

华东师范大学数据科学与工程学院上机实践报告

课程名称：计算机网络与编程	年级：2022 级	上机实践成绩：
指导教师：张召	姓名：李芳	学号：10214602404
上机实践名称： java 编程基本语法和基础 1		上机实践日期：2024.03.08
上机实践编号：02	组号：	上机实践时间：

一、题目与实现情况

task1

- task1: 给出以下基本类型，请判断这些赋值是否正确，若有错误，请在实验报告中指出错在何处：

```
char c1 = '中';
char c2 = '哈哈';
float f1 = 54.321;
boolean bo1 = 1;
```

- 正确
- 错误，Java 中，字符类型只能有一个字符，“哈哈”属于字符串。
- 错误，Java 中，小数默认 double 类型，所以 54.321 直接赋值给 float 类型的变量会出错。需要在数字后面 f 或 F。
- 错误，Java 中，不能直接用整数值来初始化布尔类型的变量，boolean 类型只能取值为 true or false。

task2

- task2: 请在实验报告中说明以下代码段的作用：

```
public static int temp(int a, int b){
    return a == 0 ? b : temp(b % a, a);
}
```

这个递归函数的作用是计算两个整数的最大公约数，采用了欧几里德算法，通过不断取两个数的余数来求得它们的最大公约数。

如果 a 为 0，则返回 b(最大公约数)。否则，递归调用 temp 函数，传入参数 b % a 和 a（辗转相除法：b 除以 a 得到余数，然后继续用 a 和余数进行递归操作，直到其中一个数变为 0）。

task3

- task3: 请在实验报告中使⤵用控制流程打印如下图案：



The screenshot shows an IDE with two tabs: 'LearnDemo02.java' and 'PrintTask3.java'. The 'PrintTask3.java' tab is active, displaying the following code:

```
1 public class PrintTask3 {
2     public static void main(String[] args){
3         String init_out = "*";
4         for (int i = 0; i < 5; i++) {
5             System.out.println(init_out);
6             init_out = init_out + "*";
7         }
8     }
9 }
10
```

Two red annotations with arrows point to specific parts of the code:

- An arrow points to the `for` loop header `for (int i = 0; i < 5; i++)` with the text: "使用 for 循环实现题目功能 便捷表示: 5.fori".
- An arrow points to the `+` operator in `init_out = init_out + "*";` with the text: "字符和字符串都可以直接用加法".

Below the code editor, the '运行' (Run) tab is active, showing the execution command and output:

```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:..."
*
**
***
****
*****
进程已结束,退出代码为 0
```

