

# MVC

Java Collections  
6 уровень, 1 лекция

ОТКРЫТА

— Привет, Амиго!

— Привет, Билаабо!

— Ты уже серьезный программист. Поэтому сегодня у нас будет лекция про MVC.

MVC расшифровывается как **Model—View—Controller**. Это принцип построения архитектуры большого приложения, при котором оно разбивается на три части.

Первая часть содержит всю **бизнес-логику приложения**. Такая часть называется **Модель (Model)**. В ней содержится код, который делает все то, для чего приложение создавалось. Эта часть наиболее независимая от остальных.

Вторая часть содержит все, что касается **отображения данных пользователю**. Такая часть называется **Вид (View)**. Именно в ней содержится код, который управляет показом окон, страниц, сообщений и т.д.

Третья часть содержит код, который занимается **обработкой действий пользователя**. Любые действия пользователя, направленные на изменения модели, должны обрабатываться тут. Такая часть называется **Controller**.

Такой подход позволяет независимо делать три вещи: **логику программы (Model)**, **механизм показа всех данных программы пользователю (View)**, **обрабатывать ввод/действия пользователя (Controller)**.

Очень часто у приложения бывает **несколько View**. Это нормально. Вы можете видеть в Excel одни и те же данные в виде чисел и диаграмм. В играх вы можете смотреть на события от лица героя, на карте, со стороны и т.д. Все это – это **разные View** для одной **Model**.

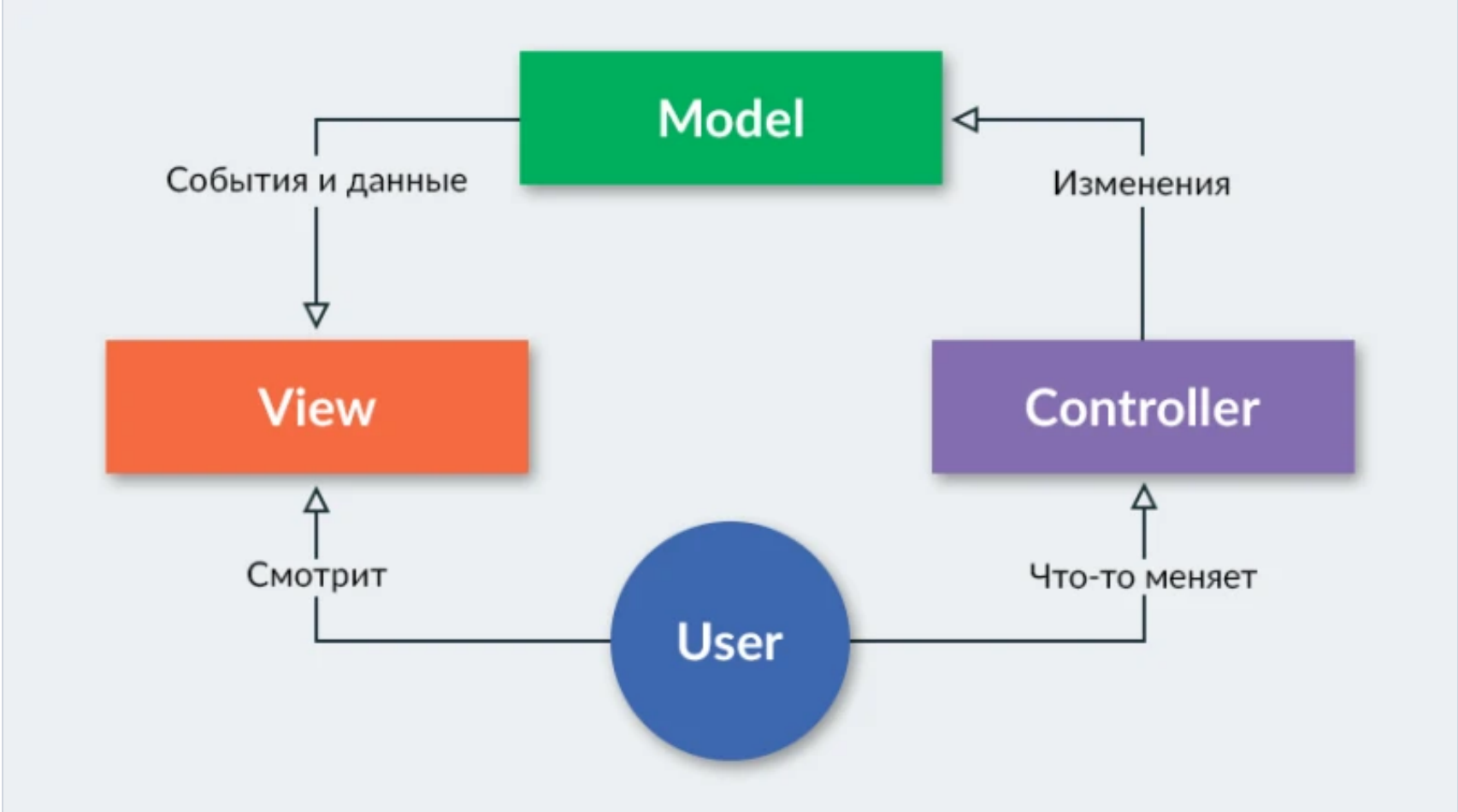
В **Controller’e** сосредоточен весь код, который принимает решение, что менять в модели в ответ на действия пользователя. Например, пользователь принял решение закрыть программу, тогда надо сохранить данные модели в файл на диск. Или пользователь ввел новые данные. Тогда надо добавить их в модель, а модель потом уведомит все View об изменении данных, чтобы они отображали только актуальное их состояние.

