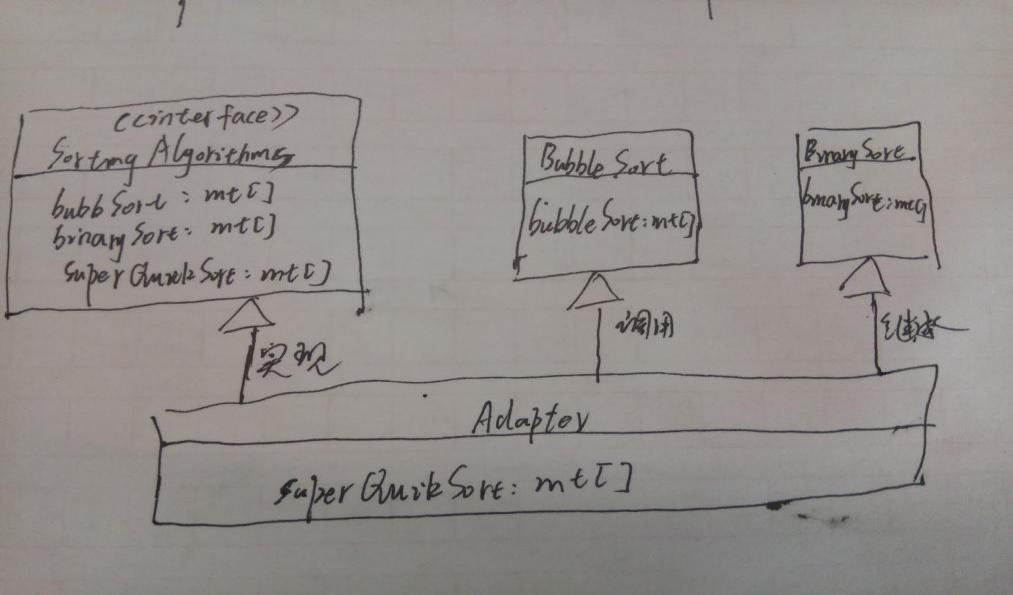
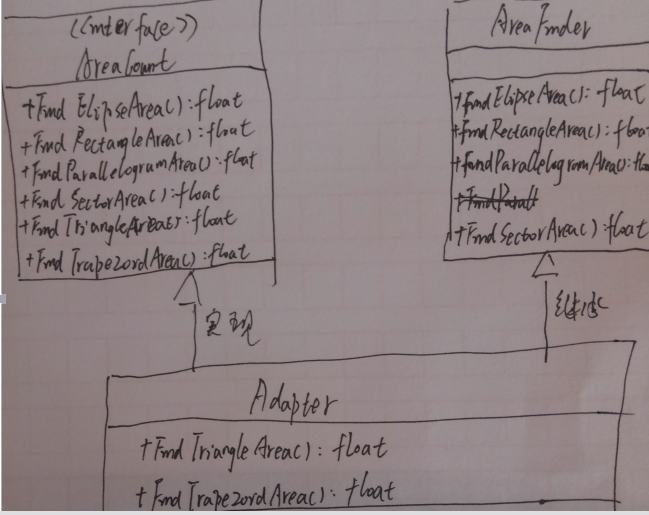
75页

1. （1）①Java不支持多继承，设计中采用了多继承。

②可以改变其中一个继承为调用即可

③

1. ①类适配器



③上述的Adapter即为Client类，可以采用上面的类适配器方法，在Adapter中实现动态的绑定（同时修改接口AreaCount与Adapter）。

76页

3.2-1

1. **Describe your finished homework, including** 
   * 1. **What method has been added to what class?**

public abstract boolean isValidEmailAddr(String EmailAddr);增加到CusInfoValidator接口中；

public boolean isValidSSNNum(String SSNNum)增加到类InformationAdapter中(实现接口中的抽象方法)。

public String getEmailAddress() {

return txtEmail.getText();