task4&bonus task1

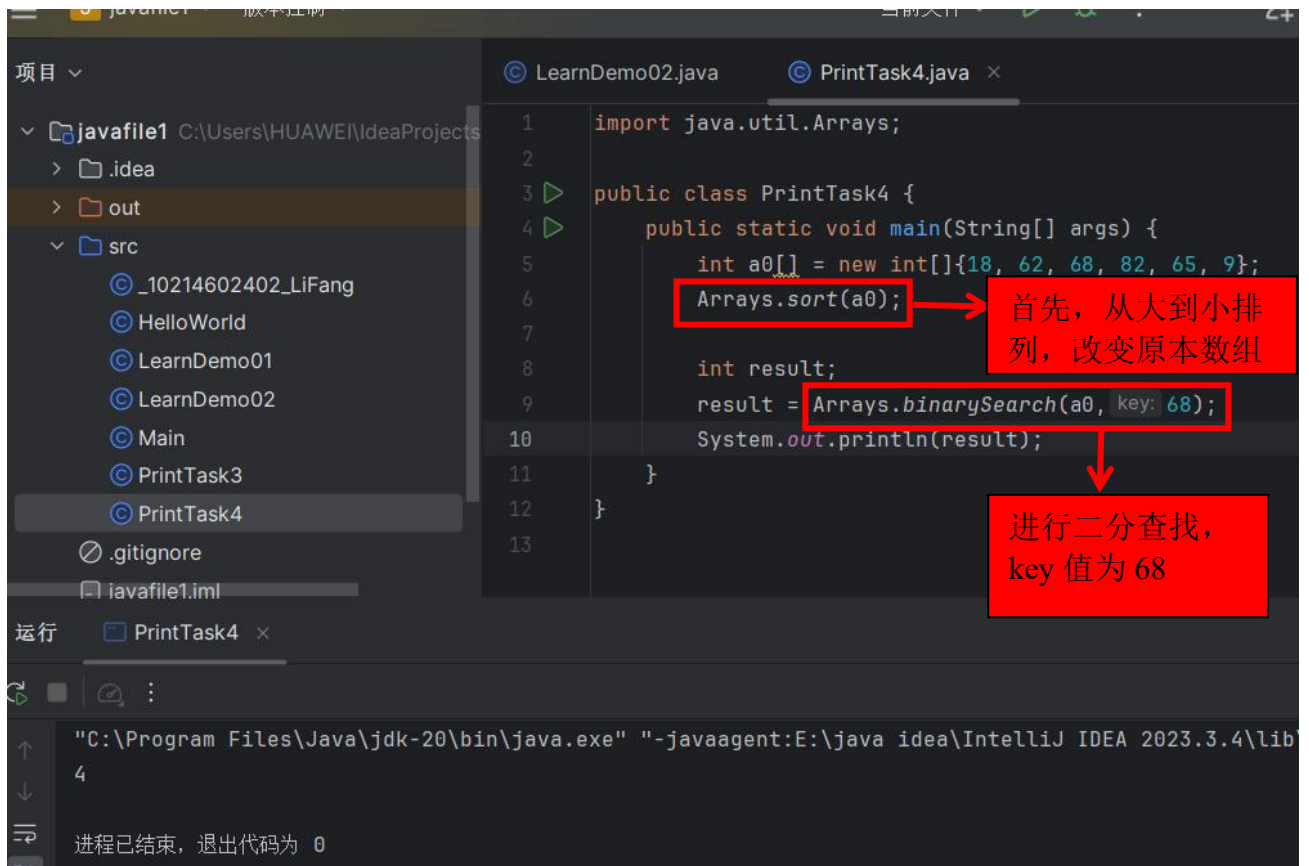
排序之前 :

[18, 62, 68, 82, 65, 9]

排序之后:

[9, 18, 62, 65, 68, 82]

- **task4:** 根据上面所给数组, 将之排序后, 使用 `Arrays.binarySearch()` 查找 68, 请在实验报告中附上结果截图
- **bonus task1 (optional):** 试分析 `Arrays.sort(int[])` 和 `Arrays.sort(Integer[])` 时有何差异



Arrays.sort(int[]) 和 Arrays.sort(Integer[]) 差异:

