



Контрольная работа - 4 семестр

Уровни программы в модели MVC

Слой модели

В шаблоне проектирования MVC модель представляет собой уровень данных, который определяет бизнес-логику системы, а также представляет состояние приложения. Объекты модели получают и сохраняют состояние модели в базе данных. На этом уровне мы применяем правила к данным, которые в конечном итоге представляют концепции, которыми управляет наше приложение. Теперь давайте создадим модель, используя Course Class.

```
package MyPackage;
public class Course {
    private String CourseName;
    private String CourseId;
    private String CourseCategory;

    public String getId() {
        return CourseId;
    }

    public void setId(String id) {
        this.CourseId = id;
    }

    public String getName() {
        return CourseName;
    }

    public void setName(String name) {
        this.CourseName = name;
    }

    public String getCategory() {
        return CourseCategory;
    }

    public void setCategory(String category) {
        this.CourseCategory = category;
    }
}
```

Код прост для понимания и не требует пояснений. Он состоит из функций, чтобы получить/установить детали Course.

Уровень представления

Этот уровень шаблона проектирования MVC представляет выходные данные приложения или пользовательского интерфейса. Он отображает данные, извлеченные из слоя модели контроллером, и представляет данные пользователю при каждом запросе. Он получает всю необходимую информацию от контроллера и ему не нужно напрямую взаимодействовать с бизнес-уровнем. Давайте создадим представление, используя ClassView Class.

```
package MyPackage;
public class CourseView {
    public void printCourseDetails(String CourseName, String CourseId, String CourseCategory){
        System.out.println("Course Details: ");
        System.out.println("Name: " + CourseName);
        System.out.println("Course ID: " + CourseId);
    }
}
```

```
System.out.println("Course Category: " + CourseCategory);  
}  
}
```

Этот код просто для печати значений на консоль. Далее у нас есть контроллер веб-приложения.

Уровень контроллера

Контроллер похож на интерфейс между моделью и представлением. Он получает пользовательские запросы от уровня представления и обрабатывает их, включая необходимые проверки. Затем запросы отправляются в модель для обработки данных. После обработки данные снова отправляются обратно в контроллер, а затем отображаются в представлении. Давайте создадим ClassContoller Class, который действует как контроллер.

```
package MyPackage;  
public class CourseController {  
    private Course model;  
    private CourseView view;  
    public CourseController(Course model, CourseView view){  
        this.model = model;  
        this.view = view;  
    }  
    public void setCourseName(String name){  
        model.setName(name);  
    }  
    public String getCourseName(){  
        return model.getName();  
    }  
    public void setCourseId(String id){  
        model.setId(id);  
    }  
    public String getCourseId(){  
        return model.getId();  
    }  
    public void setCourseCategory(String category){  
        model.setCategory(category);  
    }  
    public String getCourseCategory(){  
        return model.getCategory();  
    }  
    public void updateView(){  
        view.printCourseDetails(model.getName(), model.getId(), model.getCategory());  
    }  
}
```

Беглый взгляд на код скажет нам, что этот класс контроллера просто отвечает за вызов модели для получения/установки данных и обновления представления на основе этого.

Класс Main

◀ [Объявления](#)

Перейти на...