

# 2019A

## 一、简答题 (4 × 5 = 20分)

1. 说明类、对象及对象引用的概念，并举例。

eg: `Student s = new Student();`

Student是类，new出来的是对象，s是对象引用

2. 简述Java的多态性，以及运行时多态的三个必要条件。

Java的多态性

必要条件：

1. 继承和方法重写
2. 向上转型
3. 动态绑定

3. Java中main方法的定义为`public static void main(String[] args)`，请分别说明main方法的修饰符和参数的含义。

修饰符：

参数：args是String类型的数组，代表命令行参数，在使用Java命令时可以用于向程序提供命令行参数。

4. 简述Java异常的分类和继承关系，并写出3种常见异常类型的名称及其功能。

画图或语言描述

NullPointerException、IndexOutOfBoundsException、

## 二、读程题 (每空2分，共40分)

给出下列Java程序代码片段的运行结果

```
1. 1 public class Test1 {
2     public static void main(String[] args) {
3         int i,s = 0;
4         int a[] = {1,2,3,4,5,6,7,8,9};
5         for (i = 0; i < a.length; i++) {
6             if (a[i] % 2 == 0) {
7                 s += a[i];
8                 System.out.println("s =" + s);
9             }
10        }
11    }
12 }
```

程序输出结果：

```
1 s = 2
2 s = 6
3 s = 12
4 s = 20
```

2. 1 public class Test2 {

```

2      static int x = 1;
3      int y;
4      Test2() { y++; }
5      static { x++; }
6      public static void main(String[] args) {
7          Test2 st = new Test2();
8          System.out.println("x =" + x);
9          System.out.println("st.y =" + st.y);
10         st = new Test2();
11         System.out.println("x =" + x);
12         System.out.println("st.y =" + st.y);
13     }
14 }

```

程序输出结果:

```

1  x = 2
2  st.y = 1
3  x = 2
4  st.y = 1

```

```

3. 1  class Animal {
2      String name;
3      Animal(String name) {
4          this.name = name;
5          System.out.println("Animal " + name);
6      }
7      void eat() {
8          System.out.println("Animal eat");
9      }
10 }
11
12 public class Cat extends Animal {
13     int id;
14     Cat(int id, String name) {
15         super(name); this.id = id;
16         System.out.println("Cat " + name);
17     }
18     void eat() {
19         System.out.println("Cat name: " + name);
20         System.out.println("Cat " + id + " eat");
21     }
22     public static void main(String[] args) {
23         Animal a = new Cat(1, "mimi");
24         a.eat();
25     }
26 }

```

程序输出结果:

```

1  Animal mimi
2  Cat mimi
3  Cat name: mimi
4  Cat 1 eat

```

4.

```
1 public class Test4 {
2     public static void main(String[] args) {
3         Set<String> s = new HashSet<String>();
4         for (String a : args) {
5             if(!s.add(a)) {
6                 System.out.println("Duplicate detected: " + a);
7             }
8         }
9         System.out.println(s.size() + " distinct words detected: " + s);
10    }
11 }
12 // 如果在windows命令窗口中输入下列命令:
13 // java Test4 I come I see I go
```

程序输出结果

```
1 Duplicate detected: I
2             come
3             see
4             go
5 4 distinct words detected:
```

5.

```
1 public class Test5 {
2     public static void main(String[] args) {
3         int a[] = {5,9,6,8,7};
4         for (int i = 0; i < a.length - 1; i++) {
5             int k = i;
6             for (int j = i; j < a.length; j++) {
7                 if (a[j] > a[k]) k = j;
8                 int temp = a[i];
9                 a[i] = a[k];
10                a[k] = temp;
11            }
12        }
13        for (int i = 0; i < a.length; i++) {
14            System.out.println(a[i]);
15        }
16    }
17 }
```

程序输出结果:

```
1
2
3
4
```

### 三、改错题 (共10分)

下面Java程序中存在5处编译错误, 指出错误位置 (行号), 说明错误原因并改正。

```
1 class ValueException implements Exception {
2     String info;
```

```

3     public ValueException(String info) {
4         this.info = info;
5     }
6     public void testValue(int value) {
7         if (value > 0)
8             System.out.println("Correct Value");
9         else
10            throw valueException("value must be greater than 0");
11    }
12 }
13 public class TestResult {
14     public static void main(String[] args) {
15         try {
16             ValueException ve = new ValueException();
17             ve.testValue(1);
18             ve.testValue(0);
19         } catch (ValueException e) {
20             System.out.println("产生异常: " + e.info);
21         } final {
22             System.out.println("释放资源");
23         }
24     }
25 }

