Day 1

1.如果 `int x=20, y=5`，则语句 `System.out.println(x+y + ""+(x+y)+y);` 的输出结果是（ D ）。

A. `2530`　　B. `55`　　C. `2052055`　　D. `25255`

解：

先计算 () 里的：x + y = 25.

从左向右计算：`x + y` 为 `25`，遇到字符串将数字转为字符串，然后将后边的续接到前边：25255. ♢

2.在 Java 中，在同一包内，类 Cat 里面有个公共方法 sleep()，该方法前有 static 修饰，则可以直接用 Cat.sleep()。（ A ）

A. `正确`　　B. `错误`

Tips

用 `static` 修饰的为类方法，类方法可由类名直接调用。

3.给定以下方法声明，调用执行 `mystery(1234)` 的输出结果是（ B ）。

```
1 //precondition: x >= 0
2 public void mystery (int x) {
3     System.out.print(x % 10);
4     if ((x / 10) != 0){
5         mystery(x / 10);
6     }
7     System.out.print(x % 10);
8 }
```

A. `1441`　　B. `43211234`　　C. `3443`　　D. `12344321`

解：

x%10 = 1234 % 10 = 4 ♢

Tips

该方法的第 3~6 行的功能是实现逆序打印。

4.实现函数 `ToLowerCase()`，该函数接收一个字符串参数 `str`，并将该字符串中的大写字母转换成小写字母，之后返回新的字符串。OJ 链接-转换成小写字母

转换成小写字母

```
1 class Solution {
2     public String toLowerCase(String str) {
3         return str.toLowerCase();
4     }
5 }
```

Tips

转大写：`str.toUpperCase()`

转小写：`str.toLowerCase()`

5.给定一个数组，将数组中的元素向右移动 k 个位置，其中 k 是非负数。OJ 链接-旋转数组

旋转数组

```java
class Solution {
    public void rotate(int[] nums, int k) {
        k %= nums.length;
        reverse(nums,0,nums.length-1);
        reverse(nums,0,k-1);
        reverse(nums,k,nums.length-1);

    }
    public void reverse(int[] nums,int lo,int hi){
        while(lo < hi){
            int tmp = nums[lo];
            nums[lo] = nums[hi];
            nums[hi] = tmp;
            lo++;
            hi--;
        }
    }
}
```

Tips

在 C 语言做过这个题。

1. 要先把旋转的位置除数组的长度取余，因为向右移动数组长度个位置（或其倍数）等于没有移动。
2. 把整个数组旋转得到新的数组。
3. 依次旋转数组的前 k 个元素（ 0到k-1 ），和后边剩余的元素。

Day 2

1.如果类的方法没有返回值，该方法的返回类型是（ A ）。

A. `void` B. `null` C. `abstract` D. `default`

2. `Java Application(Java应用程序)` 源程序文件编译后的字节码文件的拓展名是（ B ）。

A. `java` B. `class` C. `exe` D. `jar`

Tips

Java 源文件的拓展名：`java`

Java 源文件编译后拓展名：`class`

3.关于继承的描述正确的是（ C ）。

A. 函数的出口应该尽可能少，最好只有一个出口

B. 为了防止程序中的内存泄漏，应该不使用动态内存分配

C. 在函数实现中应该少使用全局变量

D. 函数的功能应该单一

Tips

A. 函数应为每个逻辑分支提供出口。

B. 动态内存分配无法避免，应该及时回收空间和释放指针避免动态内存泄露。

C. 为保证封装性，尽量少用全局变量。

D. 函数可以根据不同的输入来实现相应的功能（重载）。

4.给定一个数组 `nums` 和一个值 `val`，你需要原地移除所有数值等于 `val` 的元素，返回移除后数组的新长度。不要使用额外的数组空间，你必须在原地修改输入数组并在使用 $O(1)$ 额外空间的条件下完成。OJ 链接-移除元素

