北 京 交 通 大 学 考 试 试 题（A 卷）

课程名称：Java 语言程序设计　学年学期：2019—2020 学年第二学期

课程编号：80L167Q　开课学院：计算机与信息技术学院　出题教师：李强

学生姓名：_____　学号：_____　任课教师：_____

学生学院：_____　班级：_____

题　号	一	二	四	五	六	总分
得　分						
阅卷人						

一、单选题（25 小题，每题 2 分，共 50 分）

1. Java 的基本数据类型中，（ ）。

【A】 包含 1 种浮点类型，即 float

【B】 包含 1 种浮点类型，即 double

【C】 包含 2 种浮点类型，即 float 和 double

【D】 包含 2 种浮点类型，即 single 和 double

2. Java 中声明一个 int 型二维数组 xA，其第一下标范围是 0 到 9，第二下标范围是 0 到 3，正确的语句是（ ）。

【A】 int xA=new int[9][3];

【B】 int xA=new int[10][4];

【C】 int xA[][]=new int[10][4];

【D】 int xA[][]=new int[9][3];

3. 三个 Java 变量 a、b 和 c，其类型分别为 char、short 和 byte，表达式 a+b+c 的结果类型为（ ）。

【A】 byte

【B】 short

【C】 int

【D】 long

4. 执行语句"a=b=3;System.out.println((a<<2)+","+(b>>2));"后的输出结果为（ ）。

【A】 12,0

【B】 12,3

【C】 true

【D】 false

5. Java 中的 Scanner 类在用于扫描键盘输入带有空格字符的字符串时应使用（ ）。

【A】 nextInt()方法

【B】 next()方法

【C】 Next()方法

【D】 nextLine()方法

6. 使用 for 语句构成一个由 int 类型循环变量 iX 控制的循环结构，首次循环 iX 为 0，每次循环 iX 加 3，最后一次循环 iX 为 30，正确的 for 语句写法是（ ）。

【A】 for(iX=0; iX<=30; iX+=3) {.....}

【B】 for(iX=0; iX<=30; iX+3) {.....}

【C】 for(iX=0; iX<=30; iX++3) {.....}

【D】 for(iX=0; iX<30; iX++3) {.....}

7. Java 的类的成员变量可以分为静态变量和非静态变量，（ ）。

【A】 静态变量即实例变量

【B】 静态方法只能访问静态成员

【C】 非静态变量即类变量

【D】 非静态变量在声明时有 final 修饰符

8. 如果一个 Java 类的成员变量的说明中有 final 修饰符，说明该成员变量（ ）。

【A】 是类变量

【B】 是实例变量

【C】 可以被修改

【D】 不能被修改

9. 关于 Java 的类的继承关系，正确的是（ ）。

【A】 子类拥有父类所有的属性、方法

【B】 子类不可以拥有自己的属性和方法

【C】 子类可以用自己的方式实现父类的方法

【D】 Java 支持多继承

10. 如果在一个类中的一个方法声明中有 abstract 修饰符，则这个类（ ）。

【A】 可以直接实例化对象

【B】 必须含有子类

【C】 private 和 abstract 能同时使用

【D】 abstract 可以用 `final` 修饰符修饰

11. 下列哪个选项是代码执行的结果 （）。

```
String s1 = "Java";
String s2 = "Java";
StringBuilder sb1 = new StringBuilder();
sb1.append("Ja").append("va");
System.out.println(s1 == s2);
System.out.println(s1.equals(s2));
System.out.println(sb1.toString() == s1);
System.out.println(sb1.toString().equals(s1));
```

【A】 false, true, false, true

【B】 true, true, false, true

【C】 true, false, true, true

【D】 false, true, true, true

12. 下列哪个选项的执行结果是正确的 （）。

```
interface HasTail { int getTailLength(); }

abstract class Puma implements HasTail {
    protected int getTailLength() {return 4;}
}

public class Cougar extends Puma {
    public static void main(String[] args) {
        Puma puma = new Puma();
        System.out.println(puma.getTailLength());
    }
    public int getTailLength(int length) {return 2;}
}
```

