

**学生实验实习报告册**

|  |  |
| --- | --- |
| 学年学期： | 2020－2021 学年 🗹春🞏秋学期 |
| 课程名称： | 基于Java面向对象程序设计B |
| 学生学院： | 经济管理学院 |
| 专业班级： | Z0322003 |
| 学生学号： | 2020220088 |
| 学生姓名： | 胡彪 |
| 联系电话： | 13982720426 |

**重庆邮电大学教务处制**

|  |  |  |  |
| --- | --- | --- | --- |
| **课程名称** | 基于Java面向对象程序设计B | **课程编号** |  |
| **实验地点** | 综合实验楼B413/B414 | **实验时间** |  |
| **校外指导教师** | **无** | **校内指导教师** | **张鹏** |
| **实验名称** | 实验四 继承、接口与泛型 | | |
| **评阅人签字** |  | **成绩** |  |
| **评语** |  | | |

**一、实验目的**

1. 掌握派生子类的方法  
2. 理解关键字 super 的含义  
3. 理解继承中方法的覆盖机制  
4. 区别重载和覆盖  
5. 理解在继承关系中构造方法的调用过程，掌握接口的定义及使用

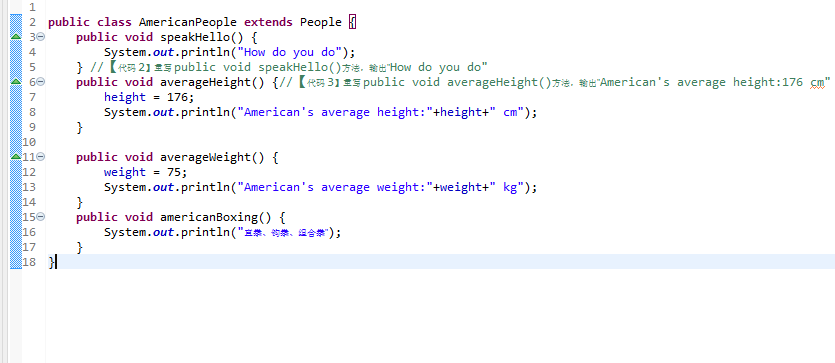
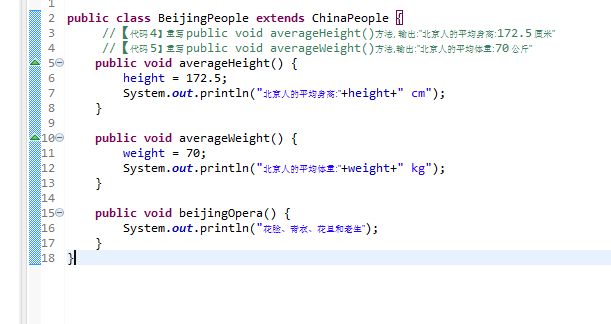
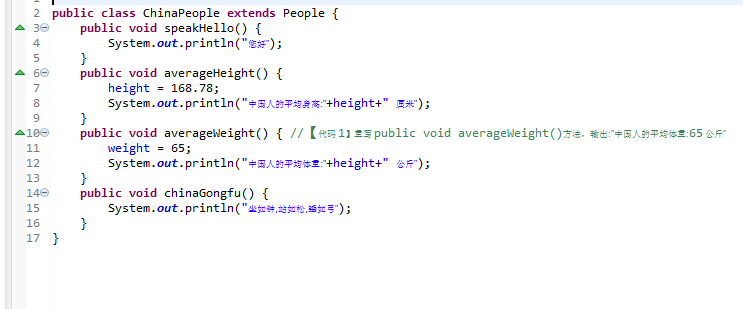
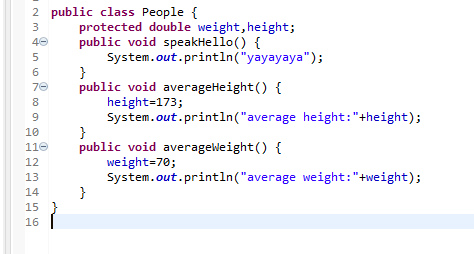
**二、实验环境**

