

Java 语言实验指导书 1

一、实验内容

安装 JDK 并配置环境变量，掌握 Java 语法基础和数组编程技能。

二、实验目标

知识目标：掌握 JDK 安装与环境变量配置，Java 基本数据类型、运算符、流程控制语句及数组的声明、初始化和操作。

能力目标：能够运用所学知识解决实际问题，培养逻辑思维、代码实现和调试能力。

素养目标：培养规范编码的习惯和严谨的程序设计思维。

二、实验内容与步骤

第一部分：基本数据类型与变量

❖ 程序设计要求

声明 Java 所有的基本数据类型变量并初始化。

尝试进行类型转换（包括自动转换和强制转换），理解其规则和潜在风险。

```
public class DataTypeDemo {  
    public static void main(String[] args) {  
        // 1. 声明并初始化各基本数据类型变量（byte, short, int, long, float, double, char,  
        boolean)  
        // 【请学生填写代码】  
        byte byteVar = 100;  
        short shortVar = 200;  
        int intVar = 1000;  
        long longVar = 100000L; // 注意 L 后缀  
        float floatVar = 3.14F; // 注意 F 后缀  
        double doubleVar = 3.1415926;  
        char charVar = 'A';  
        boolean booleanVar = true;  
  
        // 2. 输出各变量的值  
        // 【请学生填写代码】使用 System.out.println 输出每个变量  
  
        // 3. 类型转换实践  
        // 3.1 自动类型转换（隐式转换）：将一个小范围类型的值赋给一个大范围类型的  
        // 变量  
        int autoConverted = byteVar; // 将 byte 赋值给 int  
        System.out.println("autoConverted: " + autoConverted);  
        // 【请学生填写代码】尝试将 int 赋值给 long，将 float 赋值给 double  
  
        // 3.2 强制类型转换（显式转换）：将一个大范围类型的值赋给一个小范围类型的  
        // 变量  
        byte forcedConverted = (byte) intVar; // 将 int 强制转换为 byte  
        System.out.println("forcedConverted: " + forcedConverted);  
        // 【请学生填写代码】尝试将 long 强制转换为 int，将 double 强制转换为 float
```

```
// 4. 思考：如果将一个较大的 long 值（例如：long largeLong = 10000000000L;）强制转换为 short，结果会怎样？为什么？
// 【请学生填写代码并观察结果】
long largeLong = 10000000000L;
short convertedShort = (short) largeLong;
System.out.println("convertedShort: " + convertedShort);
}
```

？ 结果与知识点提问

运行程序，提供完整代码和运行结果截图。

观察各种类型转换的输出结果。强制转换 `largeLong` 为 `short` 后，输出值是你预期的吗？为什么？

什么是自动类型转换？在什么情况下会发生？

什么是强制类型转换？它可能会带来什么风险（如数据溢出、精度丢失）？如何避免？

`float` 和 `double` 变量在声明和初始化时，为什么要加 `F` 和 `L` 后缀？

第二部分：运算符与表达式

❖ 程序设计要求

使用算术运算符完成基本数学运算。

使用关系运算符比较两个数值，并输出布尔结果。

使用逻辑运算符组合多个布尔条件。

```
import java.util.Scanner;
```

```
public class OperatorDemo {
    public static void main(String[] args) {
        // 1. 算术运算符：+、-、*、/、%
        int a = 10;
        int b = 3;
        int sum = a + b;
        int difference = a - b;
        int product = a * b;
        // 【请学生填写代码】计算 a/b 和 a%b
        System.out.println("a / b = " + (a / b)); // 整数除法
        System.out.println("a % b = " + (a % b)); // 取模

        // 2. 关系运算符：>、<、>=、<=、==、!=
        // 【请学生填写代码】比较 a 和 b 的大小，输出 a > b, a == b, a != b 的结果
        System.out.println("a > b: " + (a > b));
        System.out.println("a == b: " + (a == b));
        System.out.println("a != b: " + (a != b));

        // 3. 逻辑运算符：&&、||、!
    }
}
```

```

boolean x = true;
boolean y = false;
// 【请学生填写代码】计算 x && y, x || y, !x 的值并输出
System.out.println("x && y: " + (x && y));
System.out.println("x || y: " + (x || y));
System.out.println("!x: " + (!x));

// 4. 综合应用：从键盘输入一个整数，判断其是否为偶数
Scanner scanner = new Scanner(System.in);
System.out.print("请输入一个整数: ");
int num = scanner.nextInt();
// 【请学生填写代码】使用关系运算符和逻辑运算符判断 num 是否为偶数
boolean isEven = (num % 2 == 0);
System.out.println(num + " 是偶数吗? " + isEven);
scanner.close();
}
}

```

？ 结果与知识点提问

$10 / 3$ 的结果是 3 而不是 3.333...，为什么？如何得到更精确的结果？

关系运算符和逻辑运算符的运算结果是什么类型？

表达式 $(num \% 2 == 0)$ 是如何判断一个数是否为偶数的？

第三部分：程序流程控制

❖ 程序设计要求

- 使用 if-else 语句判断一个数的正负和奇偶性。
- 使用 switch 语句根据数字输出对应的星期几。
- 使用 for 循环计算 1 到 100 的整数和。
- 使用 while 循环求 100 以内所有偶数的和。
- 使用 do-while 循环模拟简单的用户输入验证。

import java.util.Scanner;

```

public class FlowControlDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // 1. if-else 语句
        System.out.print("请输入一个整数: ");
        int number = scanner.nextInt();
        // 【请学生填写代码】判断 number 的正负
        if (number > 0) {
            System.out.println(number + " 是正数。");
        } else if (number < 0) {

```

```
        System.out.println(number + " 是负数。");
    } else {
        System.out.println(number + " 是零。");
    }
// 【请学生填写代码】判断 number 的奇偶性

// 2. switch 语句
System.out.print("请输入一个 1-7 之间的数字: ");
int day = scanner.nextInt();
// 【请学生填写代码】使用 switch 根据 day 输出对应的星期几
switch (day) {
    case 1:
        System.out.println("星期一");
        break;
    case 2:
        System.out.println("星期二");
        break;
    // ... 补充其他情况
    default:
        System.out.println("输入错误！");
}

// 3. for 循环：计算 1 到 100 的整数和
int sumFor = 0;
// 【请学生填写代码】使用 for 循环计算 sumFor
for (int i = 1; i <= 100; i++) {
    sumFor += i;
}
System.out.println("1 到 100 的整数和 (for 循环): " + sumFor);

// 4. while 循环：求 100 以内所有偶数的和
int sumWhile = 0;
int j = 1;
// 【请学生填写代码】使用 while 循环计算 sumWhile
while (j <= 100) {
    if (j % 2 == 0) {
        sumWhile += j;
    }
    j++;
}
System.out.println("100 以内所有偶数的和 (while 循环): " + sumWhile);

// 5. do-while 循环：模拟输入验证，直到输入正确的密码为止
String correctPassword = "123456";
```

```

String inputPassword;
do {
    System.out.print("请输入密码: ");
    inputPassword = scanner.next();
    if (!inputPassword.equals(correctPassword)) {
        System.out.println("密码错误, 请重新输入! ");
    }
} while (!inputPassword.equals(correctPassword)); // 【请学生填写循环条件】
System.out.println("密码正确, 登录成功! ");

scanner.close();
}
}

```

？ 结果与知识点提问

`if-else` 语句和 `switch` 语句在处理多分支选择时各有何特点？如何选择？

`for`、`while` 和 `do-while` 三种循环语句的执行流程有何异同？`do-while` 循环至少会执行几次循环体？

在 `switch` 语句中，如果忘记写 `break` 关键字，会发生什么现象？这叫做什么现象？

循环语句中的循环变量（如 `for` 中的 `i`, `while` 中的 `j`）的作用是什么？如何避免死循环？

第四部分：数组的应用

❖ 程序设计要求

声明、初始化一个整型数组。

使用循环遍历数组并输出每个元素。

找出数组中的最大值和最小值。

对数组元素进行升序排序（可使用冒泡排序）。

在排序后的数组中，使用二分查找算法查找指定元素。

```
import java.util.Arrays;
```

```
import java.util.Random;
```

```
import java.util.Scanner;
```

```
public class ArrayDemo {
```

```
    public static void main(String[] args) {
```

```
        // 1. 数组的声明、动态初始化和赋值
```

```
        int[] array = new int[10]; // 声明一个长度为 10 的整型数组
```

```
        Random random = new Random();
```

// 【请学生填写代码】使用 `Random` 类生成随机数，为数组的每个元素赋值（范围 1-100）

```
        for (int i = 0; i < array.length; i++) {
```

```
            array[i] = random.nextInt(100) + 1; // 生成 1-100 的随机数
```

```
}
```

```
// 2. 遍历数组并输出
System.out.println("原始数组:");
// 【请学生填写代码】使用 for 循环遍历并打印数组
for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
}
System.out.println(); // 换行

// 3. 求数组的最大值和最小值
int max = array[0];
int min = array[0];
// 【请学生填写代码】遍历数组，找出 max 和 min
for (int i = 1; i < array.length; i++) {
    if (array[i] > max) {
        max = array[i];
    }
    if (array[i] < min) {
        min = array[i];
    }
}
System.out.println("最大值: " + max);
System.out.println("最小值: " + min);

// 4. 数组排序（冒泡排序）
// 【请学生填写代码】实现冒泡排序算法对数组进行升序排序
for (int i = 0; i < array.length - 1; i++) {
    for (int j = 0; j < array.length - 1 - i; j++) {
        if (array[j] > array[j + 1]) {
            // 交换元素
            int temp = array[j];
            array[j] = array[j + 1];
            array[j + 1] = temp;
        }
    }
}
System.out.println("排序后的数组:");
System.out.println(Arrays.toString(array)); // 使用 Arrays.toString 快速输出

// 5. 二分查找（必须在有序数组上进行）
Scanner scanner = new Scanner(System.in);
System.out.print("请输入要在数组中查找的数字: ");
int target = scanner.nextInt();
int low = 0;
int high = array.length - 1;
```

```

int index = -1; // 初始化为-1, 表示未找到
// 【请学生填写代码】实现二分查找算法
while (low <= high) {
    int mid = (low + high) / 2;
    if (array[mid] == target) {
        index = mid;
        break;
    } else if (array[mid] < target) {
        low = mid + 1;
    } else {
        high = mid - 1;
    }
}
if (index != -1) {
    System.out.println("找到元素 " + target + ", 其在数组中的索引为: " + index);
} else {
    System.out.println("数组中未找到元素 " + target);
}
scanner.close();
}
}

```

？ 结果与知识点提问

数组的索引（下标）是从什么数字开始的？访问 `array[array.length]` 会发生什么？

冒泡排序算法的核心思想是什么？

二分查找算法的前提条件是什么？为什么？相比简单的顺序查找有何优势？

Java 中数组的长度是固定的吗？如何获取数组的长度？

四、实验总结与思考

请同学们总结实验收获与问题。

五、实验评分

请以学号+姓名方式命名实验报告，提交 word 形式，5 号宋体。

学号	姓名	评分	评语