

杭州电子科技大学学生考试卷（ A ）卷

考试课程	面向对象程序设计(Java)		考试日期	2020 年 月 日		成 绩	
课程号		教师号		任课教师姓名			
考生姓名		学号 (8 位)		年 级		专 业	

一、 选择题 （共 20 题，每题 1.5 分，共 30 分）

1. 运行名为 Test.class 的 Java 字节码文件，所用的 Java 命令是()
- A、javac Test.java

B、javac Test.class

C、java Test

D、java Test.class
2. 下列不可作为 java 语言标识符的是？（ ）
- A、a2

B、\$2

C、_2

D、22
3. 以下关于运算符优先级顺序的描述中正确的是？（ ）
- A、关系运算符<算术运算符<赋值运算符

B、赋值运算符<关系运算符<算术运算符

C、算术运算符<关系运算符<赋值运算符

D、赋值运算符<算术运算符<关系运算符
4. getCustomerInfo() 方法如下，try 中可以捕获三种类型的异常，如果在该方法中产生了一个 IOException，将会输出什么结果（ ）
- ```
public void getCustomerInfo() {
 try {
 // do something that may cause an Exception
 } catch (java.io.FileNotFoundException ex){
 System.out.print("FileNotFoundException!");
 } catch (java.io.IOException ex){
 System.out.print("IOException!");
 } catch (java.lang.Exception ex){
 System.out.print("Exception!");
 }
}
```
- A、IOException!

B、IOException!Exception!

C、FileNotFoundException!IOException!

D、FileNotFoundException!IOException!Exception!
5. 声明成员变量时，如果不使用任何访问控制符（），则以下哪种类型的类不能对该成员进行直接访问（ ）
- A、同一类

B、同一包中的子类

C、同一包中的非子类

D、不同包中的子类

6. 对于下列代码：

```
String str1 = “java”;
String str2 = “java”;
String str3 = new String(“java”);
StringBuffer str4 = new StringBuffer(“java”)
```

下列结论是正确的（ ）

- A、str1==str2

B、str1==str4

C、str2==str3

D、str3==str4

7. 关于接口，下面的叙述错误的是（ ）

- A、一个接口可以多继承多个接口
- B、一个类可以实现多个接口
- C、抽象类在实现接口时，可以不实现该接口中声明的所有方法
- D、抽象类在实现接口时，必须实现该接口中声明的所有方法

8. 以下程序运行的结果是（ ）

```
Public class Test{
 public int div (int a, int b){
 try{
 return a / b;
 }catch(Exception e){
 System.out.println("Exception");
 } catch(NullPointerException e){
 System.out.println("NullPointerException");
 } catch(ArithmeticException e){
 System.out.println("ArithmeticException");
 }finally{
 System.out.println(“finally”);
 }
 return 0;
 }
 public static void main(String[] args){
 Test demo = new Test();
 System.out.println(“商是： ”+ demo.div(9, 10));
 }
}
```