Tips

1. 边界情况：数组为空时，要返回 0。

分析：不考虑顺序，返回的为不含 `val` 的前边一段，思路：把 `val` 换到后边去。

2. 左指针 `lo` 指向数组的左边，找 `val`，如果没找到 `val`，就把这个指针向右移动；

3. 右指针 `hi` 指向数组的右边，`val` 就应该位于数组后边，所以如果值是 `val`，指针就左移。

4. 经过 2、3 步后，左指针指的是 `val`，右指针指的是非 `val`，将这俩指针对应元素交换。

5. 左右指针重合时，数组就遍历完了。

6. 左指针的左边一定是不含 `val` 的，其左边有 `lo` 个元素，如果此时的左指针对应元素的值时 `val`，那就返回 `lo`，如果左指针对应元素的值不是 `lo`，就返回 `lo + 1`。

移除元素

```java
class Solution {
    public int removeElement(int[] nums, int val) {
        if(nums.length == 0){
            return 0;
        }
        int lo = 0;
        int hi = nums.length - 1;
        int len = 0;
        while(lo < hi){
            if(nums[lo] != val && lo < hi){
                lo++;
```

```
            }
            if(nums[hi] == val && lo < hi){
                hi--;
            }
            if(lo < hi){
                int tmp = nums[lo];
                nums[lo] = nums[hi];
                nums[hi] = tmp;
            }
        }
        if(nums[lo] == val){
            return lo;
        }
        return lo + 1;
    }
}
```

5.给定一个排序数组和一个目标值，在数组中找到目标值，并返回其索引。如果目标值不存在于数组中，返回它将会被按顺序插入的位置，你可以假设数组中无重复元素。OJ 链接-搜索插入位置

Tips

有序数组，采用二分查找。

1. 找到元素：返回元素的下标。
2. 如果数组中没有该元素。退出循环的前一次左指针、右指针、中间指针指向同一元素，若该元素小于目标值，即 `nums[mid] < target`，左指针指向中间指针的下一个元素，此时左指针在右指针的右边，退出循环。同时左指针指向的位置也是目标元素要插入的位置。
3. 若该元素大于目标值，即 `target < nums[mid]`，右指针指向中间指针的前一个元素，右指针在左指针的左边，退出循环。左指针的左侧元素必定小于目标值，而左指针指向的元素大于目标值，目标值应该插入到左指针指向的位置处。
4. 综上，找到目标值就返回该目标值对应的下标，没找到该目标值就返回左指针指向的位置。

搜索插入位置

```
class Solution {
    public int searchInsert(int[] nums, int target) {
        if(nums.length == 0){
            return 0;
        }
        int lo = 0;
        int hi = nums.length - 1;
        int mid = (lo + hi)/2;
        while(lo <= hi){
            mid = (lo + hi)/2;
            if(nums[mid] < target){
                lo = mid + 1;
            }else if(target < nums[mid]){
                hi = mid - 1;
            }else{
                return mid;
            }
```

```
18            }
19            return lo;
20        }
21    }
```

Day 3

1.以下代码的循环次数是（ D ）。

```
public class Test {
    public static void main(String[] args) {
        int i = 7;
        do {
            System.out.println(--i);
            --i;
        } while (i != 0);
        System.out.println(i);
    }
}
```

A. 0　　B. 1　　C. 7　　D. 无限次

解：

i 的初始值为奇数，循环中每次减 2，退出条件为不等于 0，奇数减 2 永远不会为 0，即死循环。 ◇

2.下列选项中属于面向对象设计方法的主要特征是（ A ）。

A. 继承　　B. 自顶向下　　C. 模块化　　D. 逐步求精

Tips

面向对象最主要的几个特征：封装、继承、多态、抽象等。

3.下面的方法，当输入为 2 的时候返回值是多少（ ）。

```
public static int getValue(int i) {
    int result = 0;
    switch (i) {
        case 1:
            result = result + i;
        case 2:
            result = result + i * 2;
        case 3:
            result = result + i * 3;
    }
    return result;
}
```

A. 0　　B. 2　　C. 4　　D. 10

4.赎金信。OJ 链接-赎金信

赎金信

```
1 class Solution {
2     public boolean canConstruct(String ransomnote, String magazine){
3
4     }
```

```
}
```

