**《JAVA程序设计》期末考试试题 （六）**

**一、填空题**

**1．定义类的保留字是( class )，定义接口的保留字是( interface )。**

2．Socket通常也称为 ( **套接字** )，用于描述( **IP地址** ) 和 ( **端口** )。

3．线程的优先级在**( 1**  )至( **10** )之间，数值越大( **任务越紧急** )。

4．构造方法是一种特殊的成员方法，构造方法名与( **类名** ) 相同。

5．Java语言只允许单继承，指每个类只能有一个 ( **父类** )。

6．Java源程序的扩展名是( **.java** )，经过编译后的程序的扩展名是（ **.class** ）。

7. 在一个时间只能由一个线程访问的资源称为 ( **临界资源** ) 。访问临界资源的代码( **临界代码** )。

8. 在多线程系统中，多个线程之间有 ( **同步** ) 和 ( **互斥** ) 两种关系。

**二、选择题**

1．关于选择结构下列哪个说法正确？　（　B　）

A．if语句和 else语句必须成对出现

**B．**if语句可以没有else语句对应

C．switch结构中每个case语句中必须用break语句

D．switch结构中必须有default语句

2．while循环和 do…while循环的区别是：　　（D　　）

A．没有区别，这两个结构任何情况下效果一样

B．while循环比 do…while循环执行效率高

C．while循环是先循环后判断，所以循环体至少被执行一次

**D．**do…while循环是先循环后判断，所以循环体至少被执行一次

3．关于 for循环和 while循环的说法哪个正确？　（　　）

A．while循环先判断后执行，for循环先执行后判断。B

**B．**while循环判断条件一般是程序结果，for循环的判断条件一般是非程序结果

C．两种循环任何时候都不可以替换

D．两种循环结构中都必须有循环体，循环体不能为空

4．下列修饰符中与访问控制无关的是　（D　　）

A．private　　　 B．public

C．protected  **D．**final

5． void的含义：　　（　A　）

**A．**方法没有返回值　　　 B． 方法体为空

C．没有意义　　　　 D.　定义方法时必须使用

6． return语句：　　（　C　）

A．只能让方法返回数值　　　　 B．方法都必须含有

**C．**方法中可以有多句return 　　 D．不能用来返回对象

7．关于对象成员占用内存的说法哪个正确？　　（B　　）

A．同一个类的对象共用同一段内存

**B、**同一个类的对象使用不同的内存段，但静态成员共享相同的内存空间

C．对象的方法不占用内存

D．以上都不对

8．下列说法哪个正确？( C )

A．不需要定义类，就能创建对象

B．对象中必须有属性和方法

**C．**属性可以是简单变量，也可以是一个对象

D、属性必须是简单变量

9．下列说法哪个正确？　　（　A　）

**A、**一个程序可以包含多个源文件

B、一个源文件中只能有一个类

C、一个源文件中可以有多个公共类

D、一个源文件只能供一个程序使用

10．关于方法main（）的说法哪个正确？（　C　）

A．方法main（）只能放在公共类中

B main()的头定义可以根据情况任意更改

**C．**一个类中可以没有main()方法

D．所有对象的创建都必须放在main()方法中

11．构造函数何时被调用？　　（　A　）

**A、**创建对象时　　　　　 B、类定义时

C、使用对象的方法时　　 D、使用对象的属性时

12． 抽象方法：　　（　C　）

A、可以有方法体

B、可以出现在非抽象类中

**C、**是没有方法体的方法

D、抽象类中的方法都是抽象方法

13．关于继承的说法正确的是：　　（B　　）

A、子类将继承父类所有的属性和方法。

**B、**子类将继承父类的非私有属性和方法。

C、子类只继承父类public方法和属性

D、子类只继承父类的方法，而不继承属性

14．关于构造函数的说法哪个正确？　　（　C　）

A、一个类只能有一个构造函数

B、一个类可以有多个不同名的构造函数

**C、**构造函数与类同名

D、构造函数必须自己定义，不能使用父类的构造函数

15． this和super：( C )

A、都可以用在main()方法中

B、都是指一个内存地址

**C、**不能用在main()方法中

D、意义相同

16．关于super的说法正确的是：( C )