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div><div>A、Exception finally 商是：0</div><div>B、ArithmeticException finally 商是：0</div><div>C、finally 商是：0</div><div>D、编译报错</div></div> <div>9. 在某个类中存在一个方法：void getSort(int x)，以下能作为这个方法的重载的声明的是（ ）</div> <div><div>A、public get (float x)</div><div>B、int getSort (int y)</div><div>C、double getSort (int x, int y)</div><div>D、void get (int x, int y){}</div></div> <div>10. 以下哪句是错误的？（ ）</div> <div><div>A、编译器会搜索要 import 的类的编译结果而不是源文件</div><div>B、import 是要告诉编译器要 import 的包的信息</div><div>C、import 是要把 import 的类的源代码插入到 import 语句所在的地方</div><div>D、编译器在编译的时候需要能访问得到 import 的类的编译结果文件</div></div> <div>11. 在 Java 中，关于 HashMap 类的描述，以下选项错误的是（ ）</div> <div><div>A、HashMap 使用键/值的形式保存数据</div><div>B、HashMap 能够保证其中元素的顺序</div><div>C、HashMap 允许将 null 用作键</div><div>D、HashMap 允许将 null 用作值</div></div> <div>12. 以下说法错误的是（ ）</div> <div><div>A、静态方法可以直接访问本类的静态变量和静态方法</div><div>B、静态方法可以直接访问本类的非静态变量和非静态方法</div><div>C、非静态方法可以直接访问本类的静态变量和静态方法</div><div>D、非静态方法可以直接访问本类的非静态变量和非静态方法</div></div> <div>13. 下列哪个选项定义的抽象类是正确的？（ ）</div> <div><div>A、public abstract final class Car {abstract void move();}</div><div>B、public class abstract Car {abstract void move();}</div><div>C、public abstract class Car {abstract void drive();}</div><div>D、public abstract class Car {abstract void drive() { }}</div></div> <div>14. 以下程序段输出的结果是（ ）</div> <div><pre>public class Test{<br/>    public static void main(String args[]){<br/>        String str = "ABCDE";<br/>        str.substring(3);<br/>        str.concat("XYZ");<br/>        System.out.print(str);<br/>    }<br/>}</pre></div> <div><div>A、DE</div><div>B、DEXYZ</div><div>C、ABCDE</div><div>D、CDEXYZ</div></div> | <div>15. 在下面程序中的括号中，如下哪些代码可以创建并启动线程？（ ）</div> <div><pre>MyRunnable implements Runnable {public void run(){()}}</pre></div> <div><div>A、new Runnable(MyRunnable).start()</div><div>B、new Thread(MyRunnable).run ()</div><div>C、new Thread(new MyRunnable()).start()</div><div>D、new MyRunnable().start()</div></div> <div>16. 如下程序输出（ ）</div> <div><pre>public class Test{<br/><br/>    public static void main(String[] args){<br/>        String s = "Java";<br/>        StringBuilder buffer = new StringBuilder(s);<br/>        Change(s);<br/>        System.out.println(s);<br/>    }<br/><br/>    Private static void change(String s){<br/>        s = s + " and HTML";<br/>    }<br/>}</pre></div> <div><div>A、Java;</div><div>B、Java and HTML;</div><div>C、and HTML;</div><div>D、什么都不显示;</div></div> <div>17. 在使用 super 和 this 关键字时，以下描述正确的是（ ）</div> <div><div>A、在子类构造方法中使用 super () 显示调用父类的构造方法，super () 必须写在子类构造方法的第一行，否则编译不通过</div><div>B、this () 和 super () 可以同时出现在一个构造函数中</div><div>C、super () 和 this () 不一定要放在构造方法内第一行</div><div>D、this () 和 super ()可以在 static 环境中使用，包括 static 方法和 static 语句块</div></div> <div>18. 下面语句序列执行后，k 的值是什么？（ ）</div> <div><pre>int x = 6, y = 10, k = 5;<br/>switch (x%y) {<br/>    case 0: k=x*y<br/>    case 6: k=x/y<br/>    case 12: k=x-y<br/>    default: k=x*y - x;<br/>}</pre></div> <div><div>A、60</div><div>B、5</div><div>C、0</div><div>D、54</div></div> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

19. 下面关于缺省构造方法的描述中正确的是( )。

- A、 当类中没有定义任何构造方法时, Java 编译器将为这个类创建缺省构造方法
- B、 缺省构造方法可以初始化其他方法中定义的变量
- C、 Java 编译器会为所有的类创建缺省构造方法
- D、 如果在一个类中定义的构造方法都声明了参数, Java 编译器将为这个类创建一个缺省构造方法

20. 下列哪个流类属于面向字符的输入流( )。

- A、 `BufferWriter`;
- B、 `FileInputStream`;
- C、 `ObjectInputStream`;
- D、 `InputStreamReader`;

二、 程序阅读及填空题（共 20 空，每空 2 分，共 40 分）

1. 请在空格中填写以下程序执行的结果:

```
public class IdentifyMyParts {
 public static int x = 7;
 public int y = 3;
}

public class Test {
 public static void main(String[] args) {
 IdentifyMyParts a = new IdentifyMyParts ();
 IdentifyMyParts b = new IdentifyMyParts ();
 a.y = 5;
 b.y = 6;
 a.x = 1;
 b.x = 2;
 System.out.println("a.y = " + a.y);
 System.out.println("b.y = " + b.y);
 System.out.println("a.x = " + a.x);
 System.out.println("b.x = " + b.x);
 }
}
```