【A】 2

【B】 4

【C】 在这一行代码 Puma puma = new Puma();出现编译错误

【D】 在这一行代码 System.out.println(puma.getTailLength());出现编译错误

13. 下列哪个选项是代码的运行结果 （）。

```
boolean keepGoing = true;
int count = 0;
int x = 3;
while(count++ < 3) {
    int y = (1 + 2 * count) % 3;
    switch(y) {
        default:
        case 0: x -= 1; break;
        case 1: x += 5;
    }
}
System.out.println(x);
```

【A】 4
【B】 5
【C】 6
【D】 7
【E】 编译错误

14. 下列哪个选项是代码的运行结果（ ）。

```
System.out.print("a");
try {
    System.out.print("b");
    throw new IllegalArgumentException();
} catch (RuntimeException e) {
    System.out.print("c");
} finally {
    System.out.print("d");
}
System.out.print("e");
```

【A】 abe
【B】 abce
【C】 abcde
【D】 abde
【E】 编译错误
【F】 运行时异常

15. 运行代码，下列哪个选项是正确的（ ）。

```
public static void addToInt(int x, int amountToAdd) {
    x = x + amountToAdd;
}
public static void main(String[] args) {
// write your code here
    int a = 15;
    int b = 10;
    addToInt(a, b);
    System.out.println(a);
}
```

【A】 10
【B】 25
【C】 15
【D】 编译错误
【E】 以上都不正确

16. 运行代码，下列哪个选项是正确的（ ）。

```
int[] array = {6,9,8};
List<Integer> list = new ArrayList<>();
list.add(array[0]);
list.add(array[2]);
list.set(1, array[1]);
list.remove(0);
System.out.println(list);
```

【A】 [8]
【B】 [9]
【C】 抛出异常
【D】 编译错误
【E】 打印出类似于 [I@58ceff1

17. 下面哪些是代码运行的结果（ ）。

```
1: public class Egret {
2: private String color;
3: public Egret() {
4: this("white");
5: }
6: public Egret(String color) {
7: color = color;
8: }
9: public static void main(String[] args) {
10: Egret e = new Egret();
11: System.out.println("Color:" + e.color);
12: }
13: }
```

【A】 在第 4 行代码出现编译错误
【B】 Color:null
【C】 Color:white
【D】 在第 10 行出现编译错误
【D】 Color:

18. 下面哪些是代码运行的结果，（ ）。

```
public class Main {
    private int x = 4;
    public static void main(String[] args) {
        int x = 6;
        new Main().new Cell().slam();
    }
    class Cell {
        void slam() { System.out.println("throw away key " + x); }
    }
}
```

【A】 输出 throw away key 4
【B】 输出 throw away key 6
【C】 编译错误
【D】 运行时异常

19. 运行代码 java Main Draumur；以下哪个结果是正确的（ ）。

```
public class Main {
    public static void main(String[] args) {
        Map<Friends, String> hm = new HashMap<Friends, String>();
        hm.put(new Friends("Charis"), "Summer 2009");
        hm.put(new Friends("Draumur"), "Spring 2002");
        Friends f = new Friends(args[0]);
        System.out.println(hm.get(f));
    }
    static class Friends {
        String name;
        Friends(String n) { name = n; }
    }
}
```

【A】 null
【B】 编译错误
【C】 Spring 2002
【D】 运行时异常

20. 运行代码；以下哪个结果是正确的（ ）。

```
public class Main {
    public static void main(String[] args) {
        List<String> x = new ArrayList<String>();
        x.add("3"); x.add("7"); x.add("5");
        List<String> y = new Main().doStuff(x);
        y.add("1");
        System.out.println(x);
    }
    List<String> doStuff(List<String> z) {
        z.add("9");
        return z;
    }
}
```

【A】 [3, 7, 5]
【B】 [3, 7, 5, 9]
【C】 [3, 7, 5, 9, 1]
【D】 运行时异常
【D】 编译错误

21. 运行代码；以下哪个结果是正确的（ ）。

```
public class Dec26 {
    public static void main(String[] args) {
        short a1 = 6;
        new Dec26().go(a1);
        new Dec26().go(new Integer(7));
    }
    void go(Short x) { System.out.print("S "); }
    void go(Long x) { System.out.print("L "); }
    void go(int x) { System.out.print("i "); }
    void go(Number n) { System.out.print("N "); }
}
```

【A】 i L
【B】 i N
【C】 S L
【D】 S N
【E】 编译错误
【F】 运行时异常

22. 目录结构如下。

```
test -|
      | - Finder.class
      | - testdir -|
                   | - subdir
                   | - subdir2
                   | - testfile.txt
```

