

山东大学 2019-2020 学年 一 学期 高级程序设计语言 课程试卷 A

不公开发布

题号	一	二	三	四	五	六	七	八	九	十	总分	阅卷人
得分												

得分	阅卷人

一、单项选择题（15 题，共 30 分）
请注意：将答案写在如下的列表中，写在其他位置无效

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15					

1. 下列说法正确的是（ ）
A. main 方法必须写在类里面
B. 一个 Java 程序用到的类中只能一个类里有 main 方法
C. Java 程序中类名必须与文件名一样
D. main 方法中如果只有一条语句，可以不用{}(大括号)括起来
2. 变量命名规范说法正确的是（ ）
A. 变量名由字母、下划线、数字、\$符号随意组成
B. 变量不能以数字作为开头；
C. A 和 a 在 java 中是同一个变量；
D. main 不可以作为变量名；
3. 下列哪些是合法的标识符（ ）
A. Stu-1 B. 2_nd C. boy&girl D. Stu\$2
4. 下面赋值语句（ ）不会出现编译警告或错误。
A. float f=1.3; B. char c="A"; C. byte b=300; D.int i=100;
5. 下列哪个表达式不能作为循环条件（ ）
A. i++
B. i>5
C. flag= str.equals("q")
D. count==4
6. 下面哪个语句是语法正确的（ ）

- A. String str[]={ 'a', 'b', 'c', 'd', }
B. int[][] a = {{ 1, 2, 3 }, { 4, 5 }, { 7 } };
C. char ca = "r";
D. int i= new int [20];
7. 在 Java 中，关于构造方法，下列说法错误的是()
A. 构造方法的可见修饰符可以是 public
B. 类不一定要显式定义构造函数
C. 构造方法的形参个数必须小于或者等于类的实例变量个数
D. 构造方法绝对不能有返回值
8. 已知表达式 int num[][] = {{ 0, 1, 2 }, { 3, 4, 5 }, { 6, 7, 8 }, { 9, 10, 11 } }; 下面哪个表达式正确？（ ）
A. num.length==4 B. num[0].length==4
C. num.length()==4 D. num[0].length()==3
9. 下面的方法重载，正确的是（ ）
A. int fun(int a, float b){ } B. float fun(int a, float b){ }
 float fun(int a, float b){ } float fun(int x, float y){ }
C. float fun(float a){ } D. float fun1(int a, float b){ }
 float fun(float a, float b){ } float fun2(int a, float b){ }
10. 下列关于关键字 this 和 super 的说法正确的是()
A. super 关键字不仅可以指代子类的直接父类，还可以指代父类的父类
B. 子类通过 super 关键字只能调用父类的方法，而不能调用父类的属性
C. 同一个类的不同对象共用一个 this
D. static 修饰的方法中不能出现 this;
11. 下列方法说法错误的是()
A. 重载 Overload 表示同一个类中可以有多多个名称相同的方法，但这些方法的返回值类型不同
B. 子类中的方法可以与父类中的某个方法的名称和形参表完全相同
C. 使用 final 修饰的方法不能被重写(Override)。
D. Override 是在不同类之间的行为，Overload 是在同一个类中的行为。
12. 下列说法正确的是()
A. 可以使用子类的引用指向父类的对象，可以直接使用该引用调用父类的所有方法。
B. 可以使用父类的引用指向子类的对象，但是此引用不能调用父类中不存在的方法。

- C. 一个指向子类对象的引用赋值给父类类型的引用，必须要通过强制类型转换实现
D. 父类中定义为 private 的属性，在子类中可以继承过来直接使用
13. 以下关于 Object 类的说法错误的是（ ）
A. Exception 类也是从 Object 类派生出来的
B. Java 中的接口(interface)也继承了 Object 类
C. Java 中可以用 Object 类型的引用来指向任意类型的对象
D. 如果自定义类中没有定义 toString()方法，该类创建的对象也可以使用 toString()方法
14. 以下说法，正确的是（ ）
A. abstract 可以与 final 并列修饰同一个类
B 如果一个类的所有方法都用 final 修饰了，则这个类不能有子类
C. 关键字 static 可以用来修饰抽象方法
D. abstract 关键字修饰类，表明该类不能实例化对象
15. 关于 Java 中的接口和继承，下面说法正确的是（ ）
A. 在 Java 中允许继承多个抽象类，在 extends 关键字之后分别列出
B. 在 Java 中一个类只能实现一个接口
C. 一个抽象类如果实现一个接口，可以只实现接口中的部分方法
D. 接口类型不能用于实例化对象，也不能用于声明对象引用