屏幕输出为 1

屏幕输出为 2

屏幕输出为 3

屏幕输出为 4

2. 请在空格中填写以下程序执行的结果:

```
public class java1 {
 public static void main(String [] args){
 int arr[][] = {{34,21,45,67,20},{23,10,3,45,76},{22,3,79,56,50}};
 int i = 0;
 int j = 0;
 int min = arr[0][0];
```

```

while (i<3){
 while (__ 5 __){
 if(arr[i][j]<min)
 min = arr[i][j];
 j++;
 }
 __ 6 __ ;
 __ 7 __ ;
}
System.out.println("The min: " + min);
}
}

```

3. 写出下列程序的输出结果。

```
class Exam31 {
 public static void main(String[] args){
 int i=1, j= 2, k = 3;
 boolean b = --i > 1 && j++ < 2 && k-- < 3;
 b = ++i > 1 || j-- < 2 && --k > 3;
 b = ++i > 1 | j-- < 2 & --k > 3;
 System.out.println("i = " + i); 屏幕输出为 8
 System.out.println("j = " + j); 屏幕输出为 9
 System.out.println("k = " + k); 屏幕输出为 10
 }
}
```

4. 请写出以下程序运行结果:

```
Public class Main {
 public static void main(String[] args){
 String s = "hello";
 Try{
 s = s + " world";
 s.toUpperCase();
 String[] a = s.split("o");
 }
 }
}
```

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>        System.out.println(a.length);     }catch(Exception e){         System.out.print(s);    屏幕输出为__11__     }finally{         System.out.print(s);    屏幕输出为__12__     } } }</pre> <p>5. 以下程序为类 Animal、Bird、Parrot 和 Test 的代码，请将程序填充完整，并填写输出结果。包结构为：com.au 中有 Animal 类；com.bd 中有 Bird，Parrot，Test 类</p> <p><b>package com.au;</b></p> <pre>public __13__ class Animal{     __14__ String color;     public abstract void sound();     public Animal(){         color="";         System.out.println("动物");     } }</pre> <p><b>package com.bd;</b></p> <pre>public class Bird __15__ Animal{     public Bird (){         color = "绿色";         System.out.println("鸟");     }     public void sound(){         System.out.println("鸟叫声");     }     public void sound(String msg) {         System.out.println("鸟发出的叫声： " + msg);     } }</pre> <pre>public class Parrot extends Bird{     public Parrot(){         System.out.println("鹦鹉");     }     public void sound(){         __16__ //调用父类的方法 sound ()     } }</pre> | <pre>public class Test{     public static void main(String[] args){         Parrot p = new Parrot ();         p.sound();     } }</pre> <p>6. 在横线处填上合适的语句。</p> <p><b>//MyException 为用户自定义异常类</b></p> <pre>public class MyException extends Exception {     //用户自定义异常类 }</pre> <pre>class Person{     //在下行空白处写出声明方法可能抛出 MyException 异常的语句      public void check(String strage) __17__ {         int age = Integer.parseInt(strage);         if (age &lt; 0) {             /*在下行空白处填上抛出 MyException 异常对象的语句，异常信息描述为 “年龄不能为负数!” */             __18__         }     } }</pre> <p>7. 如果一个整数 a 能被数 b 整除，则 b 是 a 的因子。下面是一个 Java 程序，它的功能是求给定整数 n 的所有因子的和 sum，请将程序补充完整。</p> <pre>public class Test{     public static void main(String args[]){         int n=6,sum=0;         for(int k=1; __19__;k++) {             if(__20__)                 sum+=k;         }         System.out.println(sum);     } }</pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