A、是指当前对象的内存地址

B、是指当前对象的父类对象的内存地址

C、是指当前对象的父类

D、可以用在main()方法中

17．覆盖与重载的关系是　　（　A　）

**A、**覆盖只有发生在父类与子类之间，而重载可以发生在同一个类中

B．覆盖方法可以不同名，而重载方法必须同名

C．final修饰的方法可以被覆盖，但不能被重载

D．覆盖与重载是同一回事

18．关于接口哪个正确？　　（　A　）

**A、**实现一个接口必须实现接口的所有方法

B．一个类只能实现一个接口

C．接口间不能有继承关系

D．接口和抽象类是同一回事

19．异常包含下列哪些内容？　　（　A　）

**A．**程序执行过程中遇到的事先没有预料到的情况

B．程序中的语法错误

C．程序的编译错误

D．以上都是

20．   对于已经被定义过可能抛出异常的语句，在编程时：　　（　A　）

**A、**必须使用try／catch语句处理异常，或用throws将其抛出

B．如果程序错误，必须使用 try／catch语句处理异常

C．可以置之不理

D．只能使用try／catch语句处理

21．  字符流与字节流的区别在于（D　　）

A．前者带有缓冲，后者没有

B．前者是块读写，后者是字节读写

C. 二者没有区别，可以互换使用

**D.** 每次读写的字节数不同

22．下列流中哪个不属于字节流　　（D　　）

A．FileInputStream

B．BufferedInputStream

C. FilterInputStream

**D.** InputStreamReader

**三、程序填空题**

**1.**public class Sum{

public static void main(String [] args){

int j=10;

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

calculate(j);

System.out.println("At last, j is : "+j);

}

static void calculate (int j){

for (int i = 0;i<10;i++)

j++;

System.out.println("j in calculate() is: "+j);

}

}

输出结果为：

j is :  (1)

j in calculate() is : (2)

At last j is :  (3)

**答案：(1) 10； (2) 20； (3) 10。**

**2.** 按要求填空

abstract class SuperAbstract{

void a(){…}

abstract void b();

abstract int c(int i);

}

interface AsSuper

{

void x();

}

abstract class SubAbstract extends SuperAbstract implements AsSuper

{

public void b(){…}

abstract String f();

}

public class InheritAbstract extends SubAbstract{

public void x(){…}

public int c(int i ) {…}

public String f(){…}

public static void main(String args[]){

InheritAbstract instance=new InheritAbstract();

instance.x();

instance.a();

instance.b();

instance.c(100);

System.out.println(instance.f());

}

}

在以上这段程序中：

抽象类有：SuperAbstract和　（1）　　(写出类名)

非抽象类有：　　　　　　　（2）　　(写出类名)

接口有：　　　　　　　　　（3）　　(写出接口名)

AsSuper中的x()方法是（4）方法，所以在InheritAbstract中必须对它进行（5）

**答案：**

**(1) SuperAbstract；**

**(2) InheritAbstract；**

**(3) AsSuper；**

**(4) 抽象；**

**(5) 覆盖和实现。**

**3.** 按注释完成程序

public class Leaf {

private int i = 0; //此属性值用于检验

Leaf increment(){ //定义方法increment()，返回值是Leaf类的对象

i++;

return **(1)** ;//将当前对象的地址作为返回值返回

}

void print() {

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

}

public static void main(String args[]){

Leaf x =  **(2)**; //创建Leaf类的对象x

x.increment().increment().increment().print();

}//多次调用方法increment()，返回的都是x的地址，i 值表示调用次数

}

输出结果为 i =  **(3)**

**答案：**

1. **this；**
2. **new Leaf()；**
3. **3**

**4.**  按注释提示完成文件复制的程序

//FileStream源代码如下：

import java.io.\*;

class FileStream {

public static void main(String args []) {

try {

File inFile = new File("file1.txt"); //指定源文件

File outFile = new File("file2.txt"); //指定目标文件

FileInputStream fis =**（1）;**

FileOutputStream fos = new FileOutputStream(outFile);

int c;

//逐字节从源文件中输入，再输出到fos流

while ((c = fis.read ())!=-1)

**（2）;**

fis.close();

fos.close();

}

catch (Exception e) {

System.out.println("FileStreamsTest: "+e);

}

}

}

**答案：**

**(1) new FileInputStream(inFile);**

**(2) fos.write(c);**

**5.** 阅读程序，给出结果：

// AbstractClassDemo.java源代码如下：

abstract class Shape { //定义抽象类Shape和抽象方法display

abstract void display();

}

class Circle extends Shape {

void display() { //实现抽象类的方法

System.out.println("Circle");

}

}

class Rectangle extends Shape {

void display() { //实现抽象类的方法

System.out.println("Rectangle");

}

}

class Triangle extends Shape {

void display() { //实现抽象类的方法

System.out.println("Triangle");

}

}

public class AbstractClassDemo{

public static void main(String args[]){

(new Circle()).display(); //定义无名对象来调用对应的display方法

(new Rectangle()).display();

(new Triangle()).display();

}

}

输出结果是 ?