微机+ Window操作系统 + eclipse编辑器

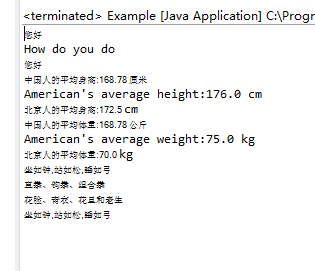
**三、实验内容及步骤**

实验 **1** 中国人与美国人  
程序模板 请按模板要求，将【代码】替换为 Java 程序代码。  
People.java  
public class People {  
protected double weight,height;  
public void speakHello() {  
System.out.println("yayayaya");  
}  
public void averageHeight() {

height=173;  
System.out.println("average height:"+height);  
}  
public void averageWeight() {  
weight=70;  
System.out.println("average weight:"+weight);  
}  
}  
**ChinaPeople.java**public class ChinaPeople extends People {  
public void speakHello() {  
System.out.println("您好");  
}  
public void averageHeight() {  
height = 168.78;  
System.out.println("中国人的平均身高:"+height+" 厘米");  
}  
【代码 1】 //重写 public void averageWeight()方法，输出:"中国人的平均体重:65 公斤"  
public void chinaGongfu() {  
System.out.println("坐如钟,站如松,睡如弓");  
}  
}  
**AmericanPeople.java**  
public class AmericanPeople extends People {  
【代码 2】 //重写 public void speakHello()方法，输出"How do you do"  
【代码 3】 //重写 public void averageHeight()方法，输出"American's average height:176 cm"  
public void averageWeight() {  
weight = 75;  
System.out.println("American's average weight:"+weight+" kg");  
}  
public void americanBoxing() {  
System.out.println("直拳、钩拳、组合拳");  
}  
}  
**BeijingPeople.java**public class BeijingPeople extends ChinaPeople {  
【代码 4】 //重写 public void averageHeight()方法, 输出:"北京人的平均身高:172.5 厘米"  
【代码 5】 //重写 public void averageWeight()方法,输出:"北京人的平均体重:70 公斤"  
public void beijingOpera() {  
System.out.println("花脸、青衣、花旦和老生");  
}  
}  
**Example.java**public class Example {  
public static void main(String args[]) {  
ChinaPeople chinaPeople=new ChinaPeople();  
AmericanPeople americanPeople=new AmericanPeople();  
BeijingPeople beijingPeople=new BeijingPeople();  
chinaPeople.speakHello();  
americanPeople.speakHello();  
beijingPeople.speakHello();  
chinaPeople.averageHeight();  
americanPeople.averageHeight();  
beijingPeople.averageHeight();  
chinaPeople.averageWeight();  
americanPeople.averageWeight();  
beijingPeople.averageWeight();  
chinaPeople.chinaGongfu();  
americanPeople.americanBoxing();  
beijingPeople.beijingOpera() ;  
beijingPeople.chinaGongfu();  
}  
}实验代码截图：

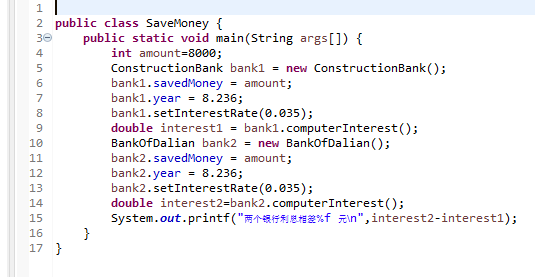
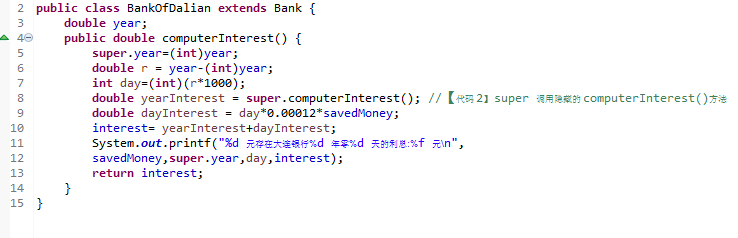
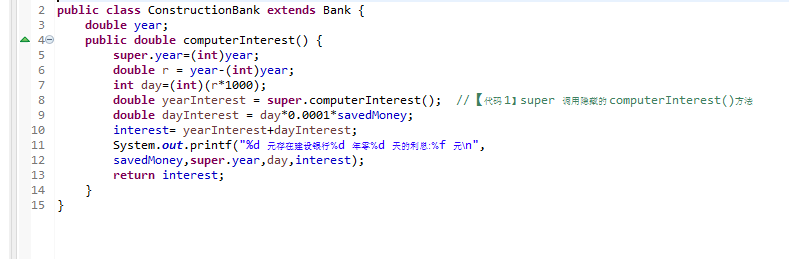
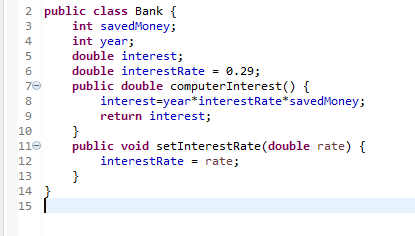


