

杭州电子科技大学学生考试卷（ A ）卷

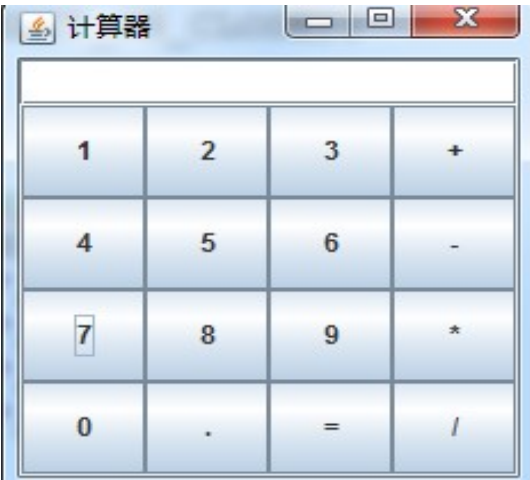
考试课程	Java 语言程序设计	考试日期	2014 年 6 月 16 日	成绩	
课程号		教师号		任课教师姓名	
考生姓名		学号（8 位）		年级	专业

注意：所有答案均写在答卷上，写在试卷上无效；

（一）单选题（每题 2 分，共 30 分）

- (1) 下列哪个类型不属于 Java 的基本数据类型\_\_\_\_\_。
- A、float      B、boolean    C、char      D、String
- (2) 以下哪个不属于 Java 的合法标识符。
- A、Class      B、\_name;              C、abc;              D、1\_name;
- (3) 若整型变量 i 的值为奇数，那么值为 true 的逻辑表达式是( )。
- A、i%2=0;                              B、i%2==1;
- C、i/2==0;                              D、i%2 equals 0;
- (4) 数学表达式 $|x|<10$  对应的 java 表达式为\_\_\_\_\_。
- A、 $|x|<10$       B、 $x<10\&\&x>-10$       C、 $x<10\|x>-10$       D、 $10>x>-10$
- (5) 比较两个字符串 s1 和 s2 的长度是否相等，如下表达式中正确的是 ( )。
- A、s1.equals(s2)                              B、s1.length().equals(s2)
- C、s1.length().equals(s2.length())      D、s1.length() == s2.length()
- (6) 设有定义：int [] a = {4, 2, -7, 5, 1, 6, 3};则 a[a[1]]的值为\_\_\_\_\_。
- A、 4                              B、 2                              C、 -7                              D、 5
- (7) 关于方法的返回值，以下说法正确的是\_\_\_\_\_。
- A、方法必须有返回值                              B、方法可以没有返回值，也可以有多个返回值
- C、方法有且仅有一个返回值                              D、方法必须有至少一个返回值
- (8) 设 Student 为已定义类名，下列语句能正确创建 Student 对象的是\_\_\_\_\_。
- A、Student s=new Student;                              B、Student s=new Student ();
- C、Student s= Student ();                              D、Student s[ ]=new Student [10];
- (9) 对于一个三位正整数 n，表达式( )可以取出 n 的百位数字。
- A、n%10/10      B、n%10      C、n/10%10      D、n/100
- (10) 指出下列哪个方法与方法 public void overload(int x, int y){} 为不合理的重载方法\_\_\_\_\_。
- A、public long overload (long x, long y){}      B、public float overload (float x, float y){}
- C、public void overload (int x){}      D、public int overload (int a, int b){}

- (10) 以下关于构造方法的描述错误的是\_\_\_\_\_。
- A、构造方法的返回类型只能是 void 型。
- B、构造方法是类的一种特殊方法，它的方法名必须与类名相同。
- C、构造方法的主要作用是完成对类的对象的初始化工作。
- D、一个类可以定义多个构造方法。
- (11) 下列各语句序列中，能够将变量 u、s 中最大值赋值到变量 t 中的是 ( )
- A、 if (u>s) t=u; t=s;
- B、 t=s; if (u>s) t=u;
- C、 if (u>s) t=s; else t=u;
- D、 t=u; if (u>s) t=s;
- (12) \_\_\_\_\_定义了一个合法的抽象类。
- A、abstract Shape {abstract void area();}
- B、class abstract Shape {abstract void area();}
- C、abstract class Shape {abstract void area();}
- D、abstract class Shape {abstract void area() {System.out.println("area");}}
- (13) 为了以字符方式将数据写到文件中，可以使用\_\_\_\_\_类。
- A、FileReader                              B、FileInputStream
- C、FileOutputStream                              D、FileWriter
- (15) 如下图中数字和符号按钮所在的面板（JPanel）采用的是下列哪种\_\_\_\_\_布局方式
- A、FlowLayout    B、GridLayout    C、BorderLayout    D、CardLayout



（二）阅读程序，回答每个程序后面的问题（每个问题 2 分，共 20 分）

程序 1

```
class test{
    public static void main(String args[]){
        int a=33;
        do{
            if(a%2==1) a=a*3+1;
            else a/=2;
            System.out.printf("%d  ",a);
        }while(a!=1);
    }
}
```

问题 1：写出程序 1 运行时输出的第 1 个数据。  
问题 2：写出程序 1 运行时输出的最后 1 个数据。

程序 2

```
public class Test{
    static boolean check (int x) {
        if (x==1)
            return false;
        for (int i=2;i<=x/2;i++)
            if (x%i==0)
                return false;
        return true;
    }
    public static void main(String args[]){
        for(int i=5;i<10;i++){
            if(check(i)==false) System.out.println(i);
        }
    }
}
```

问题 3：描述 static boolean check (int x)方法的功能  
问题 4：写出程序 2 运行的输出结果

程序 3

```
public class testComplex{
    public static void main(String[] args) {
        Complex a = new Complex (2, 3);
        Complex b = new Complex (4, 5);
        Complex c = a.add(b);
        System.out.println(c.toString());
    }
}
class Complex {
    private double realPart;
    private double imaginaryPart;
    public Complex(double a, double b) {
        this.realPart=a;
        this.imaginaryPart=b;
    }
    public Complex(){ this.realPart=0; this.imaginaryPart=0; }
    public Complex add(Complex a) {
        double x= this.realPart + a.realPart;
        double y= this.imaginaryPart + a.imaginaryPart;
        return new Complex(x,y);
    }
    public String toString() {
        String show = this.realPart + " + " + this.imaginaryPart + "i";
        return show;
    }
}
```

问题 5：为使程序可以被正常编译、运行，写出保存程序 3 的文件名  
问题 6：写出程序 3 运行的输出结果  
问题 7：程序 3 中，是否有方法重载，请简单说明（不得超过 20 字）。

程序 4

```
public class Class1{
    public static void main(String args[ ]){
        int i,Max,Min;
```

<pre>int a[]= {12,67,8,98,23,56,124,55,99,100}; Max = Min = a[0]; for (i=1;i&lt;a.length;i++) {     if (a[i]&lt;Min) Min = a[i];     if (a[i]&gt;Max) Max = a[i]; } System.out.println(Max+" "+Min); } }</pre> <p>问题 8: 写出程序 4 运行时的输出结果</p> <p><b>程序 5:</b></p> <pre>import javax.swing.*; import java.awt.event.*; import java.awt.*; class AdderFrame extends JFrame implements ActionListener{     private JTextField tfOp1,tfOp2,tfSum;     private JButton btnEqual;     AdderFrame(){         btnEqual = new JButton("=");         tfOp1 = new JTextField(10);         tfOp2 = new JTextField(10);         tfSum = new JTextField(10);         setBounds(200,300,500,80);         btnEqual.addActionListener(this);         setLayout(new FlowLayout());         add(tfOp1); add(tfOp2);         add(btnEqual); add(tfSum);     }     public void actionPerformed(ActionEvent event){         int a=Integer.parseInt(tfOp1.getText());         int b=Integer.parseInt(tfOp2.getText());         tfSum.setText(a+b+"");     }     public static void main(String[] args){         AdderFrame frm = new AdderFrame();         frm.setVisible(true);     } }</pre>	<p>问题 9: 分别写出程序 5 中出现的一个接口名和一个容器类名</p> <p>问题 10: 描述语句 <code>btnEqual.addActionListener(this)</code>的作用（不得超过 20 字）</p> <p><b>（三）程序填空题（每空 2 分，共 14 分）</b></p> <p>（1）输入一个整数，输出数字 7 在整数中出现的次数。例如，输入为-157 时，输出 1；输入 77537 时，输出为 3，请完成程序。</p> <p><b>【程序】</b></p> <pre>import java.util.*; class Test {     public static void main(String []args){         Scanner sc=new Scanner(System.in);         int k, a=sc.nextInt(),count;         if(a&lt;0) a=-a;         while(____(1)____)         {             k=a%10;             ____ (2) ____;             if(k==7){                 count++;             }         }         System.out.println(count);     } }</pre> <p>（2）编写类 Point,用以表示直角坐标系中的点。其中 <code>distance（）</code> 方法返回当前点到坐标原点的距离</p> <pre>public class Point{     private double x,y;     Point(double a ,double b){         ____ (3) ____;     }     double distance( ) {         ____ (4) ____ Math.sqrt(x*x+y*y);     }     public static void main(String args[]){         Point p1=new Point(3,4); //创建坐标为的点 p1(3,4)         System.out.print(____ (5) ____); //输出 p1 到原点的距离     } }</pre>
---	--

