**重庆交通大学信息科学与工程学院**

**综合性设计性实验报告**

**班 级： 计科1703**

**姓名 学号： 黄富灵 631707060308**

**实验项目名称： 实验二、面向对象入门**

**实验项目性质： 验证性、设计性**

**实验所属课程： 《Java程序设计》**

**实验室(中心)： B01 409**

**指 导 教 师 ： 王桂平**

**实验完成时间： 2019年 3月 4 日**

**一、实验概述**

**【实验目的】**

1. 理解面向过程和面向对象。

2. 学会查阅JDK API文档。

3. 在类的方法中熟练使用三种基本控制结构。

**【实验要求】**

1. 按照题目要求正确建立实验文件夹。

2. 使用IDE工具编写Java程序时，按照题目要求正确地建立项目、包、类等，正确地把相关文件保存在指定的位置。

3. 按题目要求记录实验结果。

**【实施环境】（使用的材料、设备、软件）**

JDK、NetBeans、Eclipse

**二、实验内容**

**每次实验前，请确认是否已经严格按实验一中的规范新建了实验所需的文件夹，是否已经按实验一中的规范新建了NetBeans和Eclipse项目。**

**【实验报告和代码提交】完成实验报告后，将“学号+姓名”文件夹压缩，发给班长，由班长汇总后再发到w\_guiping@163.com。课程结束后，再把所有实验报告打印一份交给老师存档。**

**实验任务一 面向对象入门**

**第1题 直接用JDK实现讲义第3章中的“模拟大学课堂”例子**

【实验内容】

第3章中的“模拟大学课堂”例子采用2种途径实现：

途径1：直接在Teacher类中定义main方法；

途径2：新建一个Lesson类，并在其中定义main方法。

【思考】现在要在exp2文件夹中定义两个Student类、两个Teacher类，对应到相同的源文件，怎么办？

【提示】进一步在exp2文件夹中新建test1、test2文件夹，分别存放2种途径实现的代码。

【实验步骤】

按照上述两种途径实现讲义第3章中的“模拟大学课堂”例子。

实现途径1：

1. 在test1文件夹中新建源文件Student.java、Teacher.java，把讲义中的代码拷贝到对应源文件中**（注意去掉或注释掉package语句）**。

2. 在命令行程序窗口中用javac命令编译上述两个源文件，然后用java命令运行Teacher程序。

实现途径2：

1. 在test2文件夹中新建源文件Student.java、Teacher.java、Lesson.java，把讲义中的代码拷贝到对应源文件中**（注意去掉或注释掉package语句）**。

2. 在命令行程序窗口中用javac命令编译上述三个源文件，然后用java命令运行Lesson程序。

【实验记录】

1. 请如实记录你在实验过程中出现的编译、运行等错误(如发现错误无法解决，请举手)，对程序的错误进行修改并记录。如果没有发现错误则填“无”。

回答：

无

2. 请记录Teacher程序（实现途径1）或Lesson程序（实现途径2）的运行结果（即输出内容）。注意掌握在命令行程序窗口中复制内容的方法。

回答：

Lesson starts!

I am teaching!

Who can answer a question!

I am answering a question!

Lesson ends!

**第2题 在NetBeans或Eclipse中实现讲义第3章中的“模拟大学课堂”例子**

【实验内容】

在NetBeans或Eclipse中实现讲义第3章中的“模拟大学课堂”例子。在用向导新建类时，注意设置包名为“exp2”。（如有疑问，请查阅讲义、请教老师）

【实验记录】

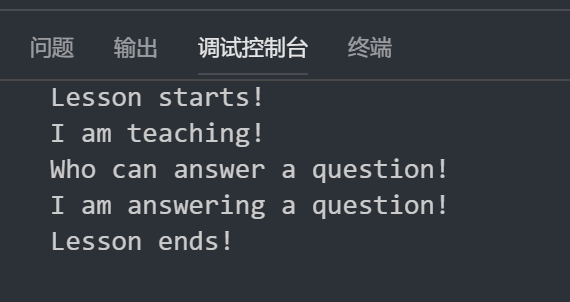
1. 请如实记录你在实验过程中出现的编译、运行等错误(如发现错误无法解决，请举手)，对程序的错误进行修改并记录。如果没有发现错误则填“无”。