```

1. 第1行, implements, Exception为类, 应该是继承关系, 应该为extends
2. 第6行, 有异常的方法应该throws该异常
3. 第10行, 异常应该new出来再抛出
4. 第16行
5. 第21行

#### 四、编程题 (共28分)

1. 按照以下要求编写程序。(15分)

需求: 根据要求补充完成4处内容方法。

```

1  interface Flyable {
2      public void fly();
3  }
4  interface Runner {
5      public void run();
6  }
7  abstract class Animal {
8      protected int id;
9      double weight;
10     public Animal(int id, double weight) {
11         this.id = id; this.weight = weight;
12     }
13     abstract public void eat();
14 }
15
16 class Bird extends Animal implements Flyable {
17     public Bird(int id, double weight) {
18         super(id, weight);

```

```

19     }
20     public void eat() { // 每次喂养体重增加1
21         weight += 1;
22     }
23     public void fly() {
24         System.out.println("Bird " + id + "flys");
25     }
26 }
27
28 class Cat extends Animal implements Runner {
29     String name;
30     public Cat(int id, String name, double weight) {
31         // 填充1:分别实现三个参数的传递赋值，完成对象初始化（2分）
32
33         super(id, weight);
34         this.name = name;
35     }
36
37     // 填充2:实现喂养方法（返回值、参数和函数体），每次喂养体重增加20（2分）
38     _____ eat(_____) {
39
40
41     public void eat() {
42         weight += 20;
43     }
44
45     }
46
47     // 填充3:实现接口方法，该方法中需输出猫 id、name 及其他必要描述信息。（3分）
48
49     public void run() {
50         System.out.println("Cat " + id + " " + name + " is running...");
51     }
52
53 }
54
55 public class TestResult {
56     public static void main(String[] args) {
57         Animal[] animals = new Animal[2];
58         animal[0] = new Cat(1, "Mimi", 100);
59         animal[1] = new Bird(2, 20);
60         raiseAllPets(animals);
61         petsEnjoy(animals);
62     }
63
64     private static void raiseAllPets(Animal[] animals) {
65         // 填充4:喂养所有的宠物（4分）
66
67         for (Animal a : animals) {
68             a.eat();
69         }
70
71     }
72
73     private static void petsEnjoy(Animal[] animals) {

```

```

74      // 填充5:使传入参数中的所有animals开心,猫开心则run,鸟开心则fly (4分)
75
76      for (Animal a : animals) {
77          if (a instanceof Cat) {
78              a.run();
79          } else if (a instanceof Bird) {
80              a.fly();
81          }
82      }
83
84  }
85 }

```

2. 编程实现文件拷贝, 将 D:\Hello.txt 拷贝到 C:\HelloWorld.bak。 (13分)

注1: 给定文件的具体名称及路径, 须考虑源文件不存在情况;

注2: 流的打开和关闭。

注3: 可能用到的类及方法如下:

```

1  class FileInputStream extends InputStream;
2  public FileInputStream(String name) throws FileNotFoundException
3  public int read() throws IOException
4  public int read(byte b[]) throws IOException
5  public void close() throws IOException
6  class FileOutputStream extends OutputStream;
7  public FileOutputStream(String name) throws FileNotFoundException
8  public void write(int b) throws IOException
9  public void write(byte b[]) throws IOException
10 public void close() throws IOException

```

// 不用考虑权限, 用可能用到的最简单。

## 2018A

### 一、简答题 (4 × 5 = 20分)

1. 简述什么是构造方法; 构造方法和成员方法的区别; 能否不使用构造函数创建对象, 并说明理由。
2. 简述接口和抽象类的区别。
3. 简述下列关键字的意义和使用场合: final、finalize、finally。
4. 简述运行时异常和非运行时异常的区别。