**答案：(1) Circle； (2) Rectangle； (3) Triangle。**

**《JAVA程序设计》期末考试试题 （一）**

一、单选择题

**1、**编译Java Application 源程序文件将产生相应的字节码文件，这些字节码文件的扩展名为( B )。

A. java B. .class

C. html D. .exe

**2、**设 x = 1 , y = 2 , z = 3，则表达式 y＋＝z－－/＋＋x 的值是( A. )。

A. 3 B. 3. 5

C. 4 D. 5

**3、**不允许作为类及类成员的访问控制符的是( C. )。

A. public B. private

C. static D. protected

**4、**为AB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AB作为前缀就可以调用它，该方法头的形式为( A. )。

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

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

二、填空题

**1、**开发与运行Java程序需要经过的三个主要步骤为 编辑源程序 、

编译生成字节码 和 解释运行字节码 。

**2、**在Java的基本数据类型中，char型采用Unicode编码方案，每个Unicode码占

用 2 字节内存空间，这样，无论是中文字符还是英文字符，都是占

用 2 字节内存空间。

**3、**设 x = 2 ，则表达式 ( x + + )／3 的值是 0 。

**4、**若x = 5，y = 10，则x < y和x >= y的逻辑值分别为 true 和 false 。

**5、** 抽象(abstract) 方法是一种仅有方法头，没有具体方法体和操作实现的方法，该方法必须在抽象类之中定义。 最终(final) 方法是不能被当前类的子类重新定义的方法。

**6、**创建一个名为 MyPackage 的包的语句是package MyPackage ; ，

该语句应该放在程序的位置为： 应该在程序第一句 。

**7、**设有数组定义：int MyIntArray[ ] = { 10 , 20 , 30 , 40 , 50 , 60 , 70}; 则执行以下几个语句后的输出结果是 120 。

int s = 0 ;

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

if ( i % 2 = = 1 )

s += MyIntArray[i] ;

System.out.println(s);

**8、**在Java程序中，通过类的定义只能实现 单 重继承，但通过接口的定义可以实现 多 重继承关系。

三、写出下面程序的运行结果

**1、** import java.io.\*;

public class abc

{

public static void main(String args [ ])

{

AB s = new AB("Hello!","I love JAVA.");

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

}

}

class AB {

String s1;

String s2;

public AB(String str1, String str2)

{

s1 = str1;

s2 = str2;

}

public String toString( )

{

return s1+s2;

}

}

运行结果：Hello! I love JAVA.

**2、** import java.io.\* ;

public class abc

{

public static void main(String args[ ])

{ int i, s = 0 ;

int a[ ] = { 10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 };

for ( i = 0 ; i < a.length ; i ++ )

if ( a[i]%3 = = 0 ) s += a[i] ;

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

}

}

运行结果：s = 180

**3、**import java.io.\* ;

public class abc

{

public static void main(String args[ ])

{

System.out.println("a="+a+"\nb="+b);

}

}

class SubClass extends SuperClass

{ int c;

SubClass(int aa, int bb, int cc)

{

super(aa, bb);

c=cc;

}

}

class SubSubClass extends SubClass

{ int a;

SubSubClass(int aa, int bb, int cc)

{ super(aa, bb, cc);

A = aa+bb+cc;

}

void show()

{

System.out.println("a="+a+"\nb="+b+"\nc="+c);