— Еще раз.

— С точки зрения Java-программиста, можно сказать, что Model, View, Controller – это три группы классов, у которых:

- а)** у каждой части есть свое назначение;
- б)** связи между классами одной группы довольно сильные;
- в)** связи между группами довольно слабые;
- г)** способы взаимодействия частей довольно сильно регламентированы.

И вот тебе еще такая картинка:



Модель – это **самая независимая часть системы**. Она не зависит от View & Controller. Модель не может использовать классы из разделов View & Controller(!).

Основное ограничение вида (View) – **вид не может менять модель**. Классы вида могут обращаться к модели за данными или подписываться на события, но менять модель классы View не могут.

Основное ограничение контроллера – он **не занимается отображением данных**. Контроллер обрабатывает действия пользователя и меняет в соответствии с ними модель.

— А зачем это мне?

— То, что ты это не применяешь сейчас, еще не значит, что не будешь применять в ближайшем будущем. Ты же тут учишься для того, чтобы работать пойти. И даже если эти знания не пригодятся тебе во время обучения, они точно пригодятся тебе во время работы.

А ведь тебя ждут еще реальные проекты и собеседования...

Мы тут сейчас с тобой разговариваем, а может через месяц ты уже работать будешь.

— Ты абсолютно прав, Билаабо. Я тебя внимательно слушаю.

— Подход MVC в архитектуре приложений встречается довольно часто. Ты должен это знать, чтобы вдруг не начать добавлять классы для View в Model, т.к. так удобнее.

Самое главное в любом проекте – это его архитектура. **Твоя задача на нынешнем этапе — не столько сделать хорошую архитектуру, сколько научиться разбираться в чужой**. До своей ты еще потом пару лет расти будешь. А чужую понимать надо. Сразу.

Когда в приложении используется стандартная архитектура, все становится гораздо понятнее. Зная архитектуру, ты знаешь, что где лежит, как все взаимодействует, как примерно программа работает, где добавить нужный тебе класс и где искать причину ошибки.

[← Предыдущая лекция](#)

×26

Но, вот если ты не знаком со стандартными подходами к архитектуре, то тебе даже самая лучшая архитектура ни о чем не скажет. Будешь смотреть, как крестьянин 15 века на новый автомобиль. Стандартный автомобиль.

– +73 +

НАЧАТЬ ОБУЧЕНИЕ

Введите текст комментария

Igor Petrashevsky

Уровень 46

25 августа, 18:59

...

осспади, снова MVC.в кажлм квесте.  
Безотносительно джавыраш, как умеют у нас запутать "ввод-обработка-вывод", преподнося это как открытие, которое никто не может внятно объяснить.

Ответить

0



LuneFox

инженер по сопровождению в BIFIT

EXPERT

6 февраля, 22:46

...

Насчёт MVC у меня, как у геймера, сложилась такая картинка:

Модель - это игровая приставка, внутри которой происходит вся основная логика игры.  
Контроллер - это джойстик, при помощи которого мы управляем игрой.  
Вид - это телевизор, на который выводится изображение.