实验结果截图：

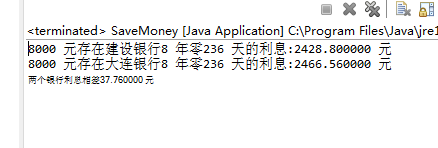


实验 **2** 银行与利息  
程序模板 请按模板要求，将【代码】替换为 Java 程序代码。  
Bank.java  
public class Bank {  
int savedMoney;  
int year;  
double interest;  
double interestRate = 0.29;  
public double computerInterest() {  
interest=year\*interestRate\*savedMoney;  
return interest;  
}  
public void setInterestRate(double rate) {  
interestRate = rate;  
}  
}  
**ConstructionBank.java**public class ConstructionBank extends Bank {  
double year;  
public double computerInterest() {  
super.year=(int)year;  
double r = year-(int)year;  
int day=(int)(r\*1000);  
double yearInterest = 【代码 1】 //super 调用隐藏的 computerInterest()方法  
double dayInterest = day\*0.0001\*savedMoney;  
interest= yearInterest+dayInterest;  
System.out.printf("%d 元存在建设银行%d 年零%d 天的利息:%f 元\n",  
savedMoney,super.year,day,interest);  
return interest;  
}  
}  
**BankOfDalian.java**public class BankOfDalian extends Bank {  
double year;  
public double computerInterest() {  
super.year=(int)year;  
double r = year-(int)year;  
int day=(int)(r\*1000);  
double yearInterest = 【代码 2】 // super 调用隐藏的 computerInterest()方法  
double dayInterest = day\*0.00012\*savedMoney;  
interest= yearInterest+dayInterest;  
System.out.printf("%d 元存在大连银行%d 年零%d 天的利息:%f 元\n",  
savedMoney,super.year,day,interest);  
return interest;  
}  
}  
**SaveMoney.java**public class SaveMoney {  
public static void main(String args[]) {  
int amount=8000;  
ConstructionBank bank1 = new ConstructionBank();  
bank1.savedMoney = amount;  
bank1.year = 8.236;  
bank1.setInterestRate(0.035);  
double interest1 = bank1.computerInterest();  
BankOfDalian bank2 = new BankOfDalian();  
bank2.savedMoney = amount;  
bank2.year = 8.236;  
bank2.setInterestRate(0.035);  
double interest2=bank2.computerInterest();  
System.out.printf("两个银行利息相差%f 元\n",interest2-interest1);  
}  
}

实验代码截图：

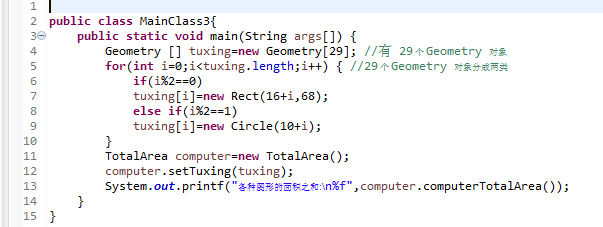
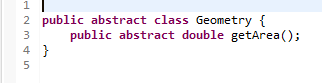
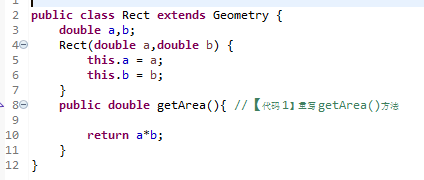
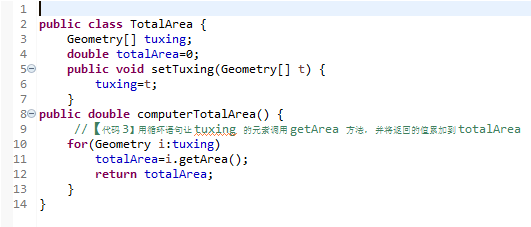
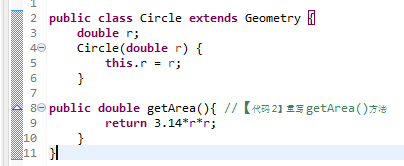