5.判断一个整数是否是回文数。回文数是指正序（从左到右）和倒叙（从右向左）读都是一样的整数。OJ链接-回文数

回文数

```
class Solution {
    public boolean isPalindrome(int x){

    }
}
```

Day 4

1.下面有关 `java final` 的基本规则，描述错误的是（ ）。

A. `final` 修饰的类不能被继承

B. `final` 修饰的成员变量只允许赋值一次，且只能在类方法赋值

C. `final` 修饰的局部变量即为常量，只能赋值一次

D. `final` 修饰的方法不允许被子类覆盖

2.选项中哪一行代码可以替换 `//add code here` 而不产生编译错误（ ）。

A. `public abstract void method(int a);`

B. `consInt = constInt + 5;`

C. `public int method;`

D. `public abstract void anotherMethod(){}`

3.第三行将输出什么（ ）。

```
1  public class SwitchTest{
2      public static void main(String[] args) {
3          System.out.println("value=" + switchit(4));
4      }
5      public static int switchit(int x) {
6          int j = 1;
7          switch (x) {
8              case 1:j++;
9              case 2:j++;
10             case 3:j++;
11             case 4:j++;
12             case 5:j++;
13             default:j++;
14         }
15         return j+x;
16     }
17 }
```

A. `value=6;` B. `value=8;` C. `value=3;` D. `value=5;`

4.给定一个仅包含大小写字母和空格 `' '` 的字符串，返回其最后一个单词的长度。如果不存在最后一个单词，请返回 0。OJ 链接-最后一个单词的长度

最后一个单词的长度

```
1  class Solution {
2      public int lengthOfLastWord(String s) {
3
4      }
5  }
```

5.给你两个有序整数数组 `nums1` 和 `nums2`，请你将 `nums2` 合并到 `nums1` 中，使 `nums1` 成为一个有序数组。OJ 链接-合并两个有序数组

合并两个有序数组

```java
class Solution {
    public void merge(int[] nums1, int m, int[] nums2, int n) {

    }
}
```

Day 5

1.下列 Java 程序输出的结果为（ ）。

```
public class Example{
    String string = new String("hello");
    char[] chars = {'a','b'};
    public static void main(String args[]){
        Example ex = new Example();
        ex.change(ex.string, ex.ch);
        System.out.print(ex.str + "and");
        System.out.print(ex.ch);
    }
    public void change(String string,char ch[]){
        string = "test ok";
        ch[0] = 'c';
    }
}
```

A. `hello and ab` B. `hello and cb` C. `hello and a` D. `test ok and ab`

2.*transient* 变量和下面哪一项有关（ ）。

A. `Cloneable` B. `Serializable` C. `Runnable` D. `Comparable`

3.已知有下列 *Test* 类的说明，在该类的 *main* 方法内，则下列那个语句是正确的（ ）。

```
public  class Test {
    private float f = 1.0f;
    int m = 12;
    static int n = 1;
    public static void main (String args[]){
        Test t = new Test();
    }
}
```

A. `t.f;` B. `this.n;` C. `Test.m;` D. `Test.f;`

4.给定一个整数数组，判断是否存在重复元素。如果存在一值在数组中出现至少两次，函数返回 `true` 。如果数组中每个元素都不相同，则返回 `false` 。OJ 链接-存在重复元素

```
1 class Solution {
2     public boolean containsDuplicate(int[] nums) {
3
4     }
5 }
```

5.你的朋友正在使用键盘输入他的名字 *name*。偶尔，在键入字符 c 时，按键可能会被长按，而字符可能被输入 1 次或多次。你将会检查键盘输入的字符 `typed` 。如果它对应的可能是你的朋友的名字（其中一些字符可能被长按），那么就返回 `True` 。OJ 链接-长按键入

```java
class Solution {
    public boolean containsDuplicate(int[] nums) {

    }
}
```