其中，test, testdir, subdir, subdir2 都是文件夹，Finder.class 和 testfile.txt 是文件。

```
import java.io.*;
public class Finder {
    public static void main(String[] args) throws IOException {
        String[] files = new String[100];
        File dir = new File(args[0]);
        files = dir.list();
        System.out.println(files.length);
} }
```

运行以下代码，执行 java Finder testdir，以下哪个结果是正确的（ ）。

【A】 1
【B】 2
【C】 3
【D】 4
【D】 5

23. 以下哪个结果是正确的（ ）。

```
2  int [][] ia2;
3  int [] ia1 = {1,2,3};
4  Object o = ia1;
5  ia2 = new int[3][3];
6  ia2[0] = (int[])o;
7  ia2[0][0] = (int)o;
```

【A】 编译错误，由于第 4 行代码
【B】 编译错误，由于第 7 行代码
【C】 运行异常，由于第 7 行代码
【D】 运行异常，由于第 5 行代码

24. 运行代码，以下哪个结果是正确的（ ）。

```
public class Tail {}
public class Main implements Serializable {
    String name;
    transient int age;
    Tail tail;
    public static void main(String[] args) throws IOException, ClassNotFoundException {
        try(OutputStream is = new ObjectOutputStream(
                new BufferedOutputStream(new FileOutputStream("birds.dat")))) {
            Main bird=new Main();
            ((ObjectOutputStream) is).writeObject(bird);
            System.out.println(bird);
        }
    }
}
```

【A】 编译正确，正常运行
【B】 编译错误，由于错误代码 ((ObjectOutputStream) is).writeObject(bird);
【C】 编译正确，运行时抛出异常
【D】 编译正确，运行时抛出文件不存在的错误

25. 运行代码，以下哪个结果是正确的（ ）。

```
public class Main  {
        private int x = 10;
         class B {
             private int x = 20;
             class C {
                 private int x = 30;
                 public void allTheX() {
                     System. out .println(Main.this.x);
         }} }
         public static void main(String[] args) {
             Main.B.C c = new Main().new B().new C();
             c.allTheX();
}}
```

【A】 编译正确，运行时异常

【B】 输出 10

【C】 输出 20

【D】 编译错误

二、多选题（5 小题，每题 2 分，共 10 分）

1．下列哪个选项是正确的（ ）。

```
1: import java.util.*;
2: public class Grasshopper {
3: public Grasshopper(String n) {
4: name = n;
5: }
6: public static void main(String[] args) {
7: Grasshopper one = new Grasshopper("g1");
8: Grasshopper two = new Grasshopper("g2");
9: one = two;
10: two = null;
11: one = null;
12: }
13: private String name; }
```

【A】 第 9 行代码执行完后，内存堆空间不存在 Grasshopper 对象

【B】 第 10 行代码执行完后，内存堆空间不存在 Grasshopper 对象

【C】 第 11 行代码执行完后，内存堆空间不存在 Grasshopper 对象

【D】 第 9 行代码执行完后，内存堆空间只存在一个 Grasshopper 对象

【E】 第 10 行代码执行完后，内存堆空间只存在一个 Grasshopper 对象

【F】 编译错误

2．下面哪些是运行时异常（runtime exceptions） （ ）。

【A】 Exception
【B】 IllegalArgumentException
【C】 IOException
【D】 NullPointerException
【E】 NumberFormatException
【F】 StackOverflowError

3. 下面哪些选项是正确的（ ）。

```
public class Main extends Thread {
    public static void main(String[] args) {
        Thread t = new Thread(new Main());
        Thread t2 = new Thread(new Main());
        t.start();
        t2.start();
    }
    public void run() {
        for(int i = 0; i < 2; i++)
            System.out.print(Thread.currentThread().getName() + " ");
    }
}
```

【A】 编译错误
【B】 输出是：Thread-1 Thread-3 Thread-1 Thread-3
【C】 输出是：Thread-3 Thread-1 Thread-1 Thread-3
【D】 输出是：Thread-3 Thread-1 Thread-1 Thread-1
【E】 输出是：Thread-1 Thread-2 Thread-1 Thread-2

4. 运行代码；以下哪个结果是正确的（ ）。

```
public class Main {
    public static void main(String[] args) {
        try {
            if(args.length == 0) throw new Exception();
        }
        catch (Exception e) {
            System.out.print("done ");
        }
        finally {
            System.out.println("finally ");
        } }}
```

【A】 "done "
【B】 "finally "
【C】 "done finally "
【D】 运行时异常
【E】 编译错误

5. 下列哪行代码，填到下划线空白处，编译和运行正常（ ）。

```
Set<? extends RuntimeException> set = _____
```

【A】 new HashSet<? extends RuntimeException>();
【B】 for(iX=0; iX<=10; iX+2) {.....}
【C】 new TreeSet<RuntimeException>();
【D】 new TreeSet<NullPointerException>();
【E】 以上都不正确

四、(10 分)当前目录存在一个文件“text.dat”，读取文件内容，并输出每一个单词出现的次数。

编写程序，满足以下要求：

（1） 写一个函数 String [] readFileString(File f)，读取文件内容，每个单词存储在字符串数组中；提示：利用标点符号和空格进行分割，得到单词；

（2） 写一个函数 void createMap(Map<String, Integer> map)，统计每一个单词出现的次数，单词不允许重复；

（3） 写一个函数 void displayMap(Map<String, Integer> map)，输出每一个单词出现的次数

（4） 在 Main 函数中测试以上函数功能。

示例：文件“text.dat”的内容（Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses.）
输出：Java 3; was 1; originally 2; developed 1; by 3… …此处省略

五、(15 分) 当前目录存在一个文件“ball.dat”，文件存储了弹跳球信息，存储了弹跳的参数（double x, double y, double speedX, double speedY, String color）。

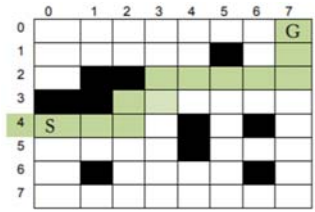
编写程序，满足以下要求：

（1） 读取文件“ball.dat”

（2） 无需考虑碰撞

（3） 生成 1 个弹跳球 JAVAFX 可视化展示弹跳球运动

六、（15 分）地图寻路问题，随机初始化一个迷宫地图（无需可视化），从源节点出发，找到一条路径，达到目的地，并输出该路径（不需要考虑最优路径）。



示例：如上图所示，其中白色的格子代表可以通行的路径，黑色的格子代表障碍物（无法通过），S 表示初始位置，G 表示目的地位置，彩色部分表示可输出的 S 到 G 的可行的路径中的一条。其中 S 的坐标是(4, 0), G 的坐标是(0,7), 路径 S 到 G 是：{(4,0), (4,1), (4,2), (3,2), (3,3), (2,3),(2,4), (2,5), (2,6), (2,7), (1,7), (0,7)}。

编写程序，满足以下要求：

（1）编写地图类 GridMap，要求满足以下要求；

int width; int length; int [] [] gridMap;	地图类的属性：长度和宽度，二维数组 gridMap 表示地图中每一个格子，0 表示该格子无障碍物，1 表示该格子有障碍物；
GridMap (int w, int l, int [] [] gm)	构造函数，初始化程度和宽度，初始化 gridMap 地图对象

（2）编写一个格子类 Cell，要求满足以下要求；

int x; int y;	格子类的属性，x 坐标和 y 坐标
Cell (int x, int y)	构造函数
double getDistance(Cell C)	计算两个格子间的距离
boolean isNeighbor(Cell C)	判断两个格子是否是邻居，是邻居返回 true，否则返回 false；
boolean isFree (GridMap m)	判断该格子是否有障碍物，具有障碍物，返回 fasle，没有，返回 true
int compareTo(Cell C)	比较两个格子，如果坐标相等，返回 0；如果当前 x 轴坐标小于格子 C 的 x 轴坐标，返回 1；否则，返回-1.

（3）编写路径类 Path，要求满足以下要求；

List<Cell> path	path 表示地图中的路径；
path ()	构造函数， 为 path 变量初始化 ArrayList 对象
void addCell (Cell C)	在 path 中，增加一个满足路径要求的格子（无障碍物）
void removeCell (Cell C)	删除的路径 path 上的格子

（4）编写 Main 函数进行测试和验证，从源节点出发，找到一条路径，达到目的地，并输出该路径。