回答：

无

2. 请记录该程序的运行结果（即输出内容），请把输出内容截图，只截输出窗口部分，并把截图粘贴在下方。

回答：



**第3题 用JDK、或NetBeans、或Eclipse实现“制造汽车”例子**

【实验内容】

模仿“大学课堂”例子，用Java程序实现“制造汽车”例子。

假设一辆汽车只有四个轮子(Wheel)、一个发动机(Engine)、一个车架(Frame)。注：这三个类可以没有任何属性和方法，但必须定义相应的类(class)。

汽车生产商(Manufacturer)类有一个组装方法(assemble)，输出一行信息：An automobile has been assembled。在Manufacturer类的main方法中new四个轮子对象、new一个发动机对象、再new一个车架对象，最后调用assemble方法组装一辆汽车。

【方案设计】

请将编写的Java源程序的代码粘贴在下方。

回答：

class Manufacturer {

public static void assemble(Engine e, Frame f, Wheel[] w) {

System.out.println("An automobile has been assembled");

}

public static void main(String[] args) {

Wheel [] wheels=new Wheel[4];

Engine engine=new Engine();

Frame frame=new Frame();

assemble(engine,frame,wheels);

}

}

【实验记录】

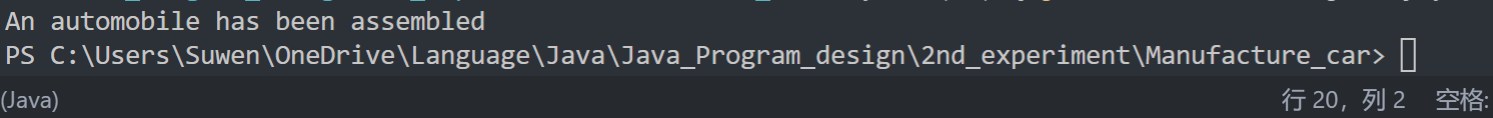
1. 请如实记录你在实验过程中出现的编译、运行等错误（如发现错误无法解决，请举手），对程序的错误进行修改并记录。如果没有发现错误则填“无”。

回答：

无

2. 请记录该程序的运行结果（即输出内容），如果是在命令行程序窗口运行，则复制其中的输出内容；如果是在NetBeans或Eclipse中运行，则截图并粘贴在下方（只截输出窗口部分）。

回答：



**实验任务二 API文档**

**第1题 编写包含文档注释的Java程序，生成API文档**

【实验内容】

在Java实验代码文件夹的exp2子文件夹中再新建一个Test子文件夹，编写以下三个类的Java源文件。参考课件和讲义中的javadoc例子，使用javadoc命令生出一份API文档。

1) Teacher类：

package classes;

/\*\*

\* 模拟大学课堂中的老师类

\*/

public class Teacher {

private String Name;

private String Tno;

/\*\*

\* 老师类的宣布上课方法

\*/

public void startLesson( ){

System.out.println("Lesson starts!");

}

/\*\*

\* 老师类的宣布下课方法

\*/

public void endLesson( ){

System.out.println("Lesson ends!");

}

/\*\*

\* 老师类的提问方法

\*/

public void ask( ){

System.out.println("Who can answer a question!");

}

/\*\*

\* 老师类的讲课方法

\*/

public void teach( ){

System.out.println("I am teaching!");

}

}

2) Student类：

package classes;

/\*\*

\* 模拟大学课堂中的学生类

\*/

public class Student {

private String Name;

private String Sno;