Day 6

1.下面关于 Java 的垃圾回收机制正确的是（ ）。

A. 当调用 `"System.gc()"` 来强制回收时，系统会立即回收垃圾

B. 垃圾回收不能确定具体的回收时间

C. 程序可明确地标识某个局部变量的引用不再被使用

D. 程序可以显式地立即释放对象占有的内存

2.以下会产生信息丢失的类型转换是（ ）。

A. `float a = 10;` B. `int a = (int)8846.0 ;` C. `byte a = 10; int b = a;` D. `double = 100`

3.面向对象方法的多态性指的是（ ）。

A. 一个类可以派生出多个特殊类

B. 一个对象在不同的运行环境中可以有不同的变体

C. 针对一消息，不同的对象可以以适合自身的方式加以响应

D. 一个对象可以是由多个其他对象组合而成的

4.给定一个按非递减顺序排序的整数数组 `A`，返回每个数字的平方组成的新数组，要求新数组也按非递减顺序排序。（注意：非递减顺序即递增，要注意原数组里的负数）OJ 链接-有序数组的平方

有序数组的平方

```java
class Solution {
    public int[] sortedSquares(int[] nums) {

    }
}
```

5.给定一个字符串 S，返回“反转后的”字符串，其中不是字母的字符都保留在原地，而所有字母的位置发生反转。OJ 链接-仅仅反转字母

仅仅反转字母

```java
class Solution {
    public String reverseOnlyLetters(String S) {

    }
}
```

Day 7

1.关于下面程序，哪些描述是正确的（ ）。

```
public class While {
    public void loop() {
        int x = 10;
        while (x) {
            System.out.print("x minus one is " + (x - 1));
            x -= 1;
        }
    }
}
```

A. 行 1 有语法错误　　B. 行 4 有语法错误　　C. 行 5 有语法错误
D. 行 6 有语法错误　　E. 行 2 有语法错误,loop 使关键字　　D. 程序能够正常编译和运行

2.在各自最优条件下，对 N 个数进行排序，算法复杂度最低的是（ ）。

A. 插入排序　　B. 快速排序　　C. 堆排序　　D. 归并排序

3.在 `java` 中，下列对继承的说法，正确的是（ ）。

A. 子类能继承父类所有的成员　　B. 子类继承父类的非私有方法和状态
C. 子类只能继承父类的 `public` 方法和状态　　D. 子类只能继承父类的方法

4.给定一个非负的整数数组 `A`，返回一个数组，在该数组中， `A` 的所有偶数元素之后跟着所有奇数元素。
OJ 链接-按奇偶排序数组

按奇偶排序数组

```
class Solution {
    public int[] sortArrayByParity(int[] A) {

    }
}
```

5.给定一个整数类型的数组 `nums`，请编写一个能够返回数组“中心索引”的方法。

我们是这样定义数组 中心索引 的：数组中心索引的左侧所有元素相加的和等于右侧所有元素相加的和。

如果数组不存在中心索引，那么我们应该返回 -1。如果数组有多个中心索引，那么我们应该返回最靠近左边的那一个。OJ 链接-寻找数组的中心索引

寻找数组的中心索引

```
class Solution {
    public int pivotIndex(int[] nums) {

    }
}
```

Day 8

1.（**多选题**）已知 `boolean result = false`，则下面哪个选项是合法的（ ）。

A. `result = 1` B. `result = true` C. `if (result != 0){ }` D. `if (result){ }`

2.（**多选题**）下面的 `switch` 语句中，`x` 可以是哪些类型的数据（ ）。

```
switch(x){
    default:
    System.out.println("hello");
}
```

A. `long` B. `char` C. `float` D. `byte` E. `double` F. `Object`

3.（**多选题**）关于运行时常量池，下列那个说法是正确的（ ）。

A. 运行时常量池大小受栈区大小的影响

B. 运行时常量池大小受方法区大小的影响

C. 存放了编译时期生成的各种字面量

D. 存放编译时期生成的符号引用

4.给定一个由**整数**组成的 **非空** 数组所表示的非负整数，在该数的基础上加一。最高位数字存放在数组的首位，数组中每个元素只存储单个数字。你可以假设除了整数 0 之外，这个整数不会以零开头。(第一个数字是 9 或者最后一个是 9 要考虑进位) OJ 链接-加一

加一

```
class Solution {
    public int[] plusOne(int[] digits) {

    }
}
```

5.给你一个非空数组，返回此数组中 第三大的数。如果不存在，则返回数组中最大的数。OJ 链接-第三大的数

第三大的数

```
class Solution {
    public int thirdMax(int[] nums) {

    }
}
```

Day 9

1.下列类定义中哪些是合法的抽象类的定义（ ）。

A. `abstract Animal{abstract void growl();}`

B. `class abstract Animal{abstract void growl();}`

C. `abstract class Animal{abstract void growl();}`

D. `abstract class Animal{abstract void growl(){System.out.println( “growl” );};}`

2.在 `java` 中，无论在何处调用，使用静态属性必须以类名做前缀。（ ）

A. 正确 B. 错误

3.哪个关键字可以对对象加互斥锁（ ）。

A. `synchronized` B. `volatile` C. `serialize` D. `static`

4.给定一个整数数组 nums 和一个整数目标值 target，请你在该数组中找出 和为目标值 的那 两个整数，并返回它们的数组下标。你可以假设每种输入只会对应一个答案。但是，数组中同一个元素不能使用两遍。你可以按任意顺序返回答案。要求时间复杂度 $O(n)$, 当然也可以选择使用暴力的 $O(n^2)$ 的解法。OJ 链接-两数之和

两数之和

```
class Solution {
    public int[] twoSum(int[] nums, int target) {

    }
}
```

5.给你两个二进制字符串，返回它们的和（用二进制表示）。输入为 **非空** 字符串且只包含数字 `1` 和 `0`。OJ 链接-二进制求和

二进制求和

```
class Solution {
    public String addBinary(String a, String b) {

    }
}
```

Day 10

1.一个 Java 源程序文件中定义几个类和接口,则编译该文件后生成几个以.class 为后缀的字节码文件。（ ）

A. 正确 B. 错误

2.要使某个类能被同一个包中的其他类访问，但不能被这个包以外的类访问，可以（ ）。

A. 让该类不使用任何关键字

B. 使用 `private` 关键字

C. 使用 `protected` 关键字

D. 使用 `void` 关键字

3.判断对错。在 java 的多态调用中，new 的是哪一个类就是调用的哪个类的方法。（ ）

A. 正确 B. 错误

4.字符串转换函数OJ 链接-字符串转换整数 atoi。

请你来实现一个 `myAtoi(string s)` 函数，使其能将字符串转换成一个 32 位有符号整数（类似 `C/C++` 中的 `atoi` 函数）。

函数 `myAtoi(string s)` 的算法如下：

- 读入字符串并丢弃无用的前导空格。
- 检查第一个字符（假设还未到字符末尾）为正还是负号，读取该字符（如果有）。确定最终结果是负数还是正数。如果两者都不存在，则假定结果为正。
- 读入下一个字符，直到到达下一个非数字字符或到达输入的结尾。字符串的其余部分将被忽略。
- 将前面步骤读入的这些数字转换为整数（即，"123" → 123，"0032" →32）。如果没有读入数字，则整数为 0 。必要时更改符号（从步骤 2 开始）。
- 如果整数数超过 32 位有符号整数范围 $[-2^{31},\ 2^{31}\ -1]$ ，需要截断这个整数，使其保持在这个范围内。具体来说，小于 -2^{31} 的整数应该被固定为 -2^{31} ，大于 $2^{31}\ -1$ 的整数应该被固定为 $2^{31}\ -1$。
- 返回整数作为最终结果。

注意：

- 本题中的空白字符只包括空格字符 `' '` 。
- 除前导空格或数字后的其余字符串外，请勿忽略 任何其他字符。

字符串转换整数 atoi

```
class Solution {
    public int myAtoi(String s) {

    }
}
```

5.给定一个按照升序排列的整数数组 `nums` ，和一个目标值 `target` 。找出给定目标值在数组中的开始位置和结束位置。如果数组中不存在目标值 `target` ，返回 $[-1,-1]$。OJ 链接-在排序数组中查找元素的第一个和最后一个位置。