Телевизор подключён к приставке, но игра не перестанет работать, если мы его отсоединим (независимость модели от представления). И всё так же будет принимать команды с джойстика, и игровой процесс будет продолжаться вслепую.

Джойстик подсоединён к приставке, но игра не перестанет работать и всё так же будет показывать картинку на телевизоре, если мы вытащим его из гнезда (отсутствие контроллера не повлияет на модель и представление). Изображение может продолжить обновляться по сигналам от приставки, если в игре происходят какие-то независимые от игрока изменения (например, ездят машины или плывут облака), поэтому отсутствие контроллера исключит только пользовательское вмешательство.

Мы можем отключить и телевизор, и джойстик от приставки (и модель продолжит работу), только кому нужна отрезанная от внешнего мира игра, которая играет сама в себя?

А вот если мы выключим приставку, то существование джойстика и телевизора сразу теряет смысл, потому что сами по себе они ничего не могут (сильная зависимость от существования модели).

Самой дорогой частью, очевидно, должна быть именно приставка (модель) - с хорошей производительностью, объёмом оперативной и графической памяти. А джойстик и телевизор должны быть "тонкими" и дешёвыми - джостик должен только передавать нажатия и ничего для игры не вычислять. Телевизор также должен только показывать результирующие пиксели и ничего в игре не менять. (Т. е. не стоит зашивать даже малую часть бизнес логики в вид и контроллер.) Иначе они станут важными компонентами игры и их нельзя будет просто так взять и отключить.

Также независимость игровой приставки позволяет заменять джойстики и телевизоры на более навороченные и удобные без риска сломать игру :)

Ответить

+13

Zheleznyak Maxim

Уровень 40, Moscow, Россия

14 февраля, 17:09

...

а ты не плох)

Ответить

+1

Andrey Karelin

Уровень 41, Sumy, Украина

3 мая, 01:07

...

Тем не менее, именно контроллер по сути запускает игру, имея у себя экземпляр (ссылку) на модель и вьюшку.

Ответить

0

LuneFox

инженер по сопровождению в BIFIT

EXPERT

3 мая, 09:23

...

Кнопку включения приставки можно представить частью интерфейса контроллера, всё норм)

Ответить

0

Maks Panteleev

Java Developer в Bell Integrator

22 июля 2021, 12:00

...

Это после написания задачи MVC прям в тему лекция, спс)

Ответить

+6

Hidden #213

Уровень 13 (Forever&Ever)

4 мая, 21:24

...

Не все осилили эту задачку  
А после этой лекции особо понятнее не стало.

Ответить

0

Maks Panteleev

Java Developer в Bell Integrator

5 мая, 22:28

...

да изи каточка))

Ответить

0

Hidden #213

Уровень 13 (Forever&Ever)

6 мая, 12:20

...

Смотрите кто объявился! Я думал ты пропал  
Приходилось встречать в работе MVC? (не писать самому, а хотя бы просто встречать)  
Или у вас там свои замутки на работе?

Maks Panteleev

Java Developer в Bell Integrator

6 мая, 21:46

...

mvc - это все что ты видишь перед собой, вся джава это mvc, это невозможно не использовать)

1

что там с Wierd West? Затянула игрушка? ;)

да я прошел ее, очень зашла, топчик) хоть и конечно куча багов, но стоит своих денег и времени)  
я играл часов 40

Ответить

0

Hidden #213

Уровень 13 (Forever&Ever)

8 мая, 21:32

...

Уже подразобрался с MVC. Стал понятен сам принцип. Но всё-равно сложновато ещё. Без подсказок бы не решил сам

Надо тоже попробовать поиграть (когда все баги выловят). Выглядит игра шикарно. И много вариантов прохождения. Обожаю такое.  
А сейчас во что-нибудь играешь? ;)

Ответить

0

Maks Panteleev

Java Developer в Bell Integrator

8 мая, 21:37

...

играю, в тактическую рпг король артур)

Ответить

0

Hidden #213

Уровень 13 (Forever&Ever)

9 мая, 12:36

...

Даже не знал, что есть такая игрушка.  
Выглядит, как моя любимая Disciples (которую в итоге упороли, но раньше была шикарной игрой)  
Надо попробовать :) Но в этой игре 40 часами наверное не отделаться...

