数据访问对象模式 拦截过滤器模式

前端控制器模式

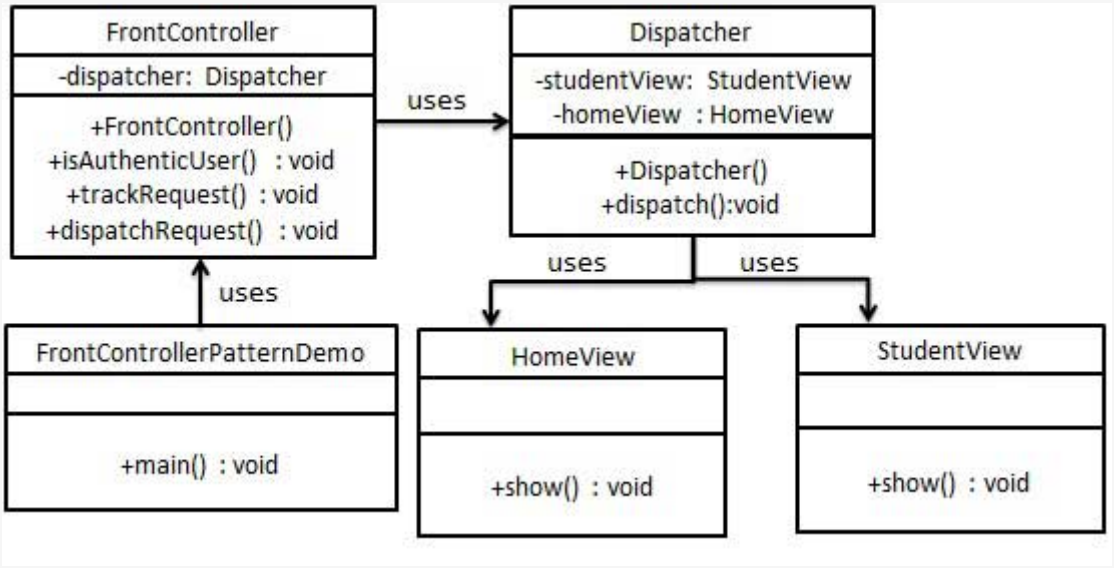
前端控制器模式（Front Controller Pattern）是用来提供一个集中的请求处理机制，所有的请求都将由一个单一的处理程序处理。该处理程序可以做认证/授权/记录日志，或者跟踪请求，然后把请求传给相应的处理程序。以下是这种设计模式的实体。

- **前端控制器（Front Controller）** - 处理应用程序所有类型请求的单个处理程序，应用程序可以是基于 web 的应用程序，也可以是基于桌面的应用程序。
- **调度器（Dispatcher）** - 前端控制器可能使用一个调度器对象来调度请求到相应的具体处理程序。
- **视图（View）** - 视图是为请求而创建的对象。

实现

我们将创建 *FrontController*、*Dispatcher* 分别当作前端控制器和调度器。*HomeView* 和 *StudentView* 表示各种为前端控制器接收到的请求而创建的视图。

FrontControllerPatternDemo，我们的演示类使用 *FrontController* 来演示前端控制器设计模式。



步骤 1

创建视图。

HomeView.java

```
public class HomeView {
public void show(){
System.out.println("Displaying Home Page");
}
}
```

StudentView.java

```
public class StudentView {
public void show(){
```

```
System.out.println("Displaying Student Page");
}
}
```

步骤 2

创建调度器 Dispatcher。

Dispatcher.java

```
public class Dispatcher {
private StudentView studentView;
private HomeView homeView;
public Dispatcher(){
studentView = new StudentView();
homeView = new HomeView();
}
public void dispatch(String request){
if(request.equalsIgnoreCase("STUDENT")){
studentView.show();
}else{
homeView.show();
}
}
}
```

步骤 3

创建前端控制器 FrontController。

FrontController.java

```
public class FrontController {
private Dispatcher dispatcher;
public FrontController(){
dispatcher = new Dispatcher();
}
private boolean isAuthenticUser(){
System.out.println("User is authenticated successfully.");
return true;
}
private void trackRequest(String request){
System.out.println("Page requested: " + request);
}
public void dispatchRequest(String request){
//记录每一个请求
trackRequest(request);
//对用户进行身份验证
if(isAuthenticUser()){
dispatcher.dispatch(request);
}
}
}
```

步骤 4

使用 *FrontController* 来演示前端控制器设计模式。

FrontControllerPatternDemo.java

```
public class FrontControllerPatternDemo {
public static void main(String[] args) {
FrontController frontController = new FrontController();
frontController.dispatchRequest("HOME");
frontController.dispatchRequest("STUDENT");
}
}
```

步骤 5

执行程序，输出结果：

```
Page requested: HOME
User is authenticated successfully.
Displaying Home Page
Page requested: STUDENT
User is authenticated successfully.
Displaying Student Page
```

← 数据访问对象模式 拦截过滤器模式 →