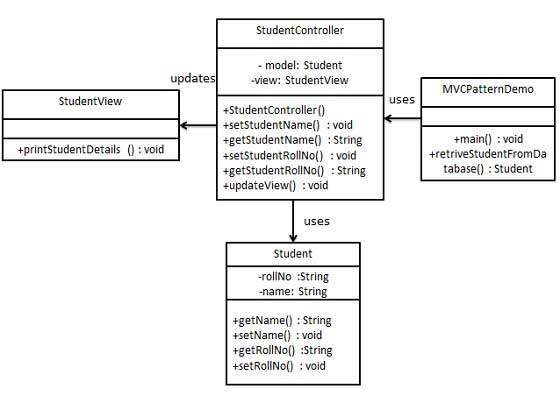
MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.

* Model - Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
* View - View represents the visualization of the data that model contains.
* Controller - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

## Implementation

We are going to create a *Student* object acting as a model.*StudentView* will be a view class which can print student details on console and *StudentController* is the controller class responsible to store data in *Student* object and update view *StudentView*accordingly.

*MVCPatternDemo*, our demo class, will use *StudentController* to demonstrate use of MVC pattern.



/\*

\* create model

\* Student.java

\*/

class Student{

private String rollNo;

private String name;

public String getRollNo() {

return rollNo;

}

public void setRollNo(String rollNo) {

this.rollNo = rollNo;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

/\*

\* create view to display

\* StudentView.java

\*/

class StudentView{

public void printStudentDetails(String studentName, String studentRoll) {

System.out.println("Student deatils");

System.out.println("Name : "+ studentName);

System.out.println("Roll No : "+ studentRoll);

}

}

/\*

\* create controller

\* StudentController.java

\*/

class StudentController{

private Student model;

private StudentView studentView;

public StudentController(Student model, StudentView studentView) {

this.model = model;

this.studentView = studentView;

}

public void setStudentName(String name) {

model.setName(name);

}

public String getStudentName() {

return model.getName();

}

public void setStudentRollNo(String rollNo) {

model.setRollNo(rollNo);

}

public String getStudentRollNo() {

return model.getRollNo();

}

public void updateView() {

studentView.printStudentDetails(model.getName(), model.getRollNo());

}

}

/\*

\* use the studentController methods to demonstrate MVC design pattern usage;

\*/

public class TestMVC {

public static void main(String[] args) {

//fetch student details from db

Student model = retriveStudentDetailFromDB();

//create view : to write student details on console

StudentView studentView = new StudentView();

StudentController studentController = new StudentController(model, studentView);

studentController.updateView();

//update model data

studentController.setStudentName("vinay");

studentController.updateView();

}

public static Student retriveStudentDetailFromDB() {

Student model = new Student();

model.setName("rajat");

model.setRollNo("1245");

return model;

}

}