}

增加到AdapterTestGUI类中（获取新增的Email文本框便于判断）；

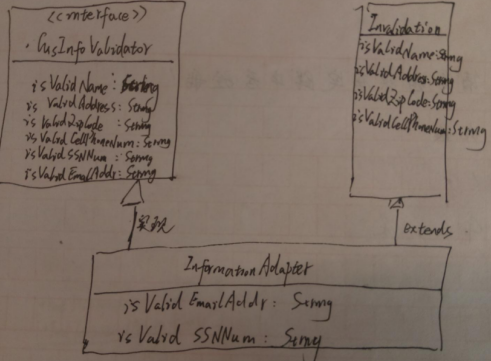
* + 1. **In order to get all the functionalities in the interface CusInfoValidator, in class InformationAdapter, why we only need to claim and implement the methods isValidSSNNum() and isValidEmailAddr?**

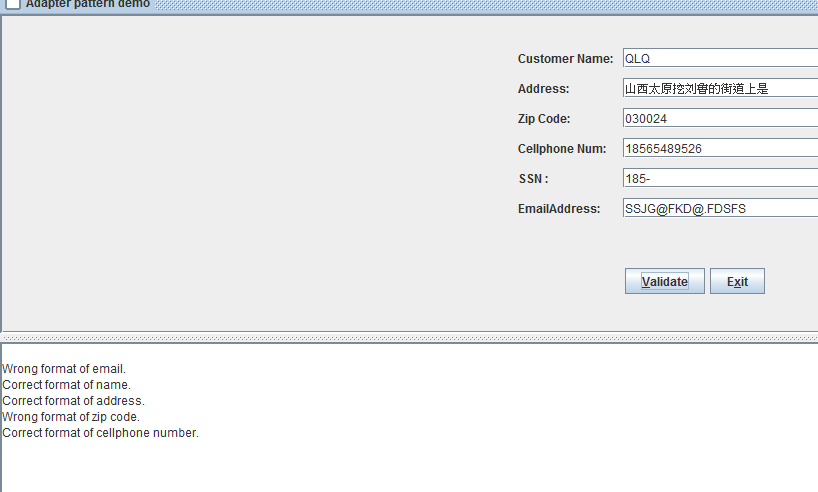
因为InformationAdapter的父类InfoValidation已经实现了除这两个方法之外的其他方法。

* + 1. **You must have noticed that this program is designed and implemented by using *class adapter pattern*. If the design is changed so such that the *object adapter pattern* is used, then in class InformationAdapter, do we need to claim all the methods inside the interface CusInfoValidator?**

不需要，只需要实现其接口定义的且其父类未实现的方法。

1. **Draw your modified class diagram here**



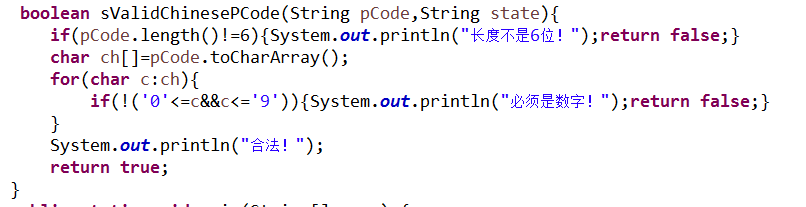
**3.(Program Testing) List your typical input and output from running your program** 

3.2-2

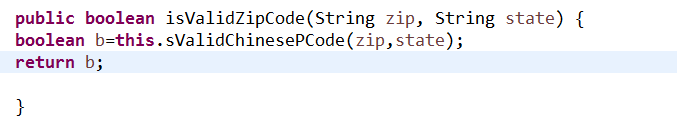
1.学生对自己已经完成的作业的描述（你具体怎样完成了作业？例如，在原来的类图中，你新增加了哪些类？新增加的类与原来的类之间有什么样的关系？你在哪个类中增添了什么代码？这些新添加的类与原来的类有什么关系？等等）

增加了两个类ChinesePostalCode与ChinesePostalCodeAdapter；类ChinesePostalCodeAdapter继承ChinesePostalCode实现ZipCodeValidator接口；

ChinesePostalCode增加的判断编码是否合法代码，第一个参数是邮编，第二个参数代表国家：



ChinesePostalCodeAdapter中实现接口中的抽象方法(调用从父类继承的方法)



