

Lab 1: Java 语法 & 面向对象初步

写在前面:

这里开始是第1个正式的Lab实验，建议在开始之前先行查看"[课程资料-作业注意事项](#)"中具体的实验要求和提交格式

每个Lab开放2周时间，建议大家合理安排时间

1. 实验目的

1. 理解 Java 语言中的关键字、标识符并能灵活使用
2. 理解、掌握 Java 语言的简单数据类型并能灵活使用（重点强制类型转换）
3. 理解引用数据类型，理解简单数据类型和引用数据类型的特点
4. 理解、掌握 Java 语言中的运算符并能灵活使用
5. 理解、掌握常量与变量的概念并能灵活使用
6. 理解、掌握 Java 语言的流程控制（重点 `switch` `case` `break` 语句）并能灵活使用
7. 理解、掌握 Java 语言的数组的声明、赋值及使用
8. 理解掌握面向对象思维方式，确定问题域，使用面向对象思维解决实际问题

2. Hello, World 进阶

Question01

学习一个新语言的第一步，自然是输出 `Hello, World!`。

但在这里我们略微增加一点点难度~ 你需要输出很多 `Hello, World!`

本题需要你实现一个 `Test` 类：

- 程序开始运行，进入 `main` 方法，并输出你的学号和姓名
- 当终端输入 `QUIT` 时，系统退出，并在终端打印一行字符：

```
1 | ----- Good Bye! -----
```

- 对于其他的输入，在终端中输出一行 `Hello, World!`，等待下一行输入

你需要用[命令行](#)运行该程序，并[提交运行时的截图](#)。😊

程序退出

和 C 语言的 `return 0` 类似，Java 程序也有其退出码，整个程序的正常的退出码应该为 0，而不应该为 -1、1 等。

在测评机中，退出状态非 0 则会被认为是程序未正常结束，很容易导致你的作业被错判，因此请务必注意在循环等进行程序退出时，确保退出状态为 0

下面给出 Java 中退出的一个例子：

```
1  class Example {
2      public static void main(String[] args) {
3          int x = 0;
4
5          while (true) {
6              x++;
7              if (x > 100) {
8                  System.out.println("get 100");
9                  System.exit(0);
10             }
11         }
12     }
13 }
14
15 // 这只是退出程序的一种方法
16 // 你也可以直接使用 break, 跳出 while 循环, 直接到达整个程序的退出区域 (默认退出状态为 0)
```

参考实现

Java 打印字符串与其他语言有所区别，一般不直接调用 `print` 函数，可以用如下语句：

```
1  String str = "Hello world!";
2  System.out.println(str);
```

Java 连续读取输入行的一种实现：

```
1  Scanner in = new Scanner(System.in);
2  String argStr;
3  while (true) {
4      argStr = in.nextLine();
5  }
```

当然，`System.out` 中也有其他的打印方法，如 `print` `printf` 等，同学们也可以按需使用。

指路到“课程资料-Java环境配置-3. Java 的命令行编译与运行”

3. 基本数据类型

Question02

编写一段代码验证 Java 中类的基本数据类型字段（成员变量）的默认初始化值，请按照对应类型的字面量填写。在报告中填写下面的表格并附代码。

基本类型	默认值	基本类型	默认值
byte		boolean	
short		char	
int		float	
long		double	

4. 三元运算符

Question03

使用三元运算符补全代码，实现 max, min 的计算，并提交代码及结果内容或截图

```
1 public class FindMinMax {
2     public static void main(String[] args) {
3         double temp, max, min;
4         double d1 = 1, d2 = -9.9, d3 = 96.9;
5         // Example: temp = d1 > d2 ? d1 : d2;
6         // Your code here
7         System.out.println("max = " + max);
8         System.out.println("min = " + min);
9     }
10 }
```

5. 短路

Question04

阅读下面这段代码：

```
1 // ShortCircuit.java
2 public class ShortCircuit {
3     static void print(String s) {
4         System.out.println(s);
5     }
6     static boolean test1(int val) {
7         print("1");
8         return val < 1;
9     }
10    static boolean test2(int val) {
11        print("2");
12        return val < 2;
13    }
14    static boolean test3(int val) {
15        print("3");
16        return val < 3;
17    }
18    public static void main(String[] args) {
19        boolean b = test1(0) && test2(2) && test3(2);
20        print("b is " + b);
21    }
22 }
```