### 二、读程题 (前4题每空2分、第5题每空1分, 共38分)

1. 注意print和println区别

```

1  class Animal {
2      { System.out.print("感谢大家"); }
3      Animal(int id) {
4          System.out.println();
5          System.out.print("ID:" + id);
6      }
7      Animal() {
8          this(10);

```

```

9      System.out.print("继续加油");
10     }
11
12     static {
13         System.out.print("本学期的努力");
14     }
15 }
16
17 public class Cat extends Animal {
18     Cat() {
19         System.out.println();
20         System.out.print("I am a Cat");
21     }
22
23     public static void main(String[] args) {
24         new Cat();
25     }
26 }

```

程序输出结果：

```

1
2

```

2.

```

1 public class Test2 {
2     public static void main(String[] args) {
3         int a[] = {11,22,33,44,55,66,77,88,99};
4         for (int i = 0; i <= a.length / 2; i++) {
5             System.out.print(a[i] + a[a.length - i - 1] + " ");
6             if (i % 2 == 0) System.out.println();
7         }
8     }
9 }

```

程序输出结果：

```

1
2

```

3.

```

1 class Super {
2     int a, b;
3     Super(int x, int y) { a = x; b = y; }
4 }
5
6 class Sub extends Super {
7     int c;
8     Sub(int xx, in yy, int zz) {
9         super(xx,yy);
10        c = zz;
11    }
12 }
13
14 public class SubSub extends Sub {
15     int a;

```

```

16     SubSub(int aa, int bb, int cc) {
17         super(aa,bb,cc);
18         a = aa+bb+cc;
19     }
20
21     void show() {
22         System.out.println("a=" + a);
23         System.out.println("b=" + b);
24         System.out.println("c=" + c);
25     }
26
27     public static void main(String[] args) {
28         SubSub s = new SubSub(1,2,3);
29         s.show();
30     }
31 }

```

程序输出结果：

```

1
2

```

```

4. 1  public class TestJoin {
2      public static void main(String[] args) {
3          MyThread2 t1 = new MyThread2("a1");
4          t1.start();
5          try {
6              t1.join();
7          } catch (InterruptedException e) { }
8          for (int i = 1; i <= 3; i++) {
9              System.out.println("Main Thread");
10         }
11     }
12 }
13
14 class MyThread2 extends Thread {
15     Mythread(String s) { super(s); }
16
17     public void run() {
18         for (int i = 1; i <= 2; i++) {
19             System.out.println("I am " + getName());
20             try {
21                 sleep(1000);
22             } catch (InterruptedException e) { }
23         }
24     }
25 }

```

程序输出结果：

```

1
2
3
4

```



```
5. 1 public class TryCatchFinally {
    2     public static int func(int i) {
    3         try {
    4             System.out.println("try block");
    5             if (i == 1) return 1;
    6             int res = 1/(i-2);
    7             return res;
    8         } catch (Exception e) {
    9             System.out.println("catch block");
   10             return -2;
   11         } finally {
   12             System.out.println("finally block");
   13         }
   14     }
   15
   16     public static void main(String[] args) {
   17         for (int i = 1; i < 4; i++) {
   18             System.out.println(func(i));
   19         }
   20     }
   21 }
```

程序输出结果：

```
1
2
3
4
5
6
7
8
9
```

### 三、改错题（共12分）

下面Java程序中存在6处编译错误，指出错误位置，说明错误原因并改正。

```
1 abstract class Person {
2     private String name;
3     public String getName() { return name }
4     public abstract boolean isGraduated() throws NameNotExistException {}
5 }
6
7 class NameNotExistException extends RuntimeException {}
8 class Student extends Person {
9     final int studentID;
10    public boolean isGraduated() throws Exception {
11        // 检查学生是否毕业，并返回结果，默认返回为假，此处代码省略
12        return false;
13    }
14
15    public String elective(final String className) {
16        className = "Java 程序设计";
```

```

17         return className;
18     }
19 }
20
21 public class TestResult {
22     public static void main(String [] args) {
23         Student p = new Student();
24         System.out.print(p.getName() + "选修了" + p.elective() + "课程");
25     }
26 }