(3) 以下程序的功能是：统计文件“d:\source.txt”中“Java”出现的次数，忽略大小写，请将程序补充完整。

```
import java.io.*;
class Count
{
    public static void main(String[] args)throws IOException{
        FileReader in=new FileReader("d:\\source.txt");
        StringBuffer str=_____(6)_____;
        String fix="Java";
        int ch;
        int count=0;
        while((ch=in.read())!=-1)
            str.append((char)(ch));
        for(int i=0;i<str.length()-3;i++)
        {
            String sub=_____(7)_____(i,i+fix.length());
            if(sub.equals (fix)) count++;
        }
        System.out.println(count);
        in.close();
    }
}
```

(四) 程序设计题（每题各 12 分，总共 36 分）

- 1、输入 50 个学生的某门课的成绩，统计不及格学生的人数，并输出平均分。
- 2、输出 1000 以内的素数。要求：素数的判断采用 public boolean isPrime(int n )方法。如果是素数返回 true，否则返回 false。
- 3、定义 Quad，用于描述一个二次函数，要求如下：
  - (1)类中包含 a,b,c 三个实型成员变量，二次项、一次项和常数项系数；
  - (2)构造方法有三个参数，分别用于对三个成员变量初始化。
  - (3)成员方法 getIntersectionNum()返回函数曲线与 x 轴交点数量；
  - (4)在 main 方法中接受键盘输入的三个实数，创建对应的二次函数对象，并调用 getIntersectionNum()方法，计算并输出该二次函数与 x 轴的交点。

杭州电子科技大学学生考试卷（ A ） 卷

考试课程	Java 语言程序设计		考试日期	2014 年 6 月 16 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）单选题（每题 2 分，共 30 分）

(1) D (2) D (3) B (4) B (5) D (6) C (7) B (8) B (9) D (10) D  
(11) A (12) B (13) C (14) D (15) B

（二）程序阅读（每题 2 分，共 20 分）

问题 1 100  
问题 2 1  
问题 3 判断一个整数是否为素数  
问题 4 6 8 9  
问题 5 testComplex.java  
问题 6 6+8i  
问题 7 有，其中的构造方法就是重载  
问题 8 124 8  
问题 9 ActionListener JFrame  
问题 10 btnEqual 注册监听器接口，可触发 ActionEvent 事件

（三）程序填空题（每空 2 分，共 14 分）

(1) a!=0 或者 a>0 (2) a=a/10 或者 a/=10 (3) x=a, y=b 或者 this.x=a; this.y=b;  
(4) return (5) p1.distance() (6) new StringBuffer()  
(7) str.substring

（三）程序设计题（每题 12 分，总计 36 分）

1、

```
import java.util.Scanner;
public class test {
    public static void main(String args[]) {

        Scanner sr= new Scanner(System.in);
        float cj[]=new float[50];
        int bjg=0;
        for(int i=0;i<50;i++){
            cj[i]=sr.nextFloat();
            if(cj[i]<60){
                bjg++;
            }
        }
        System.out.println("不及格人数: "+bjg);
    }
}
```

2、

```
public class test {
    public boolean isPrime(int n) {
        if (n == 1)
            return false;
        for (int i = 2; i <= n / 2; i++)
            if (n % i == 0)
                return false;
        return true;
    }
    public static void main(String args[]) {
        test tst = new test();
        for (int i = 1; i <= 1000; i++) {
            if (tst.isPrime(i))
                System.out.println(i);
        }
    }
}
```

3、

```
import java.util.Scanner;

public class Quad {
    double a, b, c;
    private static Scanner kb;
    Quad(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }
    int getIntersectionNum() {
        double delta = 0;
        int gs = 0;
        if (a == 0) {
            gs = 1;
        }
        else
        {
            delta = b * b - 4 * a * c;
            if (delta > 0)
            {
                gs = 2;
            }
            else if (delta == 0)
            {
                gs = 1;
            }
            else
                gs = 0;
        }
        return gs;
    }
    public static void main(String[] args) {
        kb = new Scanner(System.in);
        double a, b, c;
        a = kb.nextDouble();
        b = kb.nextDouble();
        c = kb.nextDouble();
    }
}
```

```
Quad qd=new Quad(a,b,c);
System.out.println("方程根的个数: "+qd.getIntersectionNum());
}
}
```