得分	阅卷人

二. 阅读程序，在指定的答题区写出代码的执行结果，注意输出格式（40 分，每题 8 分）

```
1. public class First {
    public static void main(String[] args){
        String s = "sdu";
        int n = s.length()-1;
        int diff;
        for (int i = -n; i<=n ;i++ ) {
            for (int j = -n; j<= n; j++){
                diff = Math.abs(i) - Math.abs(j);
                if (diff <0)
                    System.out.print("*");
                else
                    System.out.print(s.charAt(diff));
            }
            System.out.println();}}}
```

二.1 答题区

```
2. public class Second {
    public static void main(String[] args){
        String s = "Java2020";
        String s1 = s.substring(4);
        String s2 = Integer.parseInt(s1) + 1 + s.substring(0,4);
        String s3 = s1.toLowerCase().compareTo(s)>0?"Java":"2020";
        System.out.println(s1+"\n"+s2+"\n"+s3);
        for (int i = 0; i<s.length(); i++)
            System.out.print((char)(s.charAt(i)+1));
        }
    }

3. public class Third {
    public static void main(String[] args){
        int[][] a = {{1,3,2},{3,2,1}};
        m1(a[0],a[1][0],a[1][1],a[1][2]);
        for (int i=0;i<a.length;i++) {
            for (int j = 0; j < a[i].length; j++)
                System.out.print(a[i][j] + "\t");
            System.out.println();
        }
        static void m1(int[] a,int...b){
            m2(a);
            m2(b);
        }
        static void m2(int[] a){
            for (int i =0; i< a.length-1;i++){
                if (a[i]>a[i+1]){
                    int temp=a[i];
                    a[i]=a[i+1];
                    a[i+1]=temp;
                }
            }
            System.out.println(a[0]+\t" + a[a.length-1]);
        }
    }
}
```

二.2 答题区

三.3 答题区

```
4. public class Fourth {  
  
    private static int[] a= {1,2,3,4};  
    private static int[] b = new int[3];  
    public static void main(String[] args){  
        C(3,4);  
    }  
    public static void C(int m, int n){  
        int i, j;  
        for(i=m; i<=n; i++) {  
            b[m - 1] = i-1;  
            if(m>1)  
                C(m-1,i-1);  
            else {  
                for(j=0;j<3;j++)  
                    System.out.print(a[b[j]] + " ");  
                System.out.println();  
            }  
        }  
    }  
}
```

二.4 答题区

```
5. public class Fifth {  
  
    public static int f(String s) throws Exception{  
        try {  
            return Integer.parseInt(s.substring(1,4));  
        }  
        catch (NumberFormatException e) {  
            System.out.println("Improper score value");  
            return -1;  
        }  
        finally {  
            System.out.println("Finally in f()");  
        }  
    }  
    public static void main(String[] args) {  
        String[] sArray= {"a0123","a45","ab789"};  
        for(int i=0;i<=3;i++) {  
            try {  
                System.out.println(f(sArray[i])+"\n");  
            }  
            catch (Exception e) {  
                System.out.println("Caught OtherException\n");  
            }  
        }  
    }  
}
```

二.5 答题区

```
}  
catch (Exception e) {  
    System.out.println("Caught OtherException\n");  
}  
}  
}  
}
```

得分	阅卷人

三. 写程序（30 分）

1. “逝者如斯夫 不舍昼夜”。人们用四位数字表示一个时间，比如 1106 表示 11 点零 6 分。现在，编写程序实现根据起始时间和终止时间计算流逝的时间。具体要求为：
读入两个数字，第一个数字以上述四位数字格式表示起始时间，第二个数字表示终止时间，计算起始时间和终止时间之间的流逝时间并输出，结果也表示为四位数字。当小时为个位数时，没有前导的零，即流逝时间 5 小时 30 分表示为 530，流逝时间 8 分钟输出为 8。（注：默认用户输入的终止时间会大于等于起始时间，且格式正确）（6 分）

姓名

学号

级

专业

学院

密

封

线

2. 自行车（Bike）和汽车（Car）都是一种交通工具（Vehicle），不同交通工具重量不同，能够一次行驶的里程也不同。请完善如下程序，使程序能够正确运行，并得到正确的输出。（12 分）

```
abstract class Vehicle {
    private int weights; // 交通工具重量，单位：千克
    public Vehicle(int weights) {
        this.weights = weights;
    }

    public int getWeights() {
        return this.weights;
    }
    //定义抽象方法 distance()，能够被子类使用（2 分）
```

```
class Bike extends Vehicle {
    //写出类 Bike 的构造器，能够被 TestW2 类的 main（）方法正确使用（2 分）
```

```
    public int distance() { //自行车能够一次行驶 50 公里
        return 50;
    }
}
```

```
class Car extends Vehicle {
    private int volume; // 汽油容量，单位：升
    //写出类 Car 的构造器，能够被 TestW2 类的 main（）方法正确使用（2 分）
```

