d. 一个类可以包含另一个类的对象

Key:d

41.下面哪一个是合法的抽象方法的定义?

a. public abstract method();

b. public abstract void method();

c. public void abstract method();

d. public void method() {}

e. public abstract void method() {}

Key:b

42.假设Calendar calendar = new GregorianCalendar(). \_\_\_\_\_\_\_\_\_\_ 返回星期几

a. calendar.get(Calendar.MONTH)

b. calendar.get(Calendar.MONTH\_OF\_YEAR)

c. calendar.get(Calendar.WEEK\_OF\_MONTH)

d. calendar.get(Calendar.WEEK\_OF\_YEAR)

Key:d

43.\_\_\_\_\_\_\_ 不是引用类型.

a. 类

b. 接口

c. 数组

d. 原生数据类型

Key:d

44.接口和实现它的类之间的关系是

a. None

b. 聚合

c. 继承

d. 组合

Key:c

1. 多项选择题（从下列各题四个备选答案中选出一个或多个正确答案，并将其代号写在答题纸相应位置处。答案错选、少选、多选或未选者，该题不得分。每小题2分，共20分。）

1.分析下面代码

Code 1:

int number = 45;

boolean even;

if (number % 2 == 0)

even = true;

else

even = false;

Code 2:

int number = 45;

boolean even = (number % 2 == 0);

a. Code 1有编译错误

b. Code 2 有编译错误

c. Code 1和Code 2都有编译错误

d. Code 1和Code 2都没有编译错误，但Code 2更好

Key:d

2.分析下面的代码（）

public class Test {

public static void main(String[] args) {

System.out.println(m(2));

}

public static int m(int num) {

return num;

}

public static void m(int num) {

System.out.println(num);

}

}

a. 编译错误，因为两个方法有相同的签名

b. 编译错误，因为第二个m方法定义了但没有在main方法中调用

c. 可以运行并打印一次2

d. 可以运行并打印两次2.

Key:a

3.方法调用者不需要知道该方法是如何实现的，实现的详情封装在方法中，并对调用者隐藏，这特性称为\_\_\_\_\_\_\_\_\_\_.

a. 信息隐藏

b. 封装

c. 方法隐藏

d. 方法简化

key:ab

4.分析下面的代码

public class Test {

public static void main(String[] args) {

int[] x = new int[3];

System.out.println("x[0] is " + x[0]);

}

}

a. 编译错误，因为数组的大小没有指定.

b. 运行时错误，因为数组元素没有初始化

c. 运行良好，输出 x[0] is 0.

d. 运行时错误，因为数组元素x[0]没有定义

Key:c

5.下面哪一句是正确的

a. String[] list = new String{"red", "yellow", "green"};

b. String[] list = new String[]{"red", "yellow", "green"};

c. String[] list = {"red", "yellow", "green"};

d. String list = {"red", "yellow", "green"};

e. String list = new String{"red", "yellow", "green"};

Key:bc

6.分析代码

public class Test {

public static void main(String[] args) {

int[] a = new int[4];

a[1] = 1;

a = new int[2];

System.out.println("a[1] is " + a[1]);

}

}

a. 编译错误因为将new int[2]赋给了a.

b. 运行错误因为a[1]没有初始化

c. 运行结果：a[1] is 0.

d. 运行结果：a[1] is 1.

Key:c

7.假设方法p的头部如下：

public static int[][] p()下面哪一句可以作为方法p的返回值?

a. return 1;

b. return {1, 2, 3};

c. return int[]{1, 2, 3};

d. return new int[]{1, 2, 3};

e. return new int[][]{{1, 2, 3}, {2, 4, 5}};

Key:e

8.下面哪个说法是正确的?

a. 在一个类中如果没有显式地定义一个构造方法，JVM会自动添加一个默认构造方法

b. 类中必须显式定义至少一个构造器

c. 每个类都有一个默认的构造器

d. 默认构造器是一个无参的构造器

E. 如果类中已经显式定义了一个构造器，JVM仍然会添加一个默认构造器。

Key:ad

9.分析代码

public class Test {

public static void main(String[] args) {

double radius;

final double PI= 3.15169;

double area = radius \* radius \* PI;

System.out.println("Area is " + area);

}

}

a. 编译错误，因为变量radius没有初始化

b. 编译错误，因为常量PI定义在方法里面

c. 没有编译错误，但运行时错误，因为radius没有初始化

d. 编译和运行都没有错误

Key:a

10.分析代码

public class Test {

public static void main(String[] args) {

int n = 2;

xMethod(n);

System.out.println("n is " + n);

}

void xMethod(int n) {

n++;

}

}

a. 编译错误，因为xMethod没有返回值.

b. 编译错误，因为xMethod没有定义为static

c. 输出n is 1.

d. 输出 n is 2.

e. 输出 n is 3.