运行该程序，程序的输出是什么？ [给出结果或截图](#)。

请解释 Java 执行串联逻辑运算时的流程， [可以用文字、流程图、伪代码描述](#)。

6. Class初步

Question05

阅读下列 Java 源文件，并回答问题。 [将你的作答以文字或截图形式呈现](#)。

- (1) 上述源文件的名字是什么？
- (2) 编译上述源文件将生成几个字节码文件？这些字节码文件的名字都是什么？
- (3) 在命令行执行 `java Student` 得到怎样的错误提示？执行 `java exam` 得到怎样的错误提示？执行 `java Exam.class` 得到怎样的错误提示？执行 `java Exam` 得到怎样的输出结果？

7. 数组

按照注释补全代码，并提交结果及代码内容或截图。

```
14     int[][] arr1 = new int[3][4];
15     int[][] arr2 = new int[3][];
16     int[][] arr3 = { { 0, 1, 2 }, { 3, 4, 5 }, { 6, 7, 8 } };
17     int i, j, k = 0;
18
19     // 为arr1[][]赋值, 从上到下, 从左到右, 0-11
20     for (i = 0; i < 3; i++) {
21         // arr1为3行4列
22         for (j = 0; j < 4; j++) {
23
24             }
25     }
26
27     // 为arr2[][]开辟内存空间, arr2每一行是变长的, 元素个数为3、4、5
28     for (i = 0; i < 3; i++) {
29         // 参考开辟内存空间: new int[2], 开辟2个int的空间
30
31     }
32
33     // arr2[][]赋值, 从上到下, 从左到右, 从0开始递增
34     for (i = 0; i < 3; i++) {
35         for (j = 0; j < arr2[i].length; j++) {
36
37             }
38     }
39
40     // 输出arr1
41     System.out.println(" arr1:");
42     for (i = 0; i < 3; i++) {
43         for (j = 0; j < 4; j++) {
44
45             //参考: System.out.print(" " + arr1[i][j]);
46         }
47     }
48     System.out.println();
49
50     // 输出arr2
51     System.out.println(" arr2:");
52     for (i = 0; i < 3; i++) {
53         for (j = 0; j < arr2[i].length; j++) {
54
55             }
56     }
57
58     // 输出arr3
59     System.out.println(" arr3:");
```

```

60         for (i = 0; i < 3; i++) {
61             // arr3为3行3列
62             for (j = 0; j < 3; j++) {
63
64             }
65         }
66     }
67 }

```

8. 控制语句

Question07

阅读下面程序，给出程序的输出结果，并解释原因。

```

1  public class SwitchExample {
2      public static void main (String[] args) {
3          char c = '\0';
4          for(int i = 1; i ≤ 4; i++) {
5              switch (i) {
6                  case 1: c = 'J';
7                      System.out.print(c);
8                  case 2: c = 'e';
9                      System.out.print(c);
10                     break;
11                 case 3: c = 'p';
12                     System.out.print(c);
13                 default: System.out.print("好");
14             }
15         }
16     }
17 }

```

Question08

实现程序输出下面这种数字方阵。

```

1  1 2 3 4 5
2  6 7 8 9 10
3  11 12 13 14 15
4  16 17 18 19 20
5  21 22 23 24 25

```

在标准输入中输入 `5`，则输出上面这个 5×5 的方阵。如果输入 `7`，则输出一个 7×7 的方阵。你需要提交 `.java` 代码及运行结果截图。任意选几个有代表性的输入即可。

Question09: 快速排序

编写静态方法 `int[] qsort(int[] arr)`，需求如下

```
1  /**
2   *  对一个int数组快速排序，结果是升序的，并且不会开辟额外的数组空间
3   *  @param arr 待排序的数组
4   *  @return 升序排列好的 arr，如果 arr == null，则返回 null
5   */
6   public static int[] qsort(int[] arr);
```