```

1. 第3行, return name后缺少结束符";"
2. 第10行, Student类重写父类Person类的isGraduated方法, 不能抛出新的异常Exception
3. 第16行, 形参className被final关键字修饰, 不能改变其值
4. 第9行, final变量应该显示初始化
5. 第24行, p.elective()调用方法时缺少参数
6. 第4行, abstract方法没有方法体, 故应删除"{}"

#### 四、编程题

1. 按照以下要求编写程序 (10分)

需求: 根据要求补充完成4处内容方法

```

1  import java.util.ArrayList;
2  import java.util.List;
3
4  abstract class Pet {
5      private String name;
6      double weight;
7      public Pet(String name, double weight) {
8          this.name = name; this.weight = weight;
9      }
10
11     public abstract void raise();
12
13     // 填充1:输出宠物类的名称和重量信息, 并以一定的格式分隔开。(2分)
14     public String getInfo() {
15         return "name = " + this.name + ", weight = " + this.weight;
16     }
17 }
18
19 class Dog extends Pet {
20     public Dog(String name, double weight) {
21         super(name,weight);
22     }
23
24     public void raise() {
25         // 填充2:填写下一行中等号右边部分, 使每次调用该方法小狗体重增加0.2
26         weight = weight + 0.2;
27     }
28 }
29
30 class Cat extends Pet {

```

```

31     public Cat(String name, double weight) {
32         super(name, weight);
33     }
34
35     // 填充3:创建相应方法,并使小猫每次喂养体重增加0.1
36     public void raise() {
37         weight += 0.1;
38     }
39 }
40
41 public class Lady {
42     public static void main(String[] args) {
43         List<Pet> p = new ArrayList<Pet>();
44         p.add(new Dog("DogA", 20));
45         p.add(new Dog("DogB", 30));
46         p.add(new Cat("CatB", 30));
47         raiseAllPets(p);
48     }
49
50     // 填充4:为列表p中的每个宠物喂养一次,并输出宠物的名称和体重
51     public static void raiseAllPets(List<Pet> p) {
52         for (Pet pet : p) {
53             pet.raise();
54             System.out.println(pet.getInfo());
55         }
56     }
57 }

```

## 2. 按照以下要求编写程序 (20分)

要求: 在电脑 C:/ 根目录下创建一个名为 HelloWorld.java 的文件, 并判断其是文件还是目录;  
之后创建 C:/IOTest, 并将 HelloWorld.java 移动到 IOTest 目录下, 并遍历 IOTest 目录下的其它文件及目录 (不含子目录) 并输出。

注1: 使用绝对路径;

注2: 不需要使用递归

## 2020A

### 一、简答题 (5 × 4 = 20分)

#### 1. 阅读下列程序并回答问题

```

1  public class Test {
2      public static void main(String[] args) {
3          System.out.println("this is main");
4      }
5  }
6  class Student {
7      public void func() {
8          System.out.println("this is student");
9      }
10 }

```

1) 请给出该源程序的完整文件名

2) 如何在命令行下编译和运行这个程序 (不考虑路径问题)

- 3) 该源文件编译后产生什么文件
- 4) 该程序执行输出什么结果?
2. 简述Java创建线程的两种方法。
3. 简述节点流、过滤流、字节流、字符流概念。
4. 简述四种Java访问权限定义。
5. 简述接口定义结构。

## 二、读题 (4 × 10 = 40分)

```
1. 1 public class PassTest {
    2     public static void main(String[] args) {
    3         int k = 3;
    4         int pos = 1;
    5         int n = 5;
    6         int num[] = {0,1,2,3,4,5};
    7
    8         while (n > 0) {
    9             pos = (pos + k - 1) % n == 0 ? n : (pos + k - 1) % n;
   10             System.out.println(num[pos]);
   11             for (int i = pos; i < n; i++)
   12                 num[i] = num[i+1];
   13             pos = pos == n ? 1 : pos;
   14             n--;
   15         }
   16     }
   17 }
```

程序输出结果:

1 |

```
2. 1 public class PassTest {
    2     public static void main(String[] args) {
    3         String sentence = "My motherland and I, can not be separated for
    4         a moment";
    5         String word1[] = sentence.split(" ");
    6         String word2[] = sentence.split(",");
    7
    8         System.out.println(sentence.indexOf("and"));
    9         System.out.println(word1.length);
   10         System.out.println(word2[0].substring(4, 9));
   11         System.out.println(word1[word1.length-1].length());
   12         System.out.println(word2[word2.length-1].charAt(10));
   13     }
   14 }
```

