服务定位器模式 设计模式资源

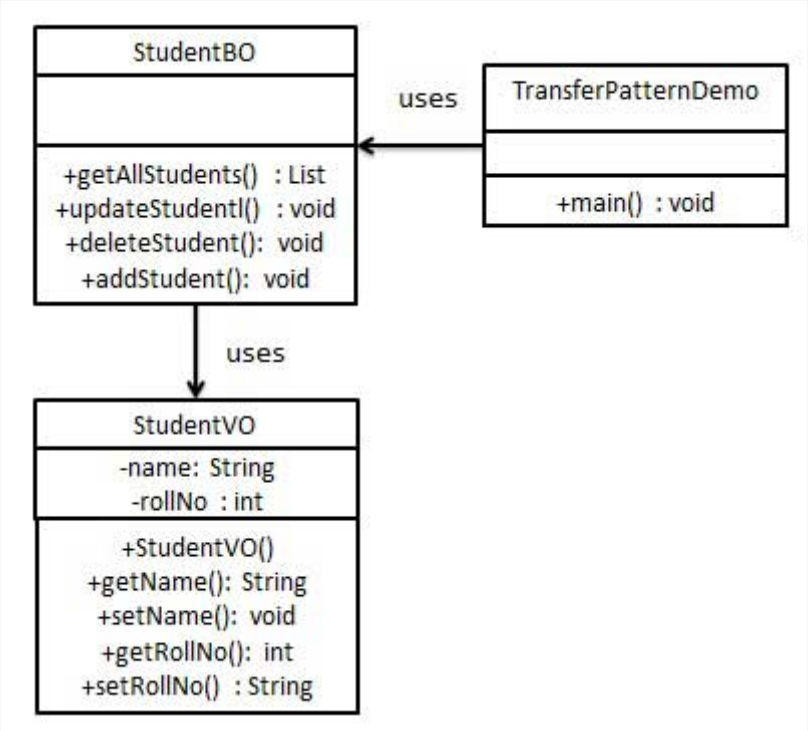
传输对象模式

传输对象模式（Transfer Object Pattern）用于从客户端向服务器一次性传递带有多个属性的数据。传输对象也被称为数值对象。传输对象是一个具有 getter/setter 方法的简单的 POJO 类，它是可序列化的，所以它可以通过网络传输。它没有任何的行为。服务器端的业务类通常从数据库读取数据，然后填充 POJO，并把它发送到客户端或按值传递它。对于客户端，传输对象是只读的。客户端可以创建自己的传输对象，并把它传递给服务器，以便一次性更新数据库中的数值。以下是这种设计模式的实体。

- **业务对象（Business Object）** - 为传输对象填充数据的业务服务。
- **传输对象（Transfer Object）** - 简单的 POJO，只有设置/获取属性的方法。
- **客户端（Client）** - 客户端可以发送请求或者发送传输对象到业务对象。

实现

我们将创建一个作为业务对象的 *StudentBO* 和作为传输对象的 *StudentVO*，它们都代表了我们的实体。*TransferObjectPatternDemo*，我们的演示类在这里是作为一个客户端，将使用 *StudentBO* 和 *Student* 来演示传输对象设计模式。



步骤 1

创建传输对象。

StudentVO.java

```
public class StudentVO {
private String name;
```

```
private int rollNo;
StudentVO(String name, int rollNo){
this.name = name;
this.rollNo = rollNo;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public int getRollNo() {
return rollNo;
}
public void setRollNo(int rollNo) {
this.rollNo = rollNo;
}
}
```

步骤 2

创建业务对象。

StudentBO.java

```
import java.util.ArrayList;
import java.util.List;
public class StudentBO {
//列表是当作一个数据库
List<StudentVO> students;
public StudentBO(){
students = new ArrayList<StudentVO>();
StudentVO student1 = new StudentVO("Robert",0);
StudentVO student2 = new StudentVO("John",1);
students.add(student1);
students.add(student2);
}
public void deleteStudent(StudentVO student) {
students.remove(student.getRollNo());
System.out.println("Student: Roll No "
+ student.getRollNo() +", deleted from database");
}
//从数据库中检索学生名单
public List<StudentVO> getAllStudents() {
return students;
}
public StudentVO getStudent(int rollNo) {
return students.get(rollNo);
}
public void updateStudent(StudentVO student) {
students.get(student.getRollNo()).setName(student.getName());
System.out.println("Student: Roll No "
+ student.getRollNo() +", updated in the database");
```

```
}
}
```

步骤 3

使用 *StudentBO* 来演示传输对象设计模式。

TransferObjectPatternDemo.java

```
public class TransferObjectPatternDemo {
public static void main(String[] args) {
StudentBO studentBusinessObject = new StudentBO();
//输出所有的学生
for (StudentVO student : studentBusinessObject.getAllStudents()) {
System.out.println("Student: [RollNo : "
+student.getRollNo()+", Name : "+student.getName()+" ]");
}
//更新学生
StudentVO student =studentBusinessObject.getAllStudents().get(0);
student.setName("Michael");
studentBusinessObject.updateStudent(student);
//获取学生
studentBusinessObject.getStudent(0);
System.out.println("Student: [RollNo : "
+student.getRollNo()+", Name : "+student.getName()+" ]");
}
}
```

步骤 4

执行程序，输出结果：

```
Student: [RollNo : 0, Name : Robert ]
Student: [RollNo : 1, Name : John ]
Student: Roll No 0, updated in the database
Student: [RollNo : 0, Name : Michael ]
```

← 服务定位器模式 设计模式资源 →

点我分享笔记