在排序数组中查找元素的第一个和最后一个位置

```
class Solution {
```

```
2       public int[] searchRange(int[] nums, int target) {
3
4       }
5   }
```

Day 11

1.执行下列代码的输出结果是（ ）。

```java
public class Demo{
    public static void main(String args[]){
        int num = 10;
        System.out.println(test(num));
    }
    public static int test(int b){
        try {
            b += 10;
            return b;
        }catch(RuntimeException e){

        }catch(Exception e2){

        }finally{
            b += 10;
            return b;
        }
    }
}
```

A. 10 B. 20 C. 30 D. 40

2.下面关于 Java package 的描述，哪个是正确的（ ）。

I. 包不提供将所有类名分区为更易管理的块的机制。

II. 包提供可见性控制机制。

III. 包的一个重要属性是包内定义的所有类都可以通过该包外的代码访问。

IV. 声明为包的一部分的类的.class 文件可以存储在多个目录中.

A. 只有 I B. 只有 II C. 只有 III D. 只有 IV

3.Java 数据库连接库 JDBC 用到哪种设计模式（ ）。

A. 生成器 B. 桥接模式 C. 抽象工厂 D. 单例模式

4.给定一个字符串，验证它是否是回文串，只考虑字母和数字字符，可以忽略字母的大小写。

说明：本题中，我们将空字符串定义为有效的回文串。

注意：如果测试用例出现错误：OP 这个情况一定注意，这是 OP。OJ 链接-验证回文串

验证回文串

```java
class Solution {
    public boolean isPalindrome(String s) {

    }
}
```

5.给定一组字符，使用原地算法将其压缩。压缩后的长度必须始终小于或等于原数组长度。数组的每个元

素应该是长度为 1 的字符（不是 `int` 整数类型）。在完成原地修改输入数组后，返回数组的新长度。OJ 链接-压缩字符串

压缩字符串串

```
class Solution {
    public int compress(char[] chars) {

    }
}
```

Day 12

1.下列哪个选项是正确计算 42°（角度）的余弦值（　　）。

A. `double d=Math.cos(42)`

B. `double d=Math.cosine(42)`

C. `double d=Math.cos(Math.toRadians(42))`

D `double d=Math.cos(Math.toDegrees(42))`

2. `String s = new String("xyz")` 创建了几个 `String` 对象（　　）。

A. 两个或一个都有可能　　B. 两个　　C. 一个　　D. 三个

3.下列代码输出结果为（　　）。

A.

动物可以移动

狗可以跑和走

狗可以吠叫

B. 动物可以移动 动物可以移动 狗可以吠叫

C. 运行错误

D. 编译错误

4.给你一个整数数组 `nums` ，你需要找出一个**连续子数组**，如果对这个子数组进行升序排序，那么整个数组都会变为升序排序。请你找出符合题意的 最短 子数组，并输出它的长度。OJ 链接-最短无序连续子数组

最短无序连续子数组

```
class Solution {
    public int findUnsortedSubarray(int[] nums) {

    }
}
```

5.根据**逆波兰表示法**，求表达式的值。有效的运算符包括 `+` , `-` , `*` , `/` 。每个运算对象可以是整数，也可以是另一个逆波兰表达式。OJ 链接-逆波兰表达式求值

说明：

- 整数除法只保留整数部分。
- 给定逆波兰表达式总是有效的。换句话说，表达式总会得出有效数值且不存在除数为 0 的情况。

逆波兰表达式求值

```
class Solution {
    public int evalRPN(String[] tokens) {

    }
}
```

Day 13

1.（**多选题**）关于 `Java` 以下描述正确的有（　　）。

A. `Class` 类是 `Object` 类的超类　　B. `Object` 类是一个 `final` 类

C. `String` 类是一个 `final` 类　　D. `Class` 类可以装载其它类

2.（**多选题**） `Java` 中的集合类包括 `ArrayList` 、 `LinkedList` 、 `HashMap` 等类，下列关于集合类描述正确的是（　　）。