Key:b

11.下面描述正确的是?

a. 使用private修饰符封装数据域

b. 封装数据域使得程序更容易维护

c. 封装数据域使得程序更短

d. 封装数据域可以阻止程序错误

Key:abd

12.运行程序

public class Test {

public static void main(String[] args) {

String s1 = "Welcome to Java!";

String s2 = "Welcome to Java!";

if (s1 == s2)

System.out.println("s1和s2 引用同一个字符串对象");

else

System.out.println("s1和s2 引用不同的字符串对象");

}

}

a. s1和s2 引用同一个字符串对象

b. s1和s2 引用不同的字符串对象

Key:a

13.运行程序

public class Test {

public static void main(String[] args) {

String s1 = new String("Welcome to Java!");

String s2 = s1.toUpperCase();

if (s1 == s2)

System.out.println("s1和s2 引用同一个字符串对象");

else if (s1.equals(s2))

System.out.println("s1和s2 有同样的内容");

else

System.out.println("s1和s2 有不同的内容");

}

}

a. s1和s2 引用同一个字符串对象

b. s1和s2 有同样的内容

c. s1和s2 有不同的内容

Key:c

14.假设s 是 "ABCABC",\_\_\_\_\_\_\_\_\_\_ 返回一个字符数组.

a. toChars(s)

b. s.toCharArray()

c. String.toChars()

d. String.toCharArray()

e. s.toChars()

Key:b

15.分析代码

public class A extends B {}

class B {

public B(String s) {

}

}

a. 编译错误，因为A没有默认构造器

b. 编译错误，意难忘A的默认构造器会调用B的默认构造器，但是B中没有默认构造器

c. 如果在A中增加构造器A(String s) { }将编译良好

d. 如果在A中增加构造器A(String s) {super(s); }将编译良好

Key:bd

16.找出下面代码的编译错误

public class Test {

public static void main(String[] args) {

m(new GraduateStudent());

m(new Student());

m(new Person());

m(new Object());

}

public static void m(Student x) {

System.out.println(x.toString());

}

}

class GraduateStudent extends Student {

}

class Student extends Person {

@Override

public String toString() {

return "Student";

}

}

class Person extends Object {

@Override

public String toString() {

return "Person";

}

}

a. m(new GraduateStudent())导致一个错误

b. m(new Student())导致一个错误

c. m(new Person()) 导致一个错误

d. m(new Object()) 导致一个错误

Key:cd

17.分析代码

public class Test {

public static void main(String[] args) {

String s = new String("Welcome to Java");

Object o = s;

String d = (String)o;

}

}

a. 当将s赋值给o时(Object o = s),一个新的对象创建了

b. 当将o转换为字符串d(String d = (String)o)时, 一个新的对象创建了

c. 当将o转换为字符串d(String d = (String)o)时,o的内容发生了改变

d. S、 o和d 引用的是同一个字符串对象

Key:d

18.分析代码

ArrayList<String> list = new ArrayList<String>();

list.add("Beijing");

list.add("Nanchang");

list.add("Shanghai");

list.set(3, "Hong Kong");

a. 最后一行将导致运行时错误，因为list索引3的位置没有元素

b. 最后一行将导致编译错误，因为list索引3的位置没有元素

c. 最后一行替换为list.add(3, "Hong Kong"), 代码将编译运行良好

d. 最后一行替换为list.add(4, "Hong Kong"), 代码将编译运行良好

Key:ac

19.假设ArrayList x 包括两个字符串[Beijing, Singapore]. 下面哪一个将导致x变成[Beijing]?

a. x.remove("Singapore")

b. x.remove(0)

c. x.remove(1)

d. x.remove(2)

Key:ac

20.分析代码

1. import java.util.\*;

2. public class Test {

3. public static void main(String[] args) {

4. Calendar[] calendars = new Calendar[10];

5. calendars[0] = new Calendar();

6. calendars[1] = new GregorianCalendar();

7. }

8. }

a. 编译错误，因为第4行中的Calendar是一个抽象类

b.编译错误，因为第5行的Calendar是一个抽象类

c. 编译错误，因为第6行Calendar[1] 不是GregorianCalendar类型

d. 没有编译错误

Key:b

21.下面哪一个是对的?

a. 一个类应该描述一个简单的实体，并且该类中所有的操作要合乎逻辑

b. 一个类必须包括无参构造器

c. 构造器总是public

d. 构造器可以是protected

Key:ad

22.下面哪一行不是Java的注释（）

a. /\*\* comments \*/

b. // comments

c. -- comments

d. /\* comments \*/

e. \*\* comments \*\*

Key:ce

三、程序阅读题（从下列各题四个备选答案中选出一个正确答案，并将其代号写在答题纸相应位置处。答案错选或未选者，该题不得分。每小题3分，共30分。）

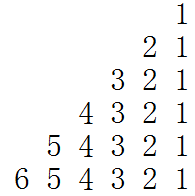
1.for (int i = 1; i <= 6; i++) {

for (int j = 6; j >= 1; j--)

System.out.print(j <= i ? j + " " : " " + " ");

System.out.println();

}

Key：

2.int balance = 10;

while (balance >= 1) {

if (balance < 9)

break;

balance = balance - 9;