程序输出结果:

1 |

```
3. 1 public abstract class T1 {
```

```

2     private int n1 = 20;
3     static int n2;
4     static {
5         n2 = T2.n2 + 10;
6         System.out.println(n2);
7     }
8     void setn1() {
9         n1 += 10;
10    }
11    int getn1() {
12        return n1;
13    }
14    public static void main(String[] args) {
15        T2 t2 = new T2();
16        t2.setn1();
17        t2.getdistance();
18    }
19 }
20
21 class T2 extends T1 {
22     private int n1 = 200;
23     static int n2;
24     static {
25         n2 = T1.n2 + 100;
26         System.out.println(n2);
27     }
28     void getdistance() {
29         System.out.println("startwith:" + getn1());
30     }
31 }

```

程序输出结果:

```

1
4. 1 class IntegerRangeException extends Exception {
2     public IntegerRangeException(String s) { super(s); }
3 }
4
5 public class Demo4 {
6     public static void main(String[] args) {
7         MyInteger m1 = null;
8
9         for (int value = 10; value <= 50; value += 20) {
10            try {
11                m1 = MyInteger.getInstance(value);
12            } catch (Exception e) {
13                System.out.println(e.getMessage());
14            }
15            if (null == m1)
16                System.out.println(m1);
17            else
18                System.out.println(m1.value);
19        }
20    }

```

```

21 }
22
23 class MyInteger {
24     int value;
25     private MyInteger(int value) {
26         this.value = value;
27     }
28     static MyInteger getInstance(int value) throws IntegerRangeException
29 {
30         if (value < 20)
31             throw new IntegerRangeException("too low to create.");
32         else if (value > 40)
33             throw new IntegerRangeException("too high to create.");
34         else
35             return new MyInteger(value);
36     }
37 }

```

程序输出结果：

### 三、改错题（共10分）

下面Java程序中存在五处编译错误，指出错误位置，说明错误原因并更正。

```

1  import java.util.*;
2  public class ArrayDemo {
3      public static void main(String[] args) {
4          List<int> list = new List<int>();
5          list.add(100);
6          list.add(30);
7          list.add(500);
8          list.add(200);
9
10         Integer[] num = new Integer[list.length];
11         // 把集合转成数组
12         list.toArray(num);
13         int max;
14         for (int[] n : num) {
15             if (n > max)
16                 max = n;
17         }
18         System.out.println("最大数值为: " + max);
19     }
20 }

```

### 四、编程题（2 × 15 = 30分）

1. 程序分次从键盘接收一个不多于5位的整数（整数有正负之分），以数据0或非法数据作为输入结束符号。针对每次接收的整数实现如下功能：
  - 1) 计算该整数的位数，并输出。
  - 2) 逆序输出该整数各位数字（负数先输出负号）。
  - 3) 最后输出全部数据各位数字的最大值（负数各位数字全为负值）。

4) 标准输入流类Scanner从属于包java.util

2. 按题目要求完成编程。

1) 设计线程安全的堆栈类Mystack

1.1) 该堆栈类包含整数的数组成员变量value, 以存储堆栈数据。

1.2) 该类包含整型数据类型的成员变量index, 以标识栈顶位置。

1.3) 该类包含一个带整型数形参的构造方法Mystack(int num), 该构造方法创建长度为num 的堆栈。

1.4) 该类包含线程安全的pop及push方法, 分别实现将堆栈顶端数据弹出及向堆栈压栈功能。

1.5) 当堆栈为空时, pop方法抛出异常IndexOutOfBoundsException。

1.6) 当堆栈满时, push方法抛出异常IndexOutOfBoundsException。

2) 创建线程类Thread1和Thread2

2.1) Thread1和Thread2包含Mystack类型的成员变量Data, 指向公共数据区。

2.2) 线程体方法run可以省略。

3) 构建测试类ThreadDemo

3.1) 在main方法中创建容量为10的Mystack的对象实例ms。

3.2) 在main方法中创建具有公共数据区ms的Thread1、Thread2线程对象实例, 并调度执行。