/\*\*

\* 学生类的听课方法

\*/

public void listenLesson( ){

System.out.println("I am listening!");

}

/\*\*

\* 学生类的回答问题方法

\*/

public void answer( ){

System.out.println("I am answering a question!");

}

}

3) Lesson类：

package use;

/\*\*

\* 测试类

\*/

public class Lesson {

/\*\*

\* 测试设计好的教师和学生类，即利用类定义对象并调用方法

\*/

public static void main(String[] arguments){

Teacher t1 = new Teacher();

Student s1 = new Student();

Student s2 = new Student();

Student s3 = new Student();

t1.startLesson();

t1.teach();

t1.ask();

s1.answer();

t1.endLesson();

}

}

【实验记录】

1. 请记录生成API文档的完整javadoc命令。

回答：

javadoc -d apidoc -windowtitle 测试 -doctitle 学习javadoc工具的测试API文档 -header 自定义类 \*

2. 打开生成的API文档，截图后粘贴在此处。

回答：



**第2题 通过查阅JDK的API文档，学会使用JDK中封装好的数据结构—栈(Stack)**

【实验内容】

借助JDK的API文档，查阅java.util包下的Stack类，使用Stack类实现以下功能：依次入栈的整数为：-5、27、35、-123、11、29、3、66，通过控制入栈和出栈操作的顺序，使得依次出栈的整数为：27、35、29、11、3、-123、66、-5。

【方案设计】

请将编写的Java源程序的代码粘贴在下方。

回答：

import java.util.Stack;

public class Stack\_test {

public static void main(String[] args){

Stack s = new Stack();

s.push(-5);

s.push(27);

System.out.println(s.peek());

s.pop();

s.push(35);

System.out.println(s.peek());

s.pop();

s.push(-123);

s.push(11);

s.push(29);

System.out.println(s.peek());

s.pop();

System.out.println(s.peek());

s.pop();

s.push(3);

System.out.println(s.peek());

s.pop();

System.out.println(s.peek());

s.pop();

s.push(66);

System.out.println(s.peek());

s.pop();

System.out.println(s.peek());

s.pop();

}

}

【实验记录】

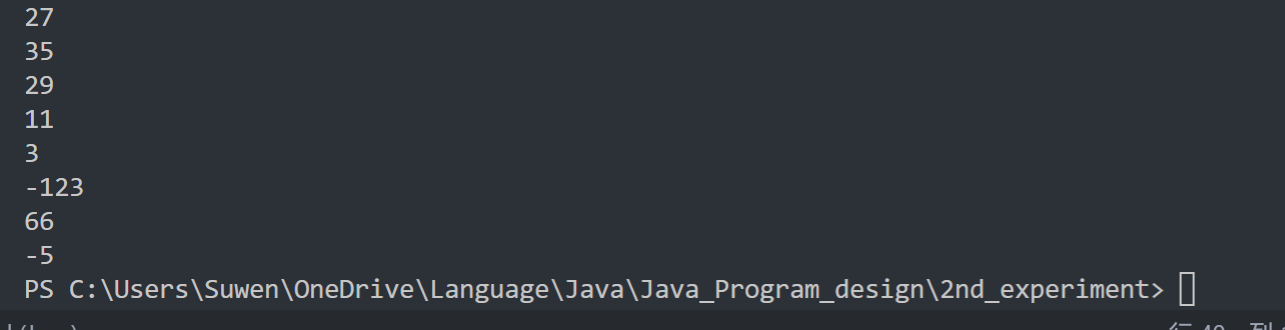
1. 请如实记录你在实验过程中出现的编译、运行等错误(如发现错误无法解决，请举手)，对程序的错误进行修改并记录。如果没有发现错误则填“无”。

回答：

无

2. 请记录该程序的运行结果（即输出内容），如果是在命令行程序窗口运行，则复制其中的输出内容；如果是在NetBeans或Eclipse中运行，则截图并粘贴在下方（只截输出窗口部分）。

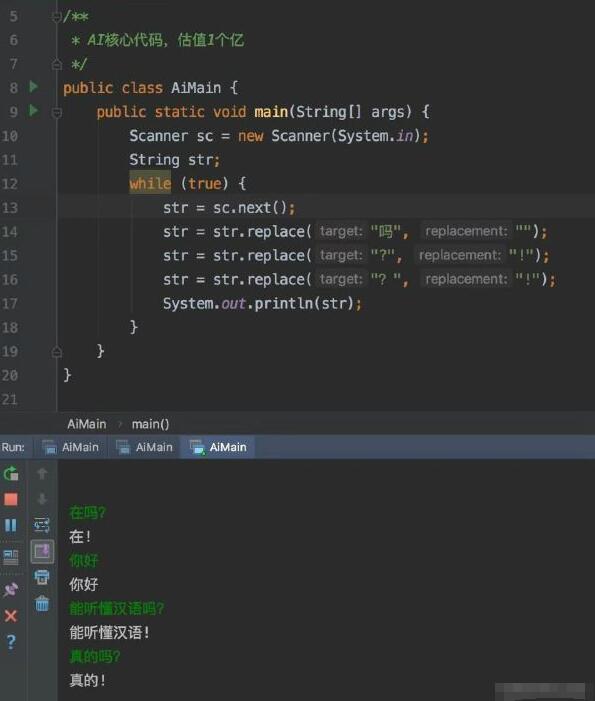
回答：



**第3题 通过查阅JDK的API文档，掌握String类中的方法**

【实验内容】

已知有以下Java代码，通过查阅JDK API文档中String类的方法，理解以下代码是如何实现“智能”的。



【方案设计】

1. 请总结String类的replace方法。

回答：

replace () 方法通过用 newChar 字符替换字符串中出现的所有 oldChar 字符，并返回替换后的新字符串。

语法

public String replace(char oldChar,

char newChar)

参数

oldChar -- 原字符。

newChar -- 新字符。

2. 请问Java中的方法是否支持命名参数，即：在调用方法时，在实参名前加上对应的形参名和冒号，这样就不必考虑方法声明时各参数出现的顺序。

回答：

否

【实验记录】

无。

**实验任务三 在类的方法中熟练使用三种基本控制结构**

**第1题 编写Java程序求解鸡兔同笼问题**

【实验内容】

编写Java程序求解鸡兔同笼问题：一个笼子里关了鸡和兔子（鸡有2只脚，兔子有4只脚）；已知笼子里面脚的总数nFeet（假设nFeet为正整数，从键盘输入），问笼子里可能有多少只鸡，有多少只兔子。

C语言实现的代码，可参考讲义参考文献[17]例1.31。采用以下三种方式之一实现：

* 直接用记事本编写源程序，用JDK命令编译和运行。
* 在NetBeans中编写、编译和运行。
* 在Eclipse中编写、编译和运行。

【方案设计】

请将编写的Java源程序的代码粘贴在下方。

回答：

import java.util.Scanner;

class Cook\_and\_rabbit {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int nFeet = scanner.nextInt();

int x, y;

for (x = 0; x < nFeet; x++) {

y=(int)(nFeet-2\*x)/4;

if(x\*2+y\*4==nFeet&&x>=0&&y>=0)

System.out.println("鸡有"+x+"只,"+"兔子有"+y+"只。");

}

scanner.close();

}

}

【实验记录】

1. 请如实记录你在实验过程中出现的编译、运行等错误(如发现错误无法解决，请举手)，对程序的错误进行修改并记录。如果没有发现错误则填“无”。

回答：

无

2. 请记录该程序的运行结果（即输出内容），如果是在命令行程序窗口运行，则复制其中的输出内容；如果是在NetBeans或Eclipse中运行，则截图并粘贴在下方（只截输出窗口部分）。

回答：