}

A. -1

B. 0

C. 1

D. 2

Key:c

3.下面代码的运行结果是

double[] myList = {1, 5, 5, 5, 5, 1};

double max = myList[0];

int indexOfMax = 0;

for (int i = 1; i < myList.length; i++) {

if (myList[i] > max) {

max = myList[i];

indexOfMax = i;

}

}

System.out.println(indexOfMax);

a. 0

b. 1

c. 2

d. 3

e. 4

Key:b

4.运行结果

public class Test {

public static void main(String[] args) {

int list[] = {1, 2, 3, 4, 5, 6};

for (int i = 1; i < list.length; i++)

list[i] = list[i - 1];

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

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

}

}

a. 1 2 3 4 5 6

b. 2 3 4 5 6 6

c. 2 3 4 5 6 1

d. 1 1 1 1 1 1

Key:d

5.运行结果

public class Test {

public static void main(String[] args) {

int[] oldList = {1, 2, 3, 4, 5};

reverse(oldList);

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

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

}

public static void reverse(int[] list) {

int[] newList = new int[list.length];

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

newList[i] = list[list.length - 1 - i];

list = newList;

}

}

a. 1 2 3 4 5.

b. 1 2 3 4 5 ，然后出现ArrayIndexOutOfBoundsException异常.

c. 5 4 3 2 1.

d. 5 4 3 2 1 ，然后出现ArrayIndexOutOfBoundsException异常.

Key:a

6.public class Test {

public static void main(String[] args) {

int[][] values = {{3, 4, 5, 1 }, {33, 6, 1, 2}};

for (int row = 0; row < values.length; row++) {

java.util.Arrays.sort(values[row]);

for (int column = 0; column < values[row].length; column++)

System.out.print(values[row][column] + " ");

System.out.println();

}

}

}

a. 输出两行：3 4 5 1 和33 6 1 2

b. 输出一行：3 4 5 1 33 6 1 2

c. 输出两行：3 4 5 1 和 2 1 6 33

d. 输出两行：1 3 4 5 和 1 2 6 33

e. 输出一行： 1 3 4 5 1 2 6 33

Key:d

7.运行结果

public class Test {

public static void main(String[] args) {

int[][] values = {{3, 4, 5, 1}, {33, 6, 1, 2}};

for (int row = 0; row < values.length; row++) {

System.out.print(m(values[row]) + " ");

}

}

public static int m(int[] list) {

int v = list[0];

for (int i = 1; i < list.length; i++)

if (v < list[i])

v = list[i];

return v;

}

}

a. 3 33

b. 1 1

c. 5 6

d. 5 33

e. 33 5

Key:d

8.运行程序

public class Foo {

static int i = 0;

static int j = 0;

public static void main(String[] args) {

int i = 2;

int k = 3;

{

int j = 3;

System.out.println("i + j is " + i + j);

}

k = i + j;

System.out.println("k is " + k);

System.out.println("j is " + j);

}

}

输出三行，分别是i + j is 23、k is 2、j is 0

9.运行程序

String[] tokens = "A,B;C;D".split("[,;]");

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

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

a. A,B;C;D

b. A B C D

c. A B C;D

d. A B;C;D

Key:b

9.运行程序

public class Test {

public static void main(String[] args) {

String s = "Java";

StringBuilder buffer = new StringBuilder(s);

change(buffer);

System.out.println(buffer);

}

private static void change(StringBuilder buffer) {

buffer.append(" and HTML");

}

}

a. Java

b. Java and HTML

c. and HTML

d. 什么都不显示

Key:b

10.分析代码

public class Test {

public static void main(String[] args) {

new B();

}

}

class A {

int i = 7;

public A() {

setI(20);

System.out.println("i from A is " + i);

}

public void setI(int i) {

this.i = 2 \* i;

}

}

class B extends A {

public B() {

// System.out.println("i from B is " + i);

}

@Override

public void setI(int i) {

this.i = 3 \* i;

}

}

a. A的构造器没有被调用

b. A的构造器被调用了并且显示"i from A is 7".

c. A的构造器被调用了并且显示"i from A is 40".

d. A的构造器被调用了并且显示"i from A is 60".

Key:d

11.运行程序

public class Test {

public static void main(String[] args) {

new Person().printPerson();

new Student().printPerson();

}

}

class Student extends Person {

private String getInfo() {

return "Student";

}

}

class Person {

private String getInfo() {

return "Person";

}

public void printPerson() {

System.out.println(getInfo());

}

}

a. Person Person

b. Person Student

c. Stduent Student

d. Student Person

Key:a

12.运行程序

interface A {}

class C { }

class B extends D implements A {}

public class Test {

public static void main(String[] args) {

B b = new B();

if (b instanceof A)

System.out.println("b is an instance of A");

if (b instanceof C)

System.out.println("b is an instance of C");

}

}

class D extends C { }

a. Nothing.

b. b is an instance of A.

c. b is an instance of C.

d. 输出两行：b is an instance of A 、b is an instance of C.

Key:d