三、 编程题 （共3题，每题10分，共30分）

1. 定义基类 Point（点）和派生类 Circle（圆），求圆的周长。Point 类有两个私有的数据成员 float x, y; Circle 类新增一个私有的数据成员半径 float r 和一个公有的求周长的函数 getCircumference( ); 请编写 Point 和 Circle 类以及测试类。

2. 编写一个应用程序实现文件的拷贝，源文件( SourceFile.txt)及目标文件(TargetFile.txt)通过命令行参数得到。(注：可以用输入输出流 FileInputStream 和 FileOutputStream，或者 FileReader 和 FileWrite，或者 BufferedReader 和 BufferedWrite 等)

3. 编写一个完整的 Java Application 程序。包含类 Shape、类 Oval、类 ShapeTest，具体要求如下：

（1）编写一个抽象类 Shape 表示形状对象，包含以下成员

- 1) 属性：double 型常量 PI，其值为 3.1415926；
- 2) 方法：包含 double area ( )和 double perimeter ( )两个抽象方法；

（2）编写一个 Shape 类的子类 Oval，表示椭圆对象，包含以下成员

- 1) 属性：double 型私有成员变量 a，表示长轴半径；double 型私有成员变量 b，表示短轴半径；
- 2) 方法：主要包含：构造方法 Oval (double a, double b)，用参数设置椭圆的长轴半径和短轴半径；构造方法 Oval(), 将椭圆的长轴半径和短轴半径都初始化为 0；成员方法 double area ( ), 重写 Shape 类中的 area 方法，返回椭圆的面积；成员方法 double perimeter ( ), 重写 Shape 类中的 perimeter 方法，返回椭圆的周长；方法 public String toString( ), 将当前的椭圆对象转换成字符串形式，例如长轴半径为 10.0，短轴半径为 5，返回字符串 “Oval(a:10.0, b:5.0)”。

（3）编写公共类 Main，实现如下功能

- 1) 创建一个长轴半径为 8.0，短轴半径为 6.0 的椭圆对象；
- 2) 分别用 area 和 perimeter 方法，求出以上椭圆的面积和宽度并输出，输出过程中要求使用到 toString 方法。

杭州电子科技大学学生考试卷（ A ）卷

|      |                |         |      |            |  |     |  |
|------|----------------|---------|------|------------|--|-----|--|
| 考试课程 | 面向对象程序设计(Java) |         | 考试日期 | 2018 年 月 日 |  | 成 绩 |  |
| 课程号  |                | 教师号     |      | 任课教师姓名     |  |     |  |
| 考生姓名 |                | 学号(8 位) |      | 年级         |  | 专业  |  |

一、 选择题 (每题 1.5 分,共 30 分)

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|    |    |    |    |    |    |    |    |    |    |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|    |    |    |    |    |    |    |    |    |    |

二、 程序阅读及填空题 (每题 2 分,共 40 分)

- (1) \_\_\_\_\_ (2) \_\_\_\_\_ (3) \_\_\_\_\_ (4) \_\_\_\_\_
- (5) \_\_\_\_\_ (6) \_\_\_\_\_ (7) \_\_\_\_\_ (8) \_\_\_\_\_
- (9) \_\_\_\_\_ (10) \_\_\_\_\_ (11) \_\_\_\_\_
- (12) \_\_\_\_\_ (13) \_\_\_\_\_
- (14) \_\_\_\_\_ (15) \_\_\_\_\_
- (16) \_\_\_\_\_
- (17) \_\_\_\_\_ (18) \_\_\_\_\_
- (19) \_\_\_\_\_ (20) \_\_\_\_\_