2.类关系图（你所设计的类关系图）

**ZipCodeTester**

isValidZipCode(zip:String):

boolean

**USAZipCode**

isValidCanadianPCode

(pcode: String): boolean

**CanadianPostalCode**

**CAPostalCodeAdapter**

isValidZipCode(zip:String):

boolean

isValidZipCode(zip:String):

boolean

**<<interface>>**

**ZipCodeValidater**

**Customer**

isValidChinesePCode

(pcode: String): boolean

**ChinesePostalCode**

**ChinesePostalCodeAdapter**

isValidZipCode(zip:String): boolean

1. 关键代码，关键参数的传递 （指出你的实现所规定的功能时所用的关键代码）

public class ChinesePostalCodeAdapter extends ChinesePostalCode implements ZipCodeValidator {

public boolean isValidZipCode(String zip, String state) {

boolean b=this.sValidChinesePCode(zip,state);

return b;

}

}

**public** **class** ChinesePostalCode {

boolean sValidChinesePCode(String pCode,String state){

if(pCode.length()!=6){System.out.println("长度不是6位！");return false;}

char ch[]=pCode.toCharArray();

for(char c:ch){

if(!('0'<=c&&c<='9')){System.out.println("必须是数字！");return false;}

}

System.out.println("合法！");

return true;

}

//测试函数

public static void main(String[] args) {

// TODO Auto-generated method stub

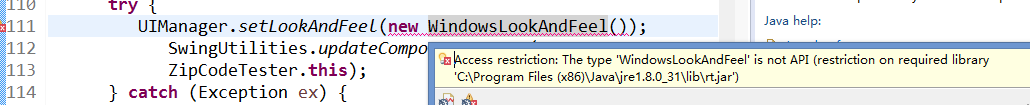
ChinesePostalCode ss=new ChinesePostalCode();

ss.sValidChinesePCode("510048","中国");

}

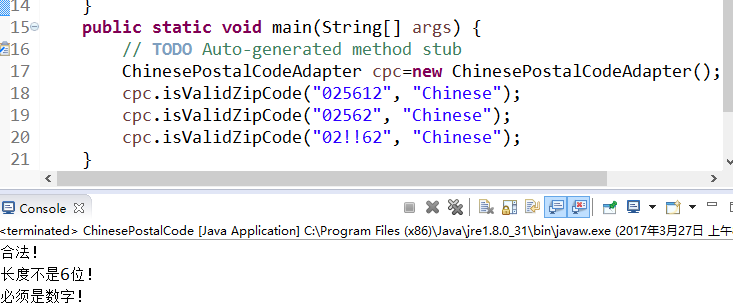
}

1. 其它说明



在调试时一个类访问被拒绝。因此只在控制台测试了自己添加的功能。

1. 典型输入与输出



89页

3.3-1

**1.Describe your finished homework, including**

* + 1. **What classes have been added to the existing class hierarchy?**

StudentTuitionModel与Tuition类

* + 1. **The relationship between the existing classes and the newly added classes.**

Tuition调用StudentTuitionModel；StudentInfoFacade调用Tuition类。

* + 1. **What code has been added to what class?**

(1)StudentInfoFacade中新加的代码：（输出信息时输出查询的学费信息：）



allInfo = "\nBasic Student Info: \n"+info+"\nAcademic record: \n"+record+"\nAwards: \n"+awards+"\nTuition: \n"+tuition;

(2)Tuition类中新加的代码

public ArrayList<StudentTuitionModel> getAllTution(String file){

aFile = file;

allTuition = new ArrayList();

try {

BufferedReader reader = new BufferedReader(new FileReader(aFile));

String line = reader.readLine();

while(line != null) {

if (line.length() != 0) {

String[] arr = line.split("\\,");

StudentTuitionModel sdam = new StudentTuitionModel (arr[0].trim(), arr[1].trim(),arr[2].trim(),arr[3].trim(), arr[4].trim());

if( sdam.getStudFirstName().equals(firstName) &&

sdam.getStudLastName().equals(lastName) &&

sdam.getStudSerialNum().equals(studNum) )

allTuition.add(sdam);

}

line = reader.readLine();

}

}

catch(FileNotFoundException exc){

exc.printStackTrace();

System.exit(1);

}

catch(IOException exc){

exc.printStackTrace();

System.exit(1);

}

return allTuition;

}

**2.Original class diagram, and new class diagram**

**原来的：**

**StudentInfoFacade**

+extractStudentInfo(): String

+extractAcademicRecord(): String

+extractAllAwards(): String

+getStudentInfo(): String

- firstName: String;

- lastName: String;

- studNum: String;

**StudentBasicInfo**

+isExistingStudentName(String name)

+getStudentBasicInfo(): ArrayList

- name: String;

- birthDate: String;

- serialNum: String;

**AcademicRecord**

+getAllScores(String file): ArrayList

- firstName: String;

- lastName: String;

- studNum: String;

**EllipseDrawerGUI**

**Award**

+getAllAwards(String file): ArrayList

- firstName: String;

- lastName: String;

- studNum: String;

**StudentAwardModel**

+getAllAwards(String file): ArrayList

- studFirstName: String;

- studLastName: String;

- studSerialNum: String;

- awardName: String;

- awardDate: String

**StudentAcademicModel**

+getAllAwards(String file): ArrayList

- studFirstName: String;

- studLastName: String;

- studSerialNum: String;

- courseTitle: String;

- courseNum: String;

- score: String;

**StudentBasicInfoModel**

+getAllAwards(String file): ArrayList

- name: String;

- birthDate: String;

- serialNumber: String;

- ssNumber: String;

- major: String;

- degree: String;

+getStudFirstName(): String

+getStudLastName(): String

+getStudSerialNum(): String

+getAwardName(): String

+ getAwardDate(): String

+getStudFirstName(): String

+getStudLastName(): String

+getStudSerialNum(): String

+getCourseTitle(): String

+getCourseNum(): String

+getScore(): String

+getName(): String

+getBirthDate(): String

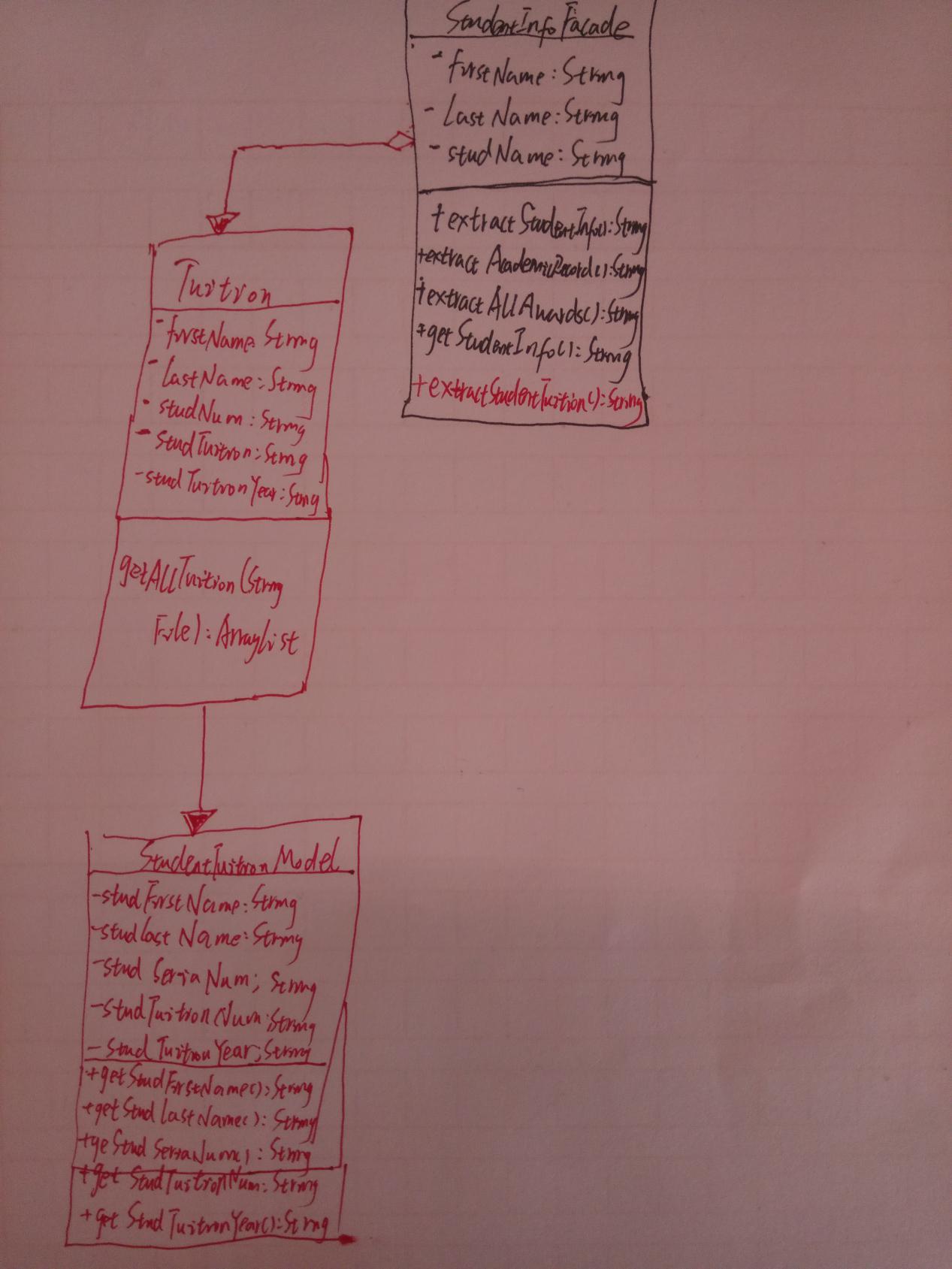
+getSerialNum(): String

+getSocialSecurityNum(): String

+getMajor(): String

+getDegree(): String

**修改后的：红色为在上面类图基础上增加**

****

**3.Your new key code, and your new key parameters in order to finish the required new functionalities**

**StudengTuitionModel中代码：**

public class StudentTuitionModel {

private String studFirstName;

private String studLastName;

private String studSerialNum;

private String studTuitionNum;

private String studTuitionYear;

public StudentTuitionModel(String studFirstName, String studLastName, String studSerialNum, String studTuitionNum,

String studTuitionYear) {

super();

this.studFirstName = studFirstName;

this.studLastName = studLastName;

this.studSerialNum = studSerialNum;

this.studTuitionNum = studTuitionNum;

this.studTuitionYear = studTuitionYear;

}

public String getStudFirstName() {

return studFirstName;

}

public String getStudLastName() {

return studLastName;

}

public String getStudSerialNum() {

return studSerialNum;

}

public String getStudTuitionNum() {

return studTuitionNum;

}

public String getStudTuitionYear() {

return studTuitionYear;

}

}

StudentBasicInfoModel类中添加的提取缴费信息代码：

public String extractTuitionInfo(){

String allInfo = " ";

tuition = new Tuition(firstName, lastName, studentNum);

ArrayList<StudentTuitionModel> tui = tuition.getAllTution(TUITION);

Iterator tuitionIterator = tui.iterator();

while (tuitionIterator.hasNext()) {

StudentTuitionModel tuitionModel = (StudentTuitionModel) tuitionIterator.next();

String firstNm = tuitionModel. getStudFirstName().trim();

String lastNm = tuitionModel. getStudLastName().trim();

String studNum = tuitionModel. getStudSerialNum().trim();

String tuiNum = tuitionModel.getStudTuitionNum().trim();

String tuiDate = tuitionModel.getStudTuitionYear().trim();

allInfo = allInfo +" " + firstNm+" "+ lastNm+ " " + studNum+ " "+ tuiNum+" "+tuiDate+"\n";

}

return allInfo;

}

}

**4.Typical input and output from running your finished program**