}

}

运行结果：a=60

b=20

c=30

**《JAVA语言程序设计》期末考试模拟试题**

**2008年6月**

**一、单选择题(每小题2分，共10分)**

**1、编译Java Application 源程序文件将产生相应的字节码文件，这些字节码文件的扩展名为( B )。**

**A. .java B. .class**

**C. .html D. .exe**

**2、设 x = 1 , y = 2 , z = 3，则表达式 y＋＝z－－/＋＋x 的值是( A )。**

**A. 3 B. 3. 5**

**C. 4 D. 5**

**3、在Java Applet程序用户自定义的Applet子类中，一般需要重载父类的( D )方法来完成一些画图操作。**

**A. start( ) B. stop( )**

**C. init( ) D. paint( )**

**4、不允许作为类及类成员的访问控制符的是( C )。**

**A. public B. private**

**C. static D. protected**

**5、为AB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AB作为前缀就可以调用它，该方法头的形式为( A )。**

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

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

**二、填空题（每空格1分，共20分）**

**1、开发与运行Java程序需要经过的三个主要步骤为 编辑源程序、**

**编译生成字节码和解释运行字节码。**

**2、如果一个Java Applet源程序文件只定义有一个类，该类的类名为MyApplet，则类MyApplet必须是 Applet 类的子类并且存储该源程序文件的文件名为 MyApplet 。**

**3、如果一个Java Applet程序文件中定义有3个类，则使用Sun公司的JDK编译**

**器 javac.exe 编译该源程序文件将产生 3 个文件名与类名相同而扩展名为**

**. class 的字节码文件。**

**4、在Java的基本数据类型中，char型采用Unicode编码方案，每个Unicode码占**

**用 2 字节内存空间，这样，无论是中文字符还是英文字符，都是占**

**用 2 字节内存空间。**

**5、设 x = 2 ，则表达式 ( x + + )／3 的值是 0 。**

**6、若x = 5，y = 10，则x < y和x >= y的逻辑值分别为 ture 和 false 。**

**7、 抽象(abstract)方法、 方法是一种仅有方法头，没有具体方法体和操作实现的方法，该方法必须在抽象类之中定义。 最终(final)方法 是不能被当前类的子类重新定义的方法。**

**8、创建一个名为 MyPackage 的包的语句是package MyPackage ;**

**语句应该放在程序的位置为：程序第一句**

**9、设有数组定义：int MyIntArray[ ] = { 10 , 20 , 30 , 40 , 50 , 60 , 70}; 则执行以下几个语句后的输出结果是 120 。**

**int s = 0 ;**

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

**if ( i % 2 = = 1 ) s += MyIntArray[i] ;**

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

**10、在Java程序中，通过类的定义只能实现 单 重继承，但通过接口的定义可以实现 多 重继承关系。**

**三、写出下列程序完成的功能。(每小题5分，共20分)**

**1、public class Sum**

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

**{ double sum = 0.0 ;**

**for ( int i = 1 ; i <= 100 ; i + + )**

**sum += 1.0/(double) i ;**

**System.out.println( "sum="+sum );**

**}**

**}**

**1、计算 1/1+1/2+1/3+...+1/100 的值。**

**2、 import java.io.\* ;**

**public class Reverse**

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

**{ int i , n =10 ;**

**int a[ ] = new int[10];**

**for ( i = 0 ; i < n ; i ++ )**

**try {**

**BufferedReader br = new BufferedReader(**

**new InputStreamReader(System.in));**

**a[i] = Integer.parseInt(br.readLine( )); // 输入一个整数**

**} catch ( IOException e ) { } ;**

**for ( i = n－1 ; i >= 0 ; i ―― )**

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

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

**}**

**}**

**2、从标准输入(即键盘)读入10个整数存入整型数组a中，然后逆序输出这10个整数。**

**3、 import java.awt.\*;**

**public class abc**

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

**{ new FrameOut(); }**

**}**

**class FrameOut extends Frame // Frame为系统定**

**{ Button btn; // 义的窗框类**

**FrameOut( )**

**{ super("按钮");**

**btn = new Button("按下我");**

**setLayout(new FlowLayout( ));**

**add(btn);**

**setSize(300,200);**

**show( );**

**}**

**}**

**4、求两个数的最大值。**

**3、创建一个标题为"按钮"的窗框，窗框中显示有"按下我"字样的按钮。**

**4、import java.io.\*;**

**public class abc**

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

**{ SubClass sb = new SubClass( );**

**System.out.println(sb.max( ));**

**}**

**}**

**class SuperClass**

**{ int a = 10 , b = 20 ; }**

**class SubClass extends SuperClass**

**{ int max( ) { return ((a>b)?a:b); } }**

**四、写出下面程序的运行结果(每小题10分，共30分)**

**1、 import java.io.\*;**

**public class abc**

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

