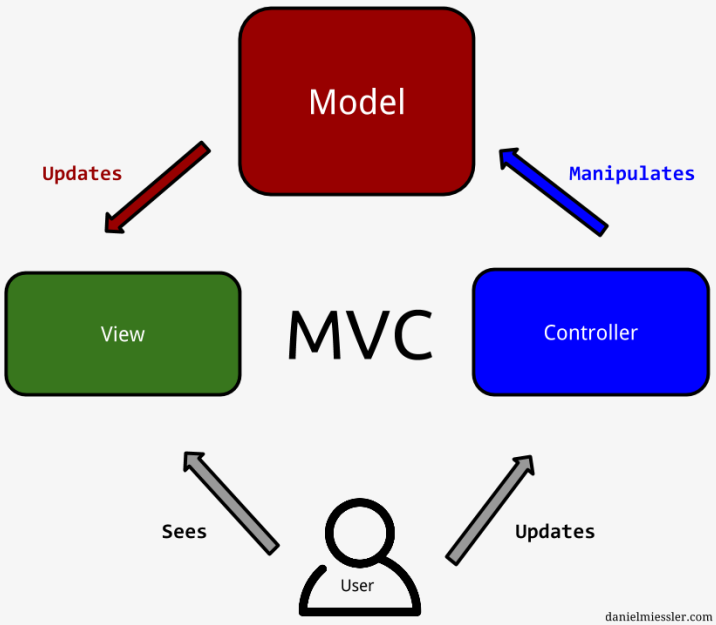
访问者模式 业务代表模式

MVC 模式

MVC 模式代表 Model-View-Controller（模型-视图-控制器）模式。这种模式用于应用程序的分层开发。

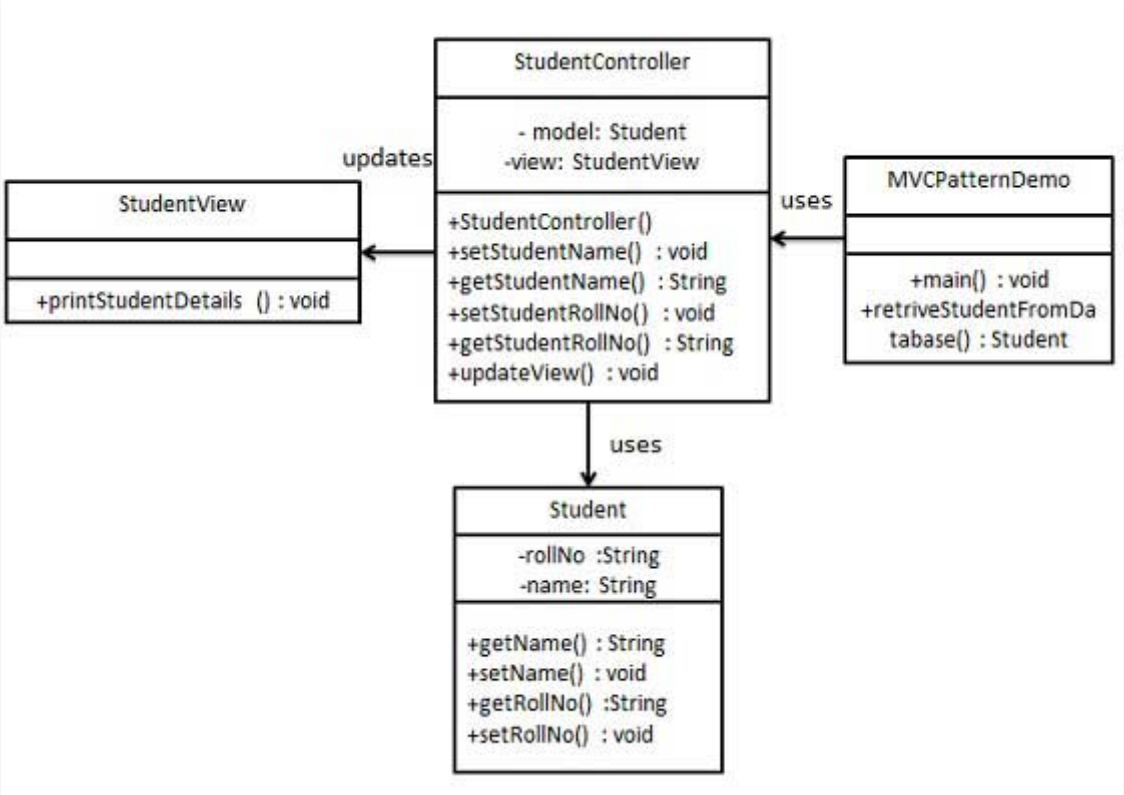
- **Model（模型）** - 模型代表一个存取数据的对象或 JAVA POJO。它也可以带有逻辑，在数据变化时更新控制器。
- **View（视图）** - 视图代表模型包含的数据的可视化。
- **Controller（控制器）** - 控制器作用于模型和视图上。它控制数据流向模型对象，并在数据变化时更新视图。它使视图与模型分离开。



实现

我们将创建一个作为模型的 *Student* 对象。*StudentView* 是一个把学生详细信息输出到控制台的视图类，*StudentController* 是负责存储数据到 *Student* 对象中的控制器类，并相应地更新视图 *StudentView*。

MVCPatternDemo，我们的演示类使用 *StudentController* 来演示 MVC 模式的用法。



步骤 1

创建模型。

Student.java

```
public 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;
}
}
```

步骤 2

创建视图。

StudentView.java

```
public class StudentView {
public void printStudentDetails(String studentName, String studentRollNo){
System.out.println("Student: ");
```

```
System.out.println("Name: " + studentName);
System.out.println("Roll No: " + studentRollNo);
}
}
```

步骤 3

创建控制器。

StudentController.java

```
public class StudentController {
private Student model;
private StudentView view;
public StudentController(Student model, StudentView view){
this.model = model;
this.view = view;
}
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(){
view.printStudentDetails(model.getName(), model.getRollNo());
}
}
```

步骤 4

使用 *StudentController* 方法来演示 MVC 设计模式的用法。

MVCPatternDemo.java

```
public class MVCPatternDemo {
public static void main(String[] args) {
//从数据库获取学生记录
Student model = retriveStudentFromDatabase();
//创建一个视图：把学生详细信息输出到控制台
StudentView view = new StudentView();
StudentController controller = new StudentController(model, view);
controller.updateView();
//更新模型数据
controller.setStudentName("John");
controller.updateView();
}
private static Student retriveStudentFromDatabase(){
```

```
Student student = new Student();
student.setName("Robert");
student.setRollNo("10");
return student;
}
}
```

步骤 5

执行程序，输出结果：

```
Student:
Name: Robert
Roll No: 10
Student:
Name: John
Roll No: 10
```

← 访问者模式 业务代表模式 →