不公开发布

//写出类 Car 的 distance() 方法，计算行驶里程，算法为 volume 乘以 10（2 分）

```
}

public class TestW2 {

    public static void main(String[] args) {

        //定义 Vehicle 类型的数组 ve_arr（2 分）
```

```
        ve_arr[0] = new Bike(20);
        ve_arr[1] = new Car(1000, 50);
        for(int i = 0; i< ve_arr.length; i++) {

            //通过方法计算，得到正确的控制台输出结果，直接字符串输出不得分（2 分）
```

```
        }
    }
}
```

控制台程序输出结果：

ve_arr[0]的重量为：20 千克，能够行驶：50 公里
ve_arr[1]的重量为：1000 千克，能够行驶：500 公里

姓名

学号

级

专业

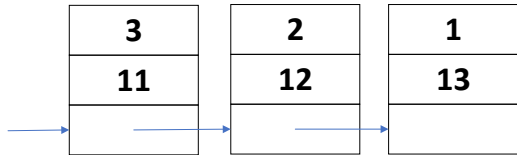
学院

密

封

线

3. 根据一个具有 2 个整数类型作为节点的链表，完成所需要的方法（12 分）。示例如下：



```
class ListNode{ //为方便，属性定义为 public，可以不用 getter 方法直接使用。
    public int data1;
    public int data2;
    public ListNode link;
}

public class List {
    private ListNode head;// 指向首元素
    public int sum() {
        //返回链表中所有 data1 和 data2 元素的数值和。（2 分）（如图例中，返回整数 42）
```

```
    }

    public void deleteItem(int m, int n) {
        //删除链表中“data1==m 并且 data2==n”的 Node（4 分）
        //如图例中，调用 deleteItem（2,12），删除第二个 Node
```

```
}
```

```
public void insertItem(int m, int n, int k) {
    //新建“data1==m, data2==n”的 Node,并插入“data1=k”的第一个 node 后面（3 分）
    //如图例中，调用 insertItem(4, 10, 2)，将在第二个 Node 后面插入一个新的 Node
```

```
    }

    public void exchangeItem() {
        //找出 data1 数值最大和 data2 数值最大的 Node（若不同 Node 中的数值一样大，找出其中
        一个即可），将这两个 Node 在 List 中互换 （3 分）
        //如图例中，第一个 Node 和第三个 Node 互换
```

```
    }
}
```

软件学院 2019 级高级程序设计语言试卷 A

参考答案与评分标准

一. 填空 (30 分, 每题 2 分)

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

二. 阅读题 (40 分)

1.

```

sduds

*sds*

**s**

*sds*

sduds

```

```

2020

2021Java

2020

Kbwb3131

```

3.

```

1  3

2  3

1  2  3

3  2  1

```

```

1  2  3

1  2  4

1  3  4

2  3  4

```

4.

5.

```
Finally in f()
12

Finally in f()
Caught OtherException

Improper score value
Finally in f()
-1

Caught OtherException
```

三. 写程序

1. 参考答案:

```
import java.util.Scanner;

public class First {
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        int start=scan.nextInt();
        int end=scan.nextInt();
        int gap=(end/100*60+end%100)-(start/100*60+start%100);
        System.out.println(gap/60*100+gap%60);
    }
}
```