Ответить

0

Maks Panteleev

Java Developer в Bell Integrator

10 мая, 14:31

...

да хз, смотря на какой сложности)

Ответить

0

Artur Galimov

Уровень 41, Москва, Россия

21 мая 2021, 15:12

...

[Что такое архитектура приложения. Паттерны MVC, MVP, MVVM.](#)

Ответить

+7

Anonymous #2491313

Java Developer в Росатом

23 февраля 2021, 19:55

...

Забыли почему-то о MVP

Ответить

0

Николай Т.

Уровень 40, Рязань, Россия

24 июля 2021, 12:06

...

MVP – Most Valuable Player

Ответить

+2

Андрей

Уровень 41, Самара

29 января 2021, 18:42

...

Топ тема для работяг с Multithreading, где каждая вторая большая задача была на MVC

Ответить

+1

Pig Man

Главная свинья в Свинарнике

12 февраля 2021, 23:31

...

В Multithreading было только 2 больших задачи, связанных с MVC: "MVC" и "2048". О чем ты?

Ответить

+1

Игорь

Full Stack Developer в IgorApplications

29 июля 2021, 22:12

...

Задача на чат очень хорошо демонстрировала, несколько клиентов и у каждого по view и controller

Ответить

0

Елена Попова

Уровень 41

23 декабря 2020, 14:32

...

[https://www.youtube.com/watch?v=1vyf-\\_5OkW8&list=RDCMUCK5d3n3kfkzIArMccS0TTXA&index=6](https://www.youtube.com/watch?v=1vyf-_5OkW8&list=RDCMUCK5d3n3kfkzIArMccS0TTXA&index=6)  
spring mvc

Ответить

+1

Евгений

Ведущий инженер в ПАО Сбербанк

EXPERT

4 июля 2020, 21:57

...

Нв... Тут. наверное. надо знать меру. чтобы не получился толстый. тупой. уродливый контроллер.

НАЧАТЬ ОБУЧЕНИЕ

Джамис Бак (Jamis Buck, автор Capistrano, сейчас работающий в 37signals) в свое время описал концепцию «Тощего контроллера, толстой модели». Крис Хартжес (Chris Hartjes) тоже написал статью на эту тему [...] В рамках этой концепции считается, что по мере возможности логику приложения (вроде бизнес-логики из примера выше) лучше всего помещать в модель, а не контроллер или представление.

Представление должно заниматься **только созданием и отображением интерфейса, через который пользователи смогут сообщить модели о своих намерениях**. Контроллеры — это организаторы, **связывающие введенные в интерфейс данные с действиями модели и передающие вывод обратно**, какое бы представление ни отображало эту модель. Контроллеры должны определять поведение приложения только в плане связи ввода пользователя с вызовами модели, но в остальном должно быть ясно, что **вся логика приложения находится в модели**. Контроллеры — это скромные существа с минимумом кода, которые обеспечивают условия для упорядоченной работы.

[источник](#)

Так что по логике controller должен быть самым маленьким звеном из этой тройки. Как вы видите, тут также утверждается, что информацию получает именно view через свой интерфейс, а потом даёт её на откуп в controller. Насколько я понял, вариаций MVC миллион, но эта на мой взгляд наиболее логичная.

Ответить

+4

**Вэйлор** Уровень 28, Россия

19 июня 2020, 17:53

Не пропущена ни одна задача. Дошел до этой лекции с 859 материи в запасе. Это нормально? Задачи не приносят столько материи, сколько ее списывают за открытие лекции. А некоторые страницы не несут никакой смысловой нагрузки

Ответить

+1

**Алексей** Backend Developer в Neoflex

29 июня 2020, 12:30

Хах, да вообще не парься по этому поводу) Под конец останется примерно 2к неиспользуемой материи

Ответить

0

**Alexey Prilessky** Уровень 40, Минск

13 октября 2020, 19:48

я пропустил пару задач, и у меня на 36-м уровне всего 80 материи и я не знаю как дальше вообще лекции открывать

Ответить

0

**Roman Morozov** Уровень 41, Санкт-Петербург, Россия

23 ноября 2020, 23:19

Это точно не пару)

Ответить

0

**Alexey Prilessky** Уровень 40, Минск

30 ноября 2020, 18:28

Я серьезно , буквально пару пропустил на тот момент. Потом уже пропустил намного больше из за задач на побитