

华东师范大学数据科学与工程学院实验报告

课程名称：计算机网络与编程

年级：2022

上机实践成绩：

指导教师：张召

姓名：郭夏辉

学号：10211900416

上机实践名称：Java 编程基本语法和基础 1

上机实践日期：2023 年 3 月 10 日

上机实践编号：02

组号：

一、实验目的

1. 熟悉掌握 IntelliJ IDEA 的使用
2. 学习并掌握 Java 编程基础，为之后使用 Java 进行网络编程打下基础

二、实验任务

熟悉变量、操作符、控制流程、数组、字符串、I/O、类和对象

三、使用环境

IntelliJ IDEA2022.3.2

JDK : Java 19

四、实验过程

1.变量

问题：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'; 正确

char c2 = '哈哈'; 不正确。因为 char 类型的变量存的是字符，不能是字符串

float f1=54.321; 不正确。在 java 中，带小数点的默认是 double 类型，其类型长度为 64，超过了 float 的长度 32，不能直接赋值。该题目中，在 54.321 后面加上 f 显式地表示 float 类型浮点数常量便可以进行赋值了。

boolean bo1=1; 不正确。1 默认为 int 类型，无法直接转换为 boolean 类型。

2.操作符

问题：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);  
}
```

根据欧几里得定理 $\gcd(a,b)=\gcd(b\%a,a)$ 可得 temp 函数的目的是求整数 a,b 的最大公约数。

3.控制流程

问题：请在实验报告中使⤵用控制流程打印如下图案：



```
*  
**  
***  
****  
*****
```

```
public class Main {
    public static void main(String[] args) {
        String a="*";
        for(int i=1;i<=5;i++){
            System.out.println(a);
            a+="*";
        }
    }
}
```

```
*
**
***
****
*****
```

进程已结束,退出代码0

4.数组

问题: 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

```
import java.util.Arrays;
public class LearnArray {
    public static void main(String[] args) {
        int a[]=new int[] {18,62,68,82,65,9};
        Arrays.sort(a);
        System.out.println("after sort:");
        System.out.println(Arrays.toString(a));
        System.out.println("binarySearch result:");
        System.out.println(Arrays.binarySearch(a,68));
    }
}
```

```
after sort:
[9, 18, 62, 65, 68, 82]
binarySearch result:
4
```

问题: 试分析 `Arrays.sort(int[])` 和 `Arrays.sort(Integer[])` 时有何差异

首先我来说一下 `int` 和 `Integer` 类型之间的差异。`int` 是基本类型, 直接能存数值。而 `Integer` 是引用类型, 是 `int` 的封装类, 包含一个 `int` 类型的字段, 实际是一个对象。

然后我来说一下 `Arrays.sort(int[])` 和 `Arrays.sort(Integer[])` 之间的差异。根据网上的资料, 结合 `Arrays.sort` 的具体实现, 我发现 `Arrays.sort(int[])` 采用的是快速排序, 但是在 `Arrays.sort(Object)` 情形下, 即 `Arrays.sort(Integer[])` 情况下, java 采用的是归并排序。这两种排序的空间复杂度是不同的, 而且最坏情况的时间复杂度也是不同的。

5. 字符串

问题: 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) {  
        String str1 = "the light";  
        String str2 = str1;  
        String str3 = new String(str1);  
        String str4 = "the light";  
        String str5 = "the "+"light";  
        System.out.println( str1 == str2);  
        System.out.println( str1 == str3);  
        System.out.println( str1 == str4);  
        System.out.println( str1 == str5);  
        System.out.println( str1.equals(str4));  
    }  
}
```

运行结果如下图所示

```
true  
false  
true  
true  
true
```

对于基本类型和引用类型 == 的作用效果是不同的, 对基本类型而言, 比较的是值是否相同; 对引用类型而言, 比较的是引用是否相同。

equals()本质上就是==, 只不过 String 和 Integer 等重写了 equals 方法, 把它变成了值比较。

然后我来分析这五个输出结果。原则还是变量存储的是引用, 而不是引用本身。

第一个 true

str1 赋值给 str2, 本质上来讲是让 str2 和 str1 指向的是同一个引用, 所以==结果返回是 true

第二个 false

new String()方法开辟了新的内存空间, 这样即便 str3 和 str1 的值是一样的, 但是因为指向的引用不同了, 故==返回结果是 false

第三个 true

str1 和 str4 指向的是同一个引用, 所以==结果返回是 true

第四个 true

"the "+"light"将两个字符串拼接成了"the light", 根据 Java 存储字符串的原理, 在没有 new 的情况下遇到了完全相同的字符串使得 str5 和 str1 指向了同一个引用, 所以==结果返回

是 true

第五个 true

str1 和 str4 是完全相同的字符串，比较数值层面自然是一样的，故 equals()得结果是 true

6.I/O

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

```
import java.util.Scanner;

public class HelloWorld {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String test1 = sc.next();
        System.out.println("next result:"+test1);
        String test2 = sc.nextLine();
        System.out.println("next result:"+test2);
    }
}
```

```
Hello There I am Tommy and I love Network.
next result:Hello
next result: There I am Tommy and I love Network.
|
进程已结束,退出代码0
```

可以看到，next()方法遇到空格、Tab 这样的特殊字符就会暂停输入。

但是，nextLine()方法是遇到回车才输入结束，这样会把空格、Tab 这样的特殊字符也输入进去。

问题: 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

```
import java.io.PrintWriter;
import java.util.Scanner;
import java.util.Arrays;
public class LearnPrintWriter {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        PrintWriter pw = new PrintWriter(System.out);
        int []a=new int[10];
        for(int i=0;i<10;i++){
            a[i]=sc.nextInt();
        }
        Arrays.sort(a);
```

```

        pw.println("After sort:");
        pw.flush();
        pw.println(Arrays.toString(a));
        pw.flush();
    }
}

666 78 9 -7 8883 5 456 712 860 13
After sort:
[-7, 5, 9, 13, 78, 456, 666, 712, 860, 8883]

进程已结束,退出代码0

```

7.类和对象

问题: 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 java 中所有类都具有的方法。

堆是线程共享的内存区域, 栈是线程独享的内存区域。堆是线程共享的内存区域, 栈是线程独享的内存区域。在 Java 虚拟机中, 堆是可供各个线程共享的运行时内存区域, 也是供基本上所有的类实例和数组对象分配内存的区域。正常情况下, 对象是要在堆上进行内存分配的, 但是随着编译器优化技术的成熟, 虽然虚拟机规范是这样要求的, 但是具体实现上还是有些差别的。如 HotSpot 虚拟机引入了 JIT(Just In Time, 即时编译)优化之后, 会对对象进行逃逸分析, 如果发现某一个对象并没有逃逸到方法外部, 那么就可能通过标量替换来实现栈上分配, 而避免堆上分配内存。

Java 中所有的类都有的方法是

(一)构造方法。如果没有显式地为类定义构造方法, Java 编译器将会为该类提供一个默认构造方法。

(二)Object 类是所有类的父类, 也就是说 Java 的所有类都继承了 Object, 子类可以使用 Object 的所有方法。总体来看 Object 类有 12 个方法, 9 个 public 修饰的方法; 2 个 protected 修饰的方法, 1 个 private 修饰的方法。

1. equals 比较两个对象是否相等
2. getClass 获取对象的运行时对象的类
3. hashCode 获取对象的 hash 值
4. notify 和 notifyAll 唤醒在该对象上等待的某个/所有线程
5. toString 返回对象的字符串表示形式
6. wait 让当前线程进入等待状态。直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法。
7. wait(long) 让当前线程处于等待(阻塞)状态, 直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法, 或者超过参数设置的 timeout 超时时间。
8. wait(long,int) 与 wait(long timeout) 方法类似, 多了一个参数, 这个参数表示额外时间(以纳秒为单位, 范围是 0-999999)。
9. clone 创建并返回一个对象的拷贝 protected
10. finalize 当 GC(垃圾回收器)确定不存在对该对象的有更多引用时, 由对象的垃圾回收器调用此方法。protected

11. registerNatives 该方法应该是用来注册本地方法的。private

五、总结

通过本次实验，在老师的悉心指导和自己广泛地上网查阅资料情况下，我了解并掌握了基础的 Java 编程语法，这不仅是为日后的实验打基础，更是为自己的后端开发之路奠定了坚实茁壮的地基。