2. 自行车（Bike）和汽车（Car）都是一种交通工具（Vehicle），不同交通工具重量不同，能够一次行驶的里程也不同。请完善如下程序，使程序能够正确运行，并得到正确的输出。（12分）

```
abstract class Vehicle {
    private int weights; // 交通工具重量，单位：千克
    public Vehicle(int weights) {
        this.weights = weights;
    }

    public int getWeights() {
```

```

        return this.weights;
    }
    //定义抽象方法 distance(), 能够被子类使用 (2 分)
    public abstract int distance();
}

class Bike extends Vehicle {
    //写出类 Bike 的构造器, 能够被 TestW2 类的 main () 方法正确使用 (2 分)
    public Bike(int weights) {
        super(weights);
    }
    public int distance() { //自行车能够一次行驶 50 公里
        return 50;
    }
}

class Car extends Vehicle {
    private int volume; // 汽油容量, 单位: 升
    //写出类 Car 的构造器, 能够被 TestW2 类的 main () 方法正确使用 (2 分)
    public Car(int weights, int volume) {
        super(weights);
        this.volume = volume;
    }
    //写出类 Car 的 distance()方法, 计算行驶里程, 算法为 volume 乘以 10 (2 分)
    public int distance() {
        return volume * 10;
    }
}

public class TestW2 {

    public static void main(String[] args) {
        //定义 Vehicle 类型的数组 ve_arr (2 分)
        Vehicle[] ve_arr = new Vehicle[2];
        ve_arr[0] = new Bike(20);
        ve_arr[1] = new Car(1000, 50);
        for(int i = 0; i<ve_arr.length; i++) {
            //通过方法计算, 得到正确的控制台输出结果, 直接字符串输出不得分 (2 分)
            System.out.println("ve_arr["+i+"]的重量为: "+ve_arr[i].getWeights()+ "千克, 能够行驶"+ve_arr[i].distance()+"公里");
        }
    }
}

```



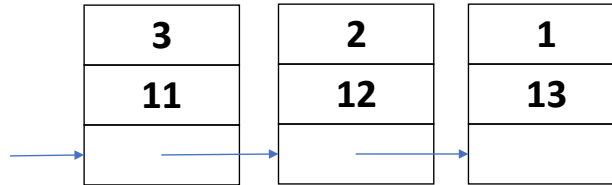
```
}
```

控制台程序输出结果：

ve_arr[0]的重量为：20 千克，能够行驶：50 公里

ve_arr[1]的重量为：1000 千克，能够行驶：500 公里

3. 根据一个具有 2 个整数类型作为元素的链表，完成所需要的方法（12 分）。示例如下，



class ListNode{ //为方便，属性定义为 public，可以不用 getter 方法直接使用。

```
    public int data1;
```

```
    public int data2;
```

```
    public ListNode link;
```

```
}
```

```
public class List {
```

```
    private ListNode head;// 指向首元素
```

```
    public int sum() {
```

```
        //返回链表中所有 data1 和 data2 元素的数值和。
```

```
        //如图例中，返回整数 42
```

```
        int count = 0;
```

```
        ListNode position = head;
```

```
        while (position != null) {
```

```
            count += (position.data1 + position.data2);
```

```
            position = position.link;
```

```
        }
```

```
        return count;
```

```
    }
```

```
    public void deleteItem(int m, int n) {
```

```
        //删除链表中“data1==m 并且 data2==n”的 Node（4 分）
```

```
        //如图例中，调用 deleteItem（2,12），删除第二个 Node
```

```
        ListNode position = head;
```

```
        ListNode current = position;
```

```
        while (position!=null && position.link != null) {
```

```
            current = position;
```

```
            if (position.link.data1 == m && position.link.data2 == n)
```

```
                position.link = position.link.link;
```

```
            else
```

```
                position = position.link;
```

```
        }
```

```
        if(head!= null && head.data1 == m && head.data2 == n)
```

```

        head = head.link;
        if (position!=null && position.data1 == m && position.data2 == n)
            current.link = null;
    }
    public void insertItem(int m, int n, int k) {
        //新建“data1==m, data2==n”的 Node,并插入“data1==k”的第一个 node 后面 (3 分)
        //如图例中, 调用 insertItem(4, 10, 2), 将在第二个 Node 后面插入一个新的 Node
        ListNode position = head;
        while (position != null) {
            if (position.data1 == k) {
                position.link = new ListNode(m, n, position.link); //if 语句 1 分
                break;
            }
            position = position.link;
        }
    }

    public void exchangeItem() {
        //找出 data1 数值最大和 data2 数值最大的 Node (若不同 Node 中的数值一样大,
        找出其中一个即可), 将这两个 Node 在 List 中互换 (3 分)
        //如图例中, 第一个 Node 和第三个 Node 互换
        if(head == null)
            return;
        ListNode Max_data1 = head, Max_data2 = head ;
        ListNode position = head;
        int temp_data1 = position.data1, temp_data2 =position.data2;//初始化 1 分
        while (position != null) {
            if (temp_data1 < position.data1) {
                temp_data1 = position.data1;
                Max_data1 = position;
            }
            if (temp_data2 < position.data2) {
                temp_data2 = position.data2;
                Max_data2 = position;
            }
            position = position.link;
        }
        int temp = Max_data1.data2;
        Max_data1.data1=Max_data2.data1;
        Max_data1.data2=temp_data2;
        Max_data2.data1=temp_data1;
        Max_data2.data2=temp;
    }
}

```