学号 X41714008 专业 计算机科学与技术 姓名 李逢时

实验日期 **19/2/25**  教师签字 成绩

实验报告

【实验名称】 第二章 类和对象

【实验目的】

1. 学习Java的基本语法
2. 学习Java的运算符和控制语句
3. 学习类和对象的概念

【实验原理】

1. Java程序设计基本原理
2. 面向对象程序设计基本原理

【实验内容】

**一、填空题**

1. System.out.println(“12”+ 5)；语句的输出结果是 125 。

2. Java语言中“2.5%1.2” 的运算结果是 0.10000000000000009 。

**二、简答**

1. 为什么类成员方法只能操作类成员变量，而不能操作实例变量？

[解答]

因为类成员变量、类成员方法在.class文件被解释过程的一开始即被初始化，而此时并不存在任何对象实例，进而不存在任何实例变量，自然不可能允许可能操作到不存在变量的操作。

1. 简述指针和引用有什么区别？

[解答]

* 引用是对象的别名，而指针代表地址。
* 引用是无法改变指向的指针的一层封装，指针可以改变其指向。
* 指针存在以下缺陷，而引用不存在：
  + 通过指针读出的某个位置数据可能不符合期望的假设
  + 可能访问到其他进程已经占用的资源
  + 指针访问存在越界的可能

3. 请解释import java.util.Scanner; 语句的含义？

[解答]

从隐式指定的rt.jar（相当于java的标准库）中，从路径java/util/中导入Scanner类。

**三、实验题**

1. 指出程序中的错误，并修正

public class Example{

public static void main(String args[]){

int a=1.0;

float fa=216;

float fb=21.656f;

double da=125e45d;

double db=123.45;

float fc=0.1;

}

}

[解答]：

错误：程序的第3行、第8行发生编译错误，错误类型为incompatible types。

原因：程序的第3行和第8行（分别对应第一个和第六个赋值操作）会发生缩窄转换，而java默认不允许隐式的缩窄转换发生，需要通过强制类型转换语法显式进行这种操作。

修正：  
第三行：int a = (int)1.0

第八行：float fc = 0.1f;

2.写出如下程序输出结果，并说明原因

**public class E\_c{**

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

**String sx1="my";**

**String sx2="god";**

**char cx='a';**

**System.out.println(sx1+sx2);**

**System.out.println(sx2+cx);**

**System.out.println(sx1+5);**

**System.out.println(cx+5);**

**}**

**}**

运行结果：  
mygod

goda

my5

102

解释：

第一行输出结果就是两个字符串append之后的结果。

第二行，因+表达式中，一个操作数为字符串，则整个表达式视为字符串append之后的结果。此处即为sx2字符串和被视为字符串的cx变量append的结果。

第三行，同2的原因，将5视为字符串，之后和sx1做append，得到my5的输出结果。

第四行，不存在任何stirng操作数，则将cx +５视为一个代数表达式，’a’的ascii码值为97，加上5的结果就是102。

3、把如下程序**改成从键盘输入两个整数，并判断大小。**

**public class FindBigone{**

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

**int i1=127;**

**int i2=59;**

**if(i1>=i2){**

**System.out.println("The Bigger One is:"+i1);**

**}else{**

**System.out.println("The Bigger One is:"+i2);**

**}**

**}**

**}**

**改动后程序：**

**import java.util.Scanner;**

**public class FindBigone**

**{**

**public static void main(String[] args)**

**{**

**Scanner reader = new Scanner(Stystem.in);**

**Syhstem.out.println(“Please input 2 integers for comparion:”);**

**int num1 = reader.nextInt();**

**int num2 = reader.nextInt();**

**if(num1>num2){**

**System.out.println("The Bigger One is:"+num1);**

**}else if (numj1 < num2){**

**System.out.println("The Bigger One is:"+num2);**

**}else{**

**System.out.println("The 2 numbers are equal.");**

**}**

**}**

**}**

**4、分析简单类型参数和引用类型参数的区别，运行下列程序并抓图，请解释原因。**

**public class PassTest{**

**float ptValue;**

**public void changeInt(int value){ //基本类型参数**

**value=55;**

**}**

**public void changeObjValue(PassTest ref){ //引用类型参数**

**ref.ptValue=99f;**

**}**

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

**int val;**

**PassTest pt=new PassTest();**

**val=11;**

**pt.changeInt(val); System.out.println(val);**

**pt.ptValue=101f;**

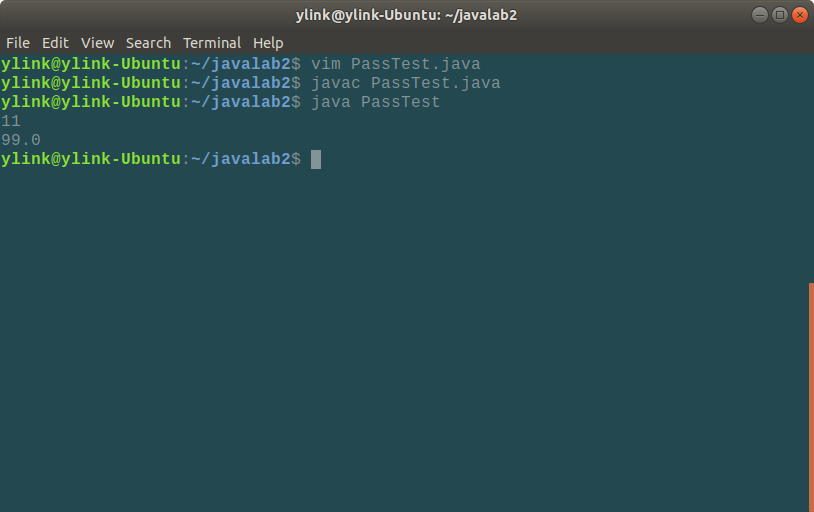
**pt.changeObjValue(pt); System.out.println(pt.ptValue);**

**}**

**}**

**原因：int在java中属于基本类型，函数传参时遵循值传递规则，故在函数内对值的修改不会影响到函数外的对象。PassTest属于用户自定义类型，函数传参遵循传引用的规则，故在函数体内对对象的修改会影响到函数体实际传入的对象本身，在函数体外该修改仍能保持·。**

**截图：**

****

**5、写出下列程序输出，并解释**

**public class Student{**

**String name;**

**static String school="北京大学";**

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

**Student stu1=new Student();**

**Student stu2=new Student();**

**Student stu3=new Student();**

**stu1.name="1";**

**stu2.name="2";**

**stu3.name="3";**

**System.out.println(stu1.name+","+stu1.school);**

**System.out.println(stu2.name+","+stu2.school);**

**System.out.println(stu3.name+","+stu3.school);**

**stu2.school="安徽大学";**

**System.out.println(stu1.name+","+stu1.school);**

**System.out.println(stu2.name+","+stu2.school);**

**System.out.println(stu3.name+","+stu3.school);**

**} //思考程序的输出结果**

**}**

**输出结果：**

**1, 北京大学**

**2, 北京大学**

**3, 北京大学**

**1, 安徽大学**

**2, 安徽大学**

**3, 安徽大学**

**解释：school是类静态成员，属于整个类，可通过具体的对象或类型进行访问。对一个类静态成员的修改会在所有该静态成员的对象实例中有所体现。name是非静态成员，对name的修改只会在对应的某个对象得到体现。**

**6、使用命令行运行下面程序。**

**package s1.s2.s3.s4;**

**public class Hello{**

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

**System.out.println("Hello");**

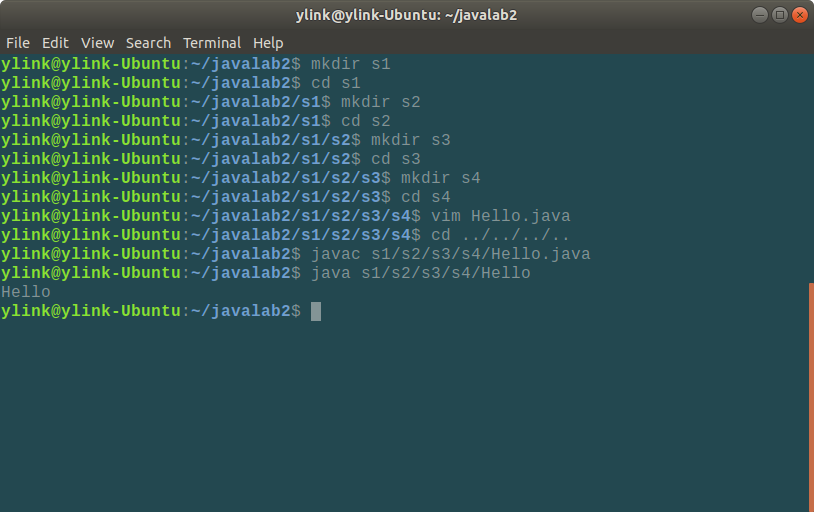
**}}**

1. **如何存放Hello.java文件？**

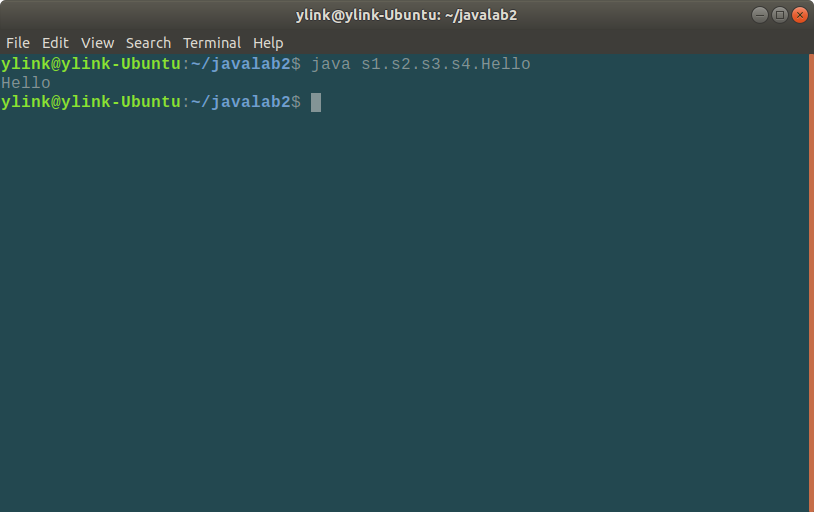
**将Hello.java**文件放在嵌套文件夹s1/s2/s3/s4下。

**（2）如何编译和运行？请抓图。**

**如截图所示。**

****

**补充另一种执行的方式：**

****

【小结或讨论】

通过本次实验，对java的import、package机制得到了充分掌握。另外通过java的引用及其与C++指针机制的对比，感受到java在程序开发过程中的优越性。