实验结果截图：

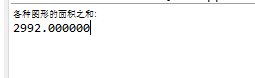


实验 **3** 面积之和  
程序模板 请按模板要求，将【代码】替换为 Java 程序代码。  
**Geometry.java**public abstract class Geometry {  
public abstract double getArea();  
}  
**TotalArea.java**public class TotalArea {  
Geometry[] tuxing;  
double totalArea=0;  
public void setTuxing(Geometry[] t) {  
tuxing=t;  
}  
public double computerTotalArea() {  
【代码 3】 //用循环语句让 tuxing 的元素调用 getArea 方法， 并将返回的值累加到 totalArea  
return totalArea;  
}  
}  
**Rect.java**public class Rect extends Geometry {  
double a,b;  
Rect(double a,double b) {  
this.a = a;  
this.b = b;  
}  
【代码 1】 //重写 getArea()方法  
}  
**Circle.java**public class Circle extends Geometry {  
18  
double r;  
Circle(double r) {  
this.r = r;  
}  
【代码 2】 //重写 getArea()方法  
}  
**MainClass.java**public class MainClass{  
public static void main(String args[]) {  
Geometry [] tuxing=new Geometry[29]; //有 29 个 Geometry 对象  
for(int i=0;i<tuxing.length;i++) { //29 个 Geometry 对象分成两类  
if(i%2==0)  
tuxing[i]=new Rect(16+i,68);  
else if(i%2==1)  
tuxing[i]=new Circle(10+i);  
}  
TotalArea computer=new TotalArea();  
computer.setTuxing(tuxing);  
System.out.printf("各种图形的面积之和:\n%f",computer.computerTotalArea());  
}  
}

实验代码截图：

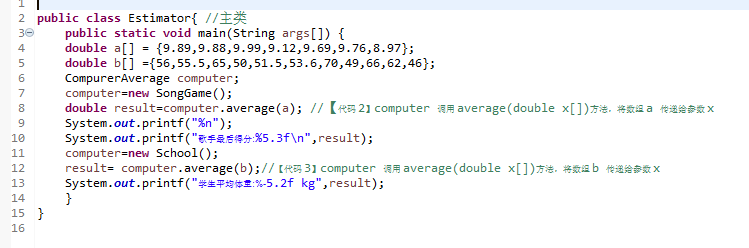
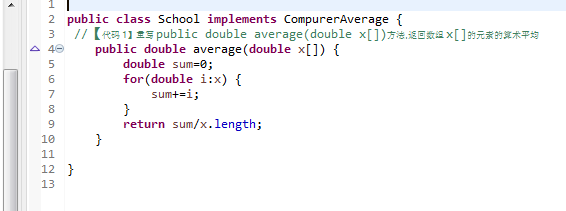
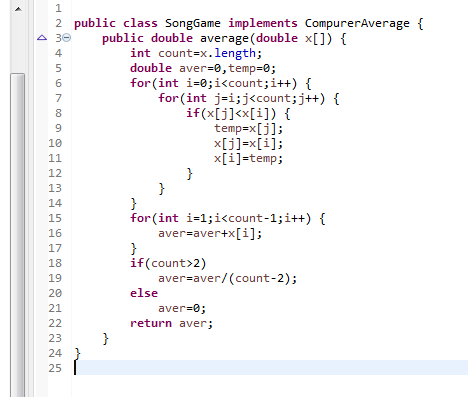
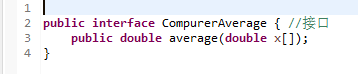


实验结果截图：

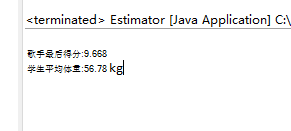


实验 **4** 歌手大赛  
程序模板 请按模板要求，将【代码】替换为 Java 程序代码。  
**CompurerAverage.java**public interface CompurerAverage { //接口  
public double average(double x[]);  
}  
**SongGame.java**public class SongGame implements CompurerAverage {  
public double average(double x[]) {  
int count=x.length;  
double aver=0,temp=0;  
for(int i=0;i<count;i++) {  
for(int j=i;j<count;j++) {  
if(x[j]<x[i]) {  
temp=x[j];  
x[j]=x[i];  
x[i]=temp;  
}  
}  
}  
for(int i=1;i<count-1;i++) {  
aver=aver+x[i];  
  