A. `ArrayList` 和 `LinkedList` 均实现了 `List` 接口

B. `ArrayList` 的访问速度比 `LinkedList` 快

C. 添加和删除元素时， `ArrayList` 的表现更佳

D. `HashMap` 实现 `Map` 接口，它允许任何类型的键和值对象，并允许将 `null` 用作键或值

3.（**多选题**）关于 `equals` 和 `hashCode` 描述正确的是　（　　）。

A. 两个 `obj` ，如果 `equals()` 相等， `hashCode()` 一定相等（符合代码规范的情况下）

B. 两个 `obj` ，如果 `hashCode()` 相等， `equals()` 不一定相等

C. 两个不同的 `obj` ， `hashCode()` 可能相等

D. 其他都不对

4.给你一个整数数组 nums ，数组中的元素互不相同。返回该数组所有可能的子集（幂集）。解集不能包含重复的子集。你可以按任意顺序返回解集。OJ 链接-子集

子集

```
class Solution {
    public List<List<Integer>> subsets(int[] nums) {

    }
}
```

5.给定一个整数矩阵，找出最长递增路径的长度。对于每个单元格，你可以往上，下，左，右四个方向移动。你不能在对角线方向上移动或移动到边界外（即不允许环绕）。OJ 链接-矩阵中最长递增路径

矩阵中最长递增路径

```
class Solution {
    public int longestIncreasingPath(int[][] matrix) {

    }
}
```

Day 14

1.（**多选题**）下面有关 `java` 的 `instanceof` 、 `?` 、 `&` 、 `&&` 说法正确的有（　　）。

A. `instanceof` 可用来判断某个实例变量是否属于某种类的类型。

B. `"? :"` 三目运算符

C. `&` 在逻辑运算中是非短路逻辑与，在位运算中是按位与

D. `&&` 逻辑运算：逻辑与

2.（**多选题**）下面哪个语句是创建数组的正确语句？（　　）。

A. `float f[][] = new float[6][6];`

B. `float []f[] = new float[6][6];`

C. `float f[][] = new float[][6];`

D. `float [][]f = new float[6][6];`

E. `float [][]f = new float[6][];`

3.下列类在多重 `catch` 中同时出现时，哪一个异常类应最后一个列出（　　）。

A. `ArithmeticException` B. `NumberFormatException` C. `Exception` D. `ArrayIndexOutOfBoundException`

4.给定一棵二叉树，想象自己站在它的右侧，按照从顶部到底部的顺序，返回从右侧所能看到的节点值。

OJ 链接-二叉树的右视图

二叉树的右视图

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
class Solution {
    public List<Integer> rightSideView(TreeNode root) {

    }
}
```

5.公交路线。OJ 链接-公交路线

我们有一系列公交路线。每一条路线 `routes[i]` 上都有一辆公交车在上面循环行驶。例如，有一条路线 `routes[0] = [1, 5, 7]` ，表示第一辆 (下标为 0) 公交车会一直按照 $1 \to 5 \to 7 \to 1 \to 5 \to 7 \to 1 \to \ldots$ 的车站路线行驶。

假设我们从 S 车站开始（初始时不在公交车上），要去往 T 站。期间仅可乘坐公交车，求出最少乘坐的公交车数量。返回 1– 表示不可能到达终点车站。

公交路线

```
class Solution {
    public int numBusesToDestination(int[][] routes, int S, int T) {

    }
}
```

6.通配符匹配。OJ 链接-通配符匹配

给定一个字符串 (s) 和一个字符模式 (p)，实现一个支持 `'?'` 和 `'*'` 的通配符匹配。

- `'?'` 可以匹配任何单个字符。
- `'*'` 可以匹配任意字符串（包括空字符串）。

两个字符串完全匹配才算匹配成功。

说明:

- s 可能为空，且只包含从 `a-z` 的小写字母。
- s 可能为空，且只包含从 `a-z` 的小写字母，以及字符 `?` 和 `*`。

通配符匹配

```
class Solution {
    public boolean isMatch(String s, String p) {

    }
}
```