三、 程序设计题（每题 10 分，共 30 分）

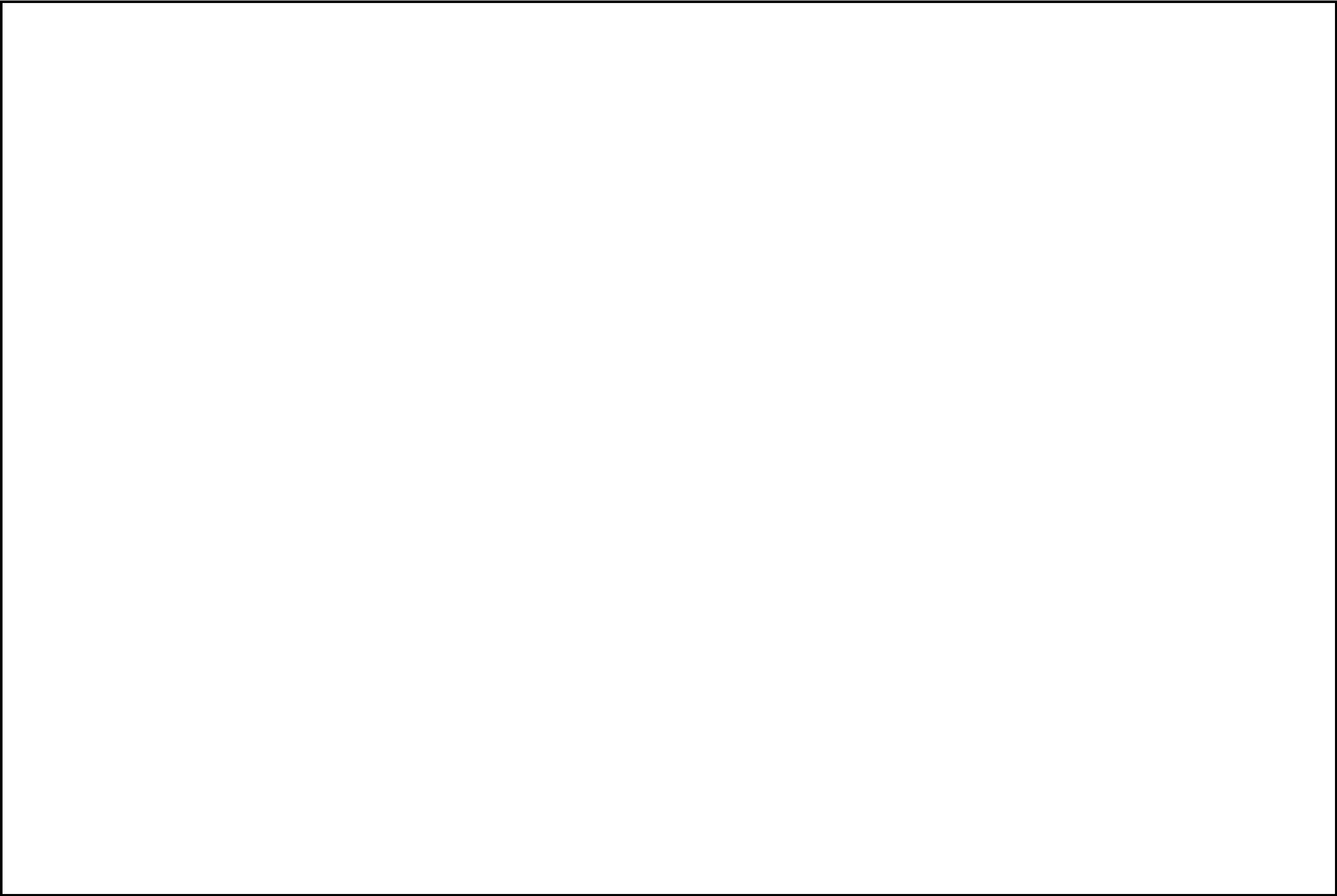
1.

杭州电子科技大学学生考试卷（ A ）卷 答题纸

|       |                      |         |      |                 |  |    |    |     |  |
|-------|----------------------|---------|------|-----------------|--|----|----|-----|--|
| 考试课程  | Java 面向对象<br>程序设计（甲） |         | 考试日期 | 2018 年 1 月    日 |  |    | 成绩 |     |  |
| 课 程 号 |                      | 教 师 号   |      | 任课教师姓名          |  |    |    |     |  |
| 考生姓名  |                      | 学号（8 位） |      | 年级              |  | 专业 |    | 座位号 |  |

2.

3.



杭州电子科技大学学生考试卷（ A ）卷

|      |                |          |      |            |  |     |  |
|------|----------------|----------|------|------------|--|-----|--|
| 考试课程 | 面向对象程序设计(Java) |          | 考试日期 | 2020 年 月 日 |  | 成 绩 |  |
| 课程号  |                | 教师号      |      | 任课教师姓名     |  |     |  |
| 考生姓名 |                | 学号 (8 位) |      | 年 级        |  | 专 业 |  |

一、 选择题 (每题 1.5 分,共 30 分)

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| C  | D  | B  | A  | D  | A  | D  | D  | C  | C  |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| B  | B  | C  | C  | C  | A  | A  | D  | A  | D  |

二、 程序阅读及填空题 (每题 2 分,共 40 分)

- (1) 5 (2) 6. (3) 2 (4) 2
- (5) j<5 (6) i=0 (7) i++ (8) 2
- (9) 0 (10) 2 (11) 1
- (12) helloworld (13) abstract
- (14) protected 或 public (15) extends
- (16) super.sound();
- (17) throws ServerTimedException (18) throw new ServerTimedException()
- (19) k<=n (20) n%k==0

三、 程序设计题 (每题 10 分,共 30 分)

1. 4 分
- class Point {  
private float x;  
private float y;  
public Point(float x, float y) {  
this.x = x;  
this.y = y;  
}  
public float getX() {  
return x;  
}  
public void setX(float x) {  
this.x = x;  
}  
}
- 3 分
- class Circle extends Point{  
private float r;  
public Circle(float x, float y,float r) {  
super(x, y);  
this.r=r;  
}  
public double getCircumference(){  
return 2\*r\*3.14;  
}  
}
- 3 分
- public class Test {  
public static void main(String args[]) {  
Circle circle = new Circle(1, 2,3);  
double circumference = circle.getCircumference();  
System.out.println(circumference);  
}  
}



杭州电子科技大学学生考试卷（ A ）卷 答题纸

|       |                      |         |      |                 |  |    |    |     |  |
|-------|----------------------|---------|------|-----------------|--|----|----|-----|--|
| 考试课程  | Java 面向对象<br>程序设计（甲） |         | 考试日期 | 2018 年 1 月    日 |  |    | 成绩 |     |  |
| 课 程 号 |                      | 教 师 号   |      | 任课教师姓名          |  |    |    |     |  |
| 考生姓名  |                      | 学号（8 位） |      | 年级              |  | 专业 |    | 座位号 |  |

2.

```
public static void main(String[] args) {
 copy("SourceFile.txt","TargetFile.txt");
 System.out.println("copy success!");
}

private static void copy(String src,String target) {
 InputStream in = null;
 OutputStream out = null;
 try {
 in = new FileInputStream(new File(src));
 out = new FileOutputStream(new File(target));
 byte[] bytes = new byte[1024];
 int len = -1;
 while((len = in.read(bytes))!=-1) {
 out.write(bytes,0,len);
 }
 } catch (FileNotFoundException e) {
 e.printStackTrace();
 } catch (IOException e) {
 e.printStackTrace();
 } finally {
 try {
 if(in != null) {
 in.close();
 }
 if(out != null) {
 out.close();
 }
 } catch (IOException e) {
 e.printStackTrace();
 }
 }
}
```

2 分

4 分

2 分

3.

```
class Application {
 abstract class Shape {
 public final double PI=3.1415926;
 abstract double area();
 abstract double perimeter();
 }

 class Oval extends Shape {
 private double a;
 private double b;

 public Oval(double a, double b) {
 this.a = a;
 this.b = b;
 }

 public Oval() {
 this.a = 0;
 this.b = 0;
 }

 @Override
 double area() {
 return PI*a*b;
 }

 @Override
 double perimeter() {
 return 2*PI*b+4*(a-b);
 }

 @Override
 public String toString() {
 return "Oval(" +
 "a:" + a +
 ", b:" + b +
 ')';
 }
 }
}
```

3 分

6 分

```
 }
}

public static void main(String[] args) {
 Oval oval = new Application().new Oval(8.0,6.0);
 System.out.println(oval.toString()+":"+oval.area());
 System.out.println(oval.toString()+":"+oval.perimeter());
}
}
```

1 分