**{ AB s = new AB("Hello!","I love JAVA.");**

**System.out.println(s.toString( ));**

**}**

**}**

**class AB {**

**String s1;**

**String s2;**

**AB( String str1 , String str2 )**

**{ s1 = str1; s2 = str2; }**

**public String toString( )**

**{ return s1+s2;}**

**}**

**1、Hello! I love JAVA.**

**2、 import java.io.\* ;**

**public class abc**

**{**

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

**{ int i , s = 0 ;**

**int a[ ] = { 10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 };**

**for ( i = 0 ; i < a.length ; i ++ )**

**if ( a[i]%3 = = 0 ) s += a[i] ;**

**System.out.println("s="+s);**

**}**

**}**

**2、s = 180**

**3、import java.io.\* ;**

**public class abc**

**{**

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

**)**

**{ System.out.println("a="+a+"\nb="+b); }**

**}**

**class SubClass extends SuperClass**

**{ int c;**

**SubClass(int aa,int bb,int cc)**

**{ super(aa,bb);**

**c=cc;**

**}**

**}**

**class SubSubClass extends SubClass**

**{ int a;**

**SubSubClass(int aa,int bb,int cc)**

**{ super(aa,bb,cc);**

**a=aa+bb+cc;**

**}**

**void show()**

**{ System.out.println("a="+a+"\nb="+b+"\nc="+c); }**

**}**

**3、a=60**

**b=20**

**c=30**

**五、使用Java语言编写程序。(每小题10分，共20分)**

**1、编写一个字符界面的Java Application 程序，接受用户输入的10个整数，并输出这10个整数的最大值和最小值。**

**答:1、参考程序如下：**

**import java.io.\* ;**

**public class abc**

**{**

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

**{ int i , n = 10 , max = 0 , min = 0 , temp = 0;**

**try {**

**BufferedReader br = new BufferedReader(**

**new InputStreamReader(System.in));**

**max = min = Integer.parseInt(br.readLine( ));**

**} catch ( IOException e ) { } ;**

**for ( i = 2 ; i <= n ; i ++ ) {**

**try {**

**BufferedReader br = new BufferedReader(**

**new InputStreamReader(System.in));**

**temp = Integer.parseInt(br.readLine( ));**

**if (temp > max ) max=temp;**

**if (temp < min) min=temp;**

**} catch ( IOException e ) { } ;**

**}**

**System.out.println("max="+max+"\nmin="+min);**

**}**

**}**

**2、编写一个完整的Java Applet 程序使用复数类Complex验证两个复数 1+2i 和3+4i 相加产生一个新的复数 4+6i 。**

**复数类Complex必须满足如下要求：**

**(1) 复数类Complex 的属性有：**

**RealPart : int型，代表复数的实数部分**

**ImaginPart : int型，代表复数的虚数部分**

**(2) 复数类Complex 的方法有：**

**Complex( ) : 构造函数，将复数的实部和虚部都置0**

**Complex( int r , int i ) : 构造函数，形参 r 为实部的初值，i为虚部的初值。**

**Complex complexAdd(Complex a) : 将当前复数对象与形参复数对象相加，所得的结果仍是一个复数值，返回给此方法的调用者。**

**String ToString( ) : 把当前复数对象的实部、虚部组合成 a+bi 的字符串形式，其中a 和 b分别为实部和虚部的数据。**

**答: 2、参考程序如下：**

**import java.applet.\* ;**

**import java.awt.\* ;**

**public class abc extends Applet**

**{**

**Complex a,b,c ;**

**public void init( )**

**{**

**a = new Complex(1,2);**

**b = new Complex(3,4);**

**c = new Complex();**

**}**

**public void paint(Graphics g)**

**{**

**c=a.complexAdd(b);**

**g.drawString("第一个复数："+a.toString(),10,50);**

**g.drawString("第二个复数："+b.toString(),10,70);**

**g.drawString("两复数之和："+c.toString(),10,90);**

**}**

**}**

**class Complex**

**{**

**int RealPart ; // 复数的实部**

**int ImaginPart ; // 复数的虚部**

**Complex() { RealPart = 0 ; ImaginPart = 0 ; }**

**Complex(int r , int i)**

**{ RealPart = r ; ImaginPart = i ; }**

**Complex complexAdd(Complex a)**

**{**

**Complex temp = new Complex( ); // 临时复数对象**

**temp.RealPart=RealPart+a.RealPart;**

**temp.ImaginPart=ImaginPart+a.ImaginPart;**

**return temp;**

**}**

**public String toString( )**

**{ return ( RealPart+" + "+ImaginPart+" i "); }**

**}**