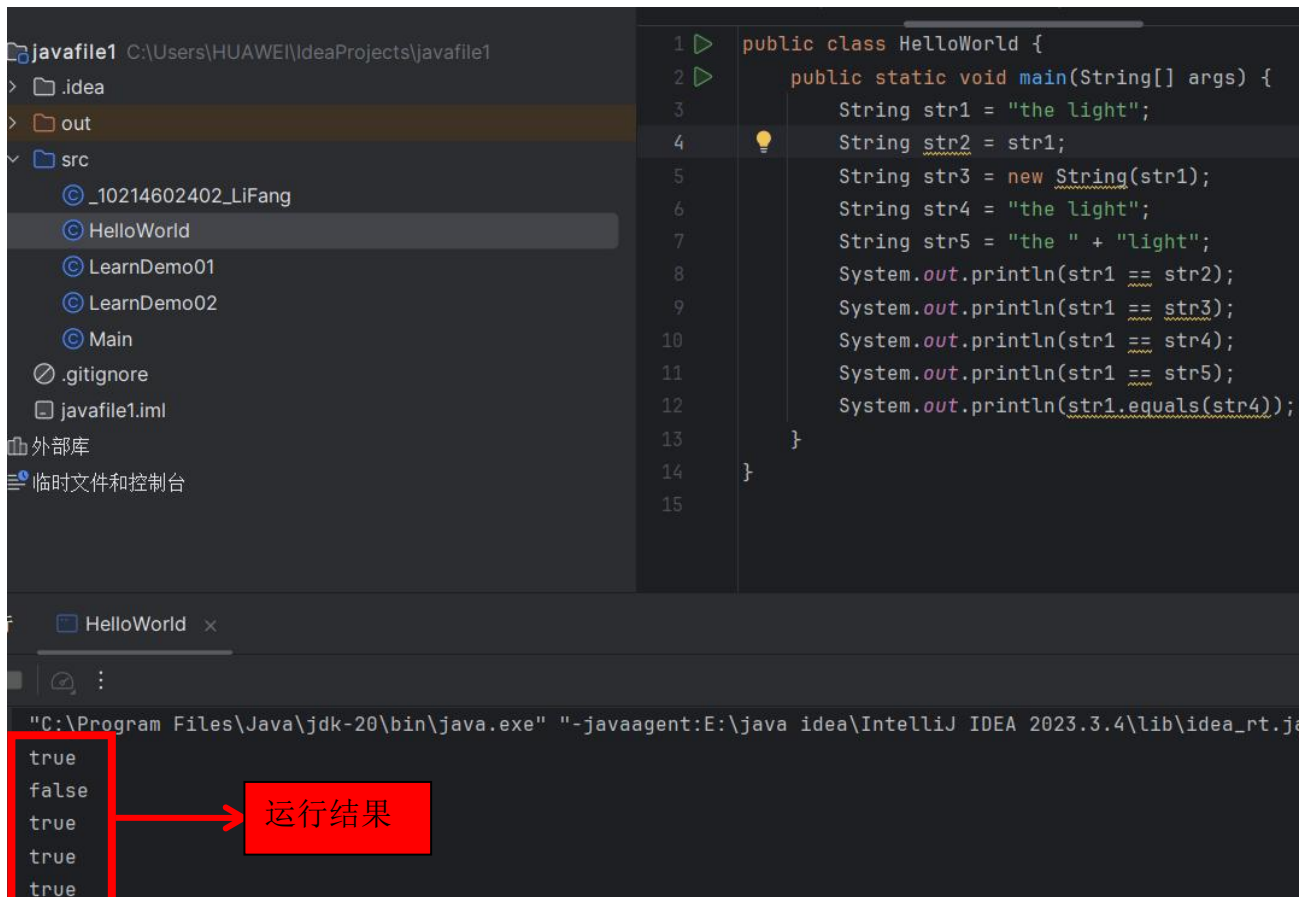
Arrays.sort(int[]) 的参数是基本数据类型 int 的数组，Arrays.sort(Integer[]) 的参数是 Integer 类型的对象数组。所以当使用 Arrays.sort(int[]) 时，需要将基本数据类型 int 转换为对应的包装类 Integer，而 Arrays.sort(Integer[]) 接受的是对象数组，因此不需要手动进行装箱操作。另外，对于前者不支持 null 值，后者支持。

task5

- task5: 试运行下列代码，在实验报告中附上结果截图，首先详细阐述 == 和 equals() 的区别，然后查阅相关资料依次说明原因

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

结果截图:



The screenshot shows the IntelliJ IDEA IDE. On the left is the Project Explorer with a tree view containing folders like .idea, out, and src, and files like _10214602402_LiFang, HelloWorld, LearnDemo01, LearnDemo02, Main, .gitignore, and javafile1.iml. The main editor displays the source code of a Java class named HelloWorld. The code defines a main method that creates several String variables and prints the results of equality comparisons. The output window at the bottom shows the execution results of the program.

```
1 public class HelloWorld {
2     public static void main(String[] args) {
3         String str1 = "the light";
4         String str2 = str1;
5         String str3 = new String(str1);
6         String str4 = "the light";
7         String str5 = "the " + "light";
8         System.out.println(str1 == str2);
9         System.out.println(str1 == str3);
10        System.out.println(str1 == str4);
11        System.out.println(str1 == str5);
12        System.out.println(str1.equals(str4));
13    }
14 }
15
```

Output:

```
true
false
true
true
true
```

A red box highlights the output, and a red arrow points to it with the text "运行结果" (Execution Result).