你需要提交 `.java` 代码。这一次请不要使用 `java.util.Array.sort()` 之类的操作。

约束

- 返回值必须是原数组，即 `return arr`。
- 可以任意次创建指向原数组的引用，但是请不要在排序过程中创建新的数组。
- 可以根据需要定义新的方法。
- 如果数组是 `null`，返回 `null`。
- 对于空数组(即 `arr.length == 0`)，原样返回，而不是 `null`。

9. 引用

Question10

阅读下面这段代码

```
1  class Point {
2      int x, y;
3
4      void setXY(int m, int n) {
5          x = m;
6          y = n;
7      }
8  }
9
10 public class Reference {
11     public static void main(String[] args) {
12         Point p1, p2;
13         p1 = new Point();
```



```

14         p2 = new Point();
15         System.out.println("p1 的引用: " + p1);
16         System.out.println("p2 的引用: " + p2);
17         p1.setXY(1111, 2222);
18         p2.setXY(-100, -200);
19         System.out.println("p1 的 x, y 坐标: " + p1.x + ", " + p1.y); // 1
20         System.out.println("p2 的 x, y 坐标: " + p2.x + ", " + p2.y); // 2
21         p1 = p2;
22         p1.setXY(0, 0);
23         System.out.println("p1 的引用: " + p1);
24         System.out.println("p2 的引用: " + p2);
25         System.out.println("p1 的 x, y 坐标: " + p1.x + ", " + p1.y); // 3
26         System.out.println("p2 的 x, y 坐标: " + p2.x + ", " + p2.y); // 4
27     }
28 }

```

(1) 指出代码中注释标注出的四行输出的内容会是什么。

(2) 什么是**浅拷贝**？什么是**深拷贝**？请你查阅一些资料，并给出你的资讯。

(3) 如果你要为一个类实现 `copy()` 方法，思考如何避免引用间赋值导致的浅拷贝？或者说，如何确保进行的是深拷贝。

(4) 解释 `String` 的 `=` 和 `equals()` 的区别。

10. 可变参数

Question11

编写静态方法 `String strscat(String... args)`，需求如下

```

1  /**
2   * 将任意个字符串顺序连接，不应该改变任意一个原有参数
3   * @param args 字符串们
4   * @return args中的字符串顺序连接组成的新字符串
5   */
6   public static String strscat(String... args);

```

样例：

调用方法	返回值
<code>strscat("a", "b", "c", "", "e")</code>	<code>"abce"</code>
<code>strscat("str")</code>	<code>"str"</code>

你需要提交 `.java` 代码文件并给出几个有代表性的测试的截图。随后回答下面几个问题：

- (1) 尝试理解 Java 可变参数的实现机制，说说你的想法。
- (2) 调用 `strscat(new String[]{"a", "b"})` 能通过编译吗？为什么？
- (3) 如果还有静态方法 `String strscat(String[] args)` 同时存在，代码能通过编译吗？给出 IDE 的编译结果的截图。
- (4) 如果我们声明的是 `String strscat(String[] args)`，`strscat("a", "b", "c")` 这样的调用还能通过编译吗？给出 IDE 的编译结果的截图。

11. 大数

Question12

编写代码实现大数（任意大小数字）的加法，并提交结果及代码内容或截图。

样例：

输入	输出
第一行：12345678901234567890 第二行：98765432109876543210	112112233112233112210

补全代码：（参考引入的库）

```
1 import java.math.BigInteger;
2 import java.util.Scanner;
3
4 public class BigNumberAddition {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         // 读取两个大数
8
9
10        // 计算它们的和
11
12
13        // 输出结果
14        System.out.println(sum);
15    }
16 }
```

12. 面向对象

Question13

制作一桌丰盛的年夜饭。定义采购员类和厨师类的属性和行为，在测试类中产生采购员的对象：姑姑，妈妈和你，在测试类中产生出厨师的对象：爸爸和婶婶，协同制作年夜饭。你需要提交 `.java` 代码文件。（能用代码体现你的思考即可，没有标准答案，也没有 online judge）