}  
if(count>2)  
aver=aver/(count-2);  
else  
aver=0;  
return aver;  
}  
}  
**School.java**public class School implements CompurerAverage {  
【代码 1】 //重写 public double average(double x[])方法,返回数组 x[]的元素的算术平均  
}  
**Estimator.java**public class Estimator{ //主类  
public static void main(String args[]) {  
double a[] = {9.89,9.88,9.99,9.12,9.69,9.76,8.97};  
double b[] ={56,55.5,65,50,51.5,53.6,70,49,66,62,46};  
CompurerAverage computer;  
computer=new SongGame();  
double result=【代码 2】 //computer 调用 average(double x[])方法，将数组 a 传递给参数 x  
System.out.printf("%n");  
System.out.printf("歌手最后得分:%5.3f\n",result);  
computer=new School();  
result=【代码 3】 //computer 调用 average(double x[])方法，将数组 b 传递给参数 x  
System.out.printf("学生平均体重:%-5.2f kg",result);  
}  
}

实验代码截图：

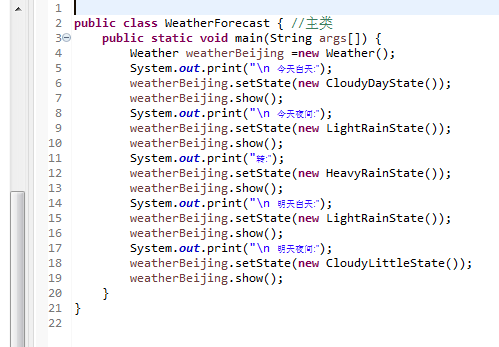
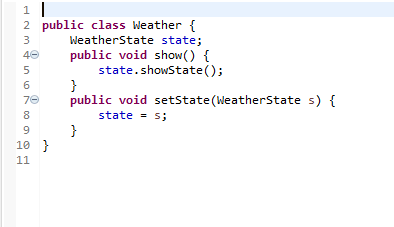
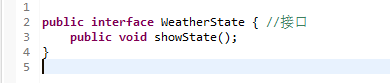


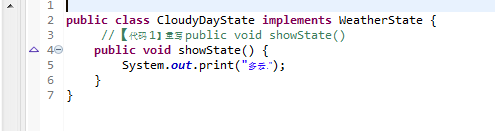
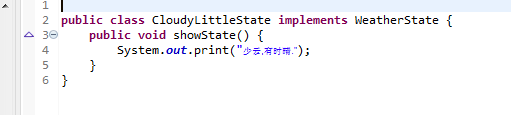
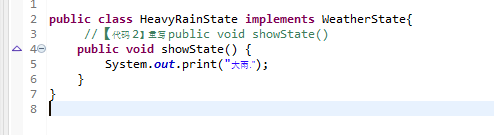
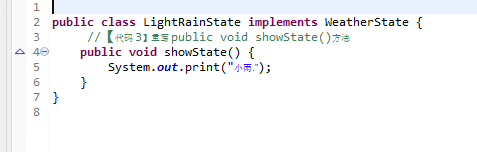
实验结果截图：



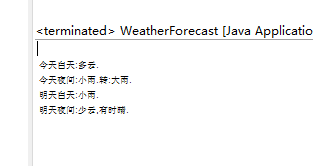
实验 **5** 天气预报  
程序模板 请按模板要求，将【代码】替换为 Java 程序代码。  
**WeatherState.java**public interface WeatherState { //接口  
public void showState();  
}  
**Weather.java**public class Weather {  
WeatherState state;  
public void show() {  
state.showState();  
}  
public void setState(WeatherState s) {  
state = s;  
}  
}  
**WeatherForecast.java**public class WeatherForecast { //主类  
public static void main(String args[]) {  
Weather weatherBeijing =new Weather();  
System.out.print("\n 今天白天:");  
weatherBeijing.setState(new CloudyDayState());  
weatherBeijing.show();  
System.out.print("\n 今天夜间:");  
weatherBeijing.setState(new LightRainState());  
weatherBeijing.show();  
System.out.print("转:");  
weatherBeijing.setState(new HeavyRainState());  
weatherBeijing.show();  
System.out.print("\n 明天白天:");  
weatherBeijing.setState(new LightRainState());  
weatherBeijing.show();  
System.out.print("\n 明天夜间:");  
weatherBeijing.setState(new CloudyLittleState());  
weatherBeijing.show();  
}  
}  
**CloudyLittleState.java**public class CloudyLittleState implements WeatherState {  
public void showState() {  
System.out.print("少云,有时晴.");  
}  
}  
**CloudyDayState.java**public class CloudyDayState implements WeatherState {  
【代码 1】 //重写 public void showState()  
}  
**HeavyRainState.java**public class HeavyRainState implements WeatherState{  
【代码 2】 //重写 public void showState()  
}  
**LightRainState.java**public class LightRainState implements WeatherState {  
【代码 3】 //重写 public void showState()方法  
}

实验代码截图





实验结果截图



实验5 利用接口做参数，写计算器，完成加减乘除运算。示例：

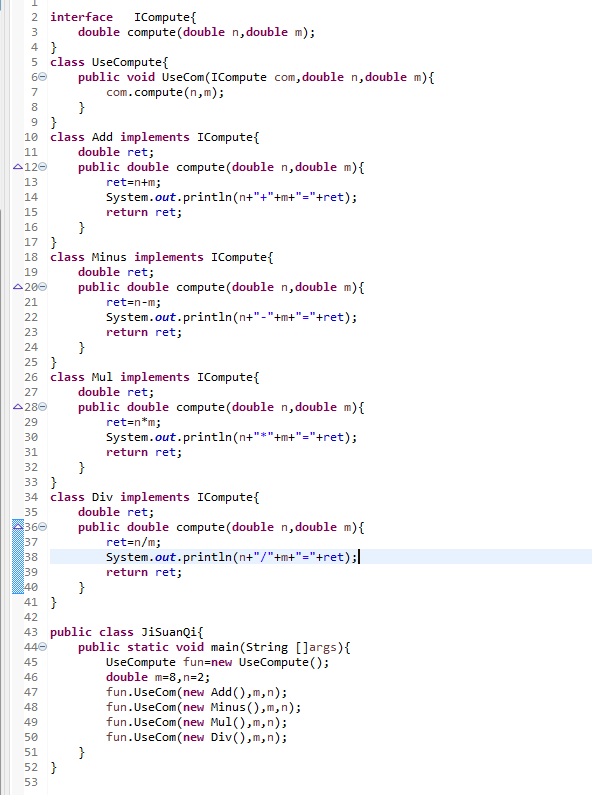
（1）定义接口Compute含有方法int computer(int n, int m)

（2）设计四个类分别实现此接口，完成加减乘除运算

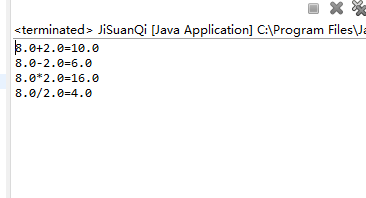
（3）设计类UseCompute，类中含方法：useCom(Compute com, int one, int two)，此方法能用传递来的对象调用computer方法完成运算，输出运算结果

（4）设计主类Test，调用UseCompute的方法useCom来完成加减乘除运算。

实验程序截图



实验结果截图



**四、实验心得及体会**

难得学校安排了实训，我利用机会给自己巩固知识、加深开发经验，使理论与实践达到最完美的相结合。另外，这次实训也让我明白了学习是一个长时间慢慢积累的过程，我想在以后的工作、生活中都应该不断的学习，努力提高自己的知识与综合素质。

此外，还要感谢指导老师对我们的细心教化，感谢老师给我们的帮助。在设计过程中，我懂得了如何查阅资料，如何与同学交流和自学，使我充分体会到了在创造过程中探索的艰难和成功时的兴奋。

人非生而知之，要学得知识，一靠学习，二靠实践。没有实践，学习就是无源之水，无本之木。以上就是我这次实训的心得和感受。

最后，通过本次实训使我学到很多东西，便于今后让我提早进入工作状态。因为我知道无论是今后的学习还是日后的工作生活中，我会清楚的知道自己想要做什么、如何做和怎样才能把它做到最好。