== 和 equals() 的区别:

== 是用来比较两个对象的引用是否相同, 判断它们是否引用同一个内存地址; 而 equals() 方法是用来比较两个对象的内容是否相等。

结果原因:

1. str1 和 str2 都指向 "the light", 它们的引用地址相同, 所以 == 比较返回 true
2. str1 指向 "the light", 而 str3 是通过 new String() 创建属于堆内存, 两者的引用地址不同, 所以 == 比较返回 false
3. str1 和 str4 都指向相同的字符串对象 "the light", 因此引用地址相同, 所以 == 比较返回 true
4. str5 运算结果是 "the light", 与 str1 的 "the light" 对象相同, 因此两者的引用地址也相同, 所以 == 比较返回 true
5. str1 和 str4 都指向相同的字符串对象 "the light", 内容相同, 因此 equals() 返回 true

task6

- task6: 请在实验报告中分析 Scanner 内 next() 和 nextLine() 方法的区别

`//.next()`一次只接收一个不包含空格的字符串，空格出现就会再进行下次读取

```
Scanner sc1 = new Scanner(System.in);
while(sc1.hasNext()){    //判断是否有输入
    String line = sc1.next();    //把输入赋值给字符串line
    System.out.println("next()接收到字符串: "+line);
}
```

`//.nextLine()`一次接收一整行包含空格的字符串

```
/*Scanner sc2 = new Scanner(System.in);
while(sc2.hasNext()){    //判断是否有输入
    String line = sc2.nextLine();    //把输入赋值给字符串line
    System.out.println("nextLine()接收到字符串: "+line);
}*/
```

示例如下:

```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:E:\java_idea\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=5040:E:\java_idea\IntelliJ IDEA 2023.3.4\bin" -Dfile.encoding=UTF-8
Hello World !
next()接收到字符串: Hello
next()接收到字符串: World
next()接收到字符串: !
进程已结束，退出代码为 130
```

`next()`:

task7

- task7: 请使用 `Scanner` 和 `PrintWriter` 编写一个从控制台输入 10 个数字将它们从小到大排序后输出至控制台的程序，请将代码和结果一起截图附在实验报告中

`nextLine()`:

```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:E:\java_idea\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=5040:E:\java_idea\IntelliJ IDEA 2023.3.4\bin" -Dfile.encoding=UTF-8
Hello World !
nextLine()接收到字符串: Hello World !
进程已结束，退出代码为 130
```

The screenshot displays the IntelliJ IDEA IDE with a project named 'Javafile1'. The file 'PrintTask7.java' is open, showing the following code:

```
1 import java.io.PrintWriter;
2 import java.util.Arrays;
3 import java.util.Scanner;
4
5 public class PrintTask7 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int a[] = new int[10];
9         int i = 0;
10
11         while (i < 10) {    //判断条件用sc.hasNext()会死循环
12             a[i++] = sc.nextInt();
13         }
14         Arrays.sort(a);
15         PrintWriter pw = new PrintWriter(System.out);
16         pw.println(Arrays.toString(a));
17         pw.flush();    //注释掉该句无法正常输出
18     }
19 }
20
21
```

The output window shows the execution of the program:

```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:E:\java_idea\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=5040:E:\java_idea\IntelliJ IDEA 2023.3.4\bin" -Dfile.encoding=UTF-8
90 12 3 75 36 48 98 32 0 3
[0, 3, 3, 12, 32, 36, 48, 75, 90, 98]
进程已结束，退出代码为 0
```


task8

- **task8:** 此节需理解类和对象的概念，请在实验报告中阐述 Java 中对象被分配在哪，并列出现 Java 中所有类都具有的方法

Java 中，对象被分配在堆内存中。

所有的类都具有 Object 类中定义的方法：equals(Object obj)（比较两个对象是否相等）、hashCode()（返回对象的哈希码值）、toString()（返回对象的字符串表示）、getClass()（返回对象的运行时类）、notify()（唤醒在该对象上等待的单个线程）、notifyAll()（唤醒在该对象上等待的所有线程）、wait()（导致当前线程等待，直到其他线程唤醒它）、clone()（用于创建并返回对象的副本）、finalize()（在对象被垃圾回收器回收之前调用）。都可以重写 equals()、hashCode() 和 toString() 方法。

二、总结

通过这节课的学习，我已经基本复习了原本自己学习的 java 基础知识，过程中又有新的值得钻研的小点，另外，在进行 task7 时遇到了死循环和无法打印的问题，在自我纠错和请教老师的过程中得到完美解决，bonus task&task8 开始没有思路，通过查找 java 编程书找到了答案。