DAO

The Data Access Object pattern is mechanism to abstract away the details of persistence in an application. Application uses, the domain logic speaks to a DAO layer instead. This DAO layer then communicates with the underlying persistence system or service.

I have 3 packages: model, ui, dao.

Model: Student.java contains fields (id, first name, last name) and getter, setter methods.

Dao: StudentDAO.java gets connection to database. Contains functions for getting all students from table, updating student and deleting student using mysql statements.

```
public void updateStudent(Student theStudent)

public void deleteStudent(int stId)

public void addStudent(Student theStudent)

public List<Student> getAllStudents()
```

ui: StudentsTable.java, StudentsAppTable.java, StudentDialog.java, AddStudentDialog.java

StudentsTable.java used to for getting values from table on exact row and column.

```
public Object getValueAt(int row, int col)
```

StudentsAppTable.java is the main GUI that contains table, buttons. Here is implemented functions that recalls StudentDialog.java for updating and AddStudentDialog.java for adding.

StudentDialog.java is dialog that appears after pushing “update” button, used for updating dialog. Here are functions that gets values from textfields then updates row in the table on main GUI.

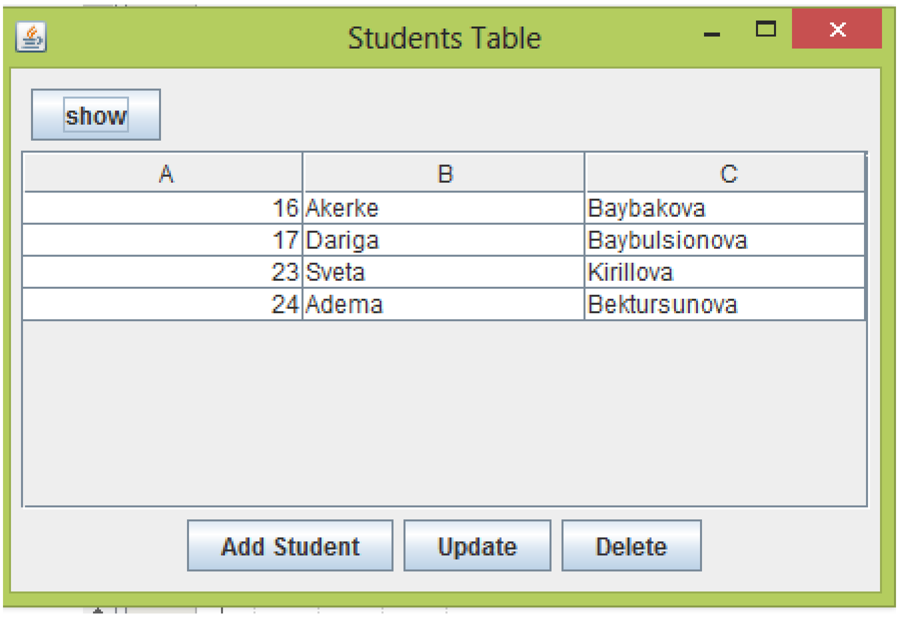
```
protected void saveStudent(){

        String firstName = firstNameTextField.getText();

        String lastName = lastNameTextField.getText();

if (updateMode) {

        tempStudent = prevStudent;

        tempStudent.setFirstName(firstName);

        tempStudent.setLastName(lastName);


        }….}
```

AddStudentDialog.java is dialog that appears after pushing “add” button. Here are functions that gets values from textfields then adds new row in the table on main GUI.

```
protected void saveStudent() {

        String firstName = firstNameTextField.getText();
        String lastName = lastNameTextField.getText();
        Student tempStudent = new Student(firstName,lastName);
        try {
            dao.addStudent(tempStudent);
            setVisible(false);
```

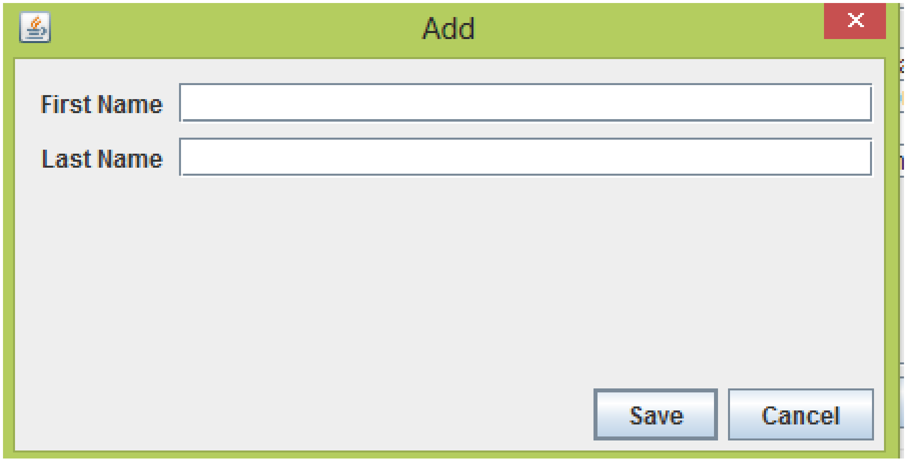
```
                    dispose();

                    studentsTabApp.refreshStudentsView(); }
```

The main GUI. Table with students appears after pushing “show” button.

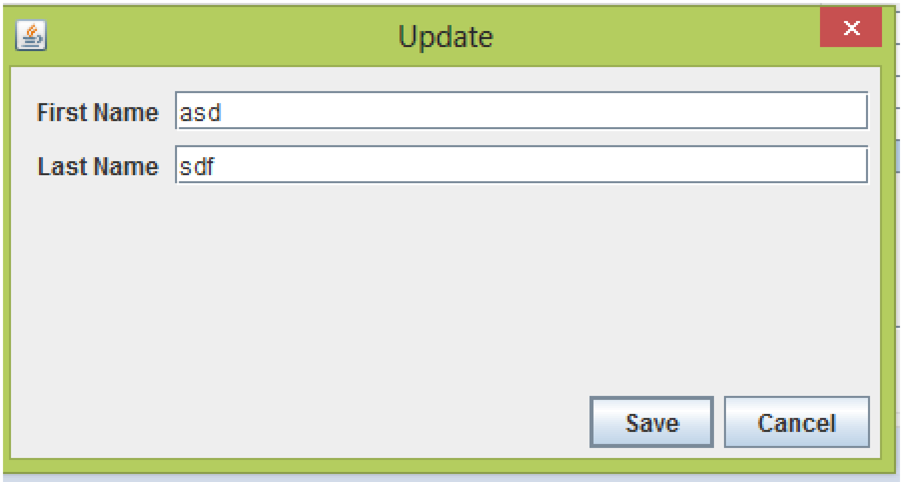


A	B	C
16	Akerke	Baybakova
17	Dariga	Baybulsionova
23	Sveta	Kirillova
24	Adema	Bektursunova

Dialog for adding new student



Dialog for updating.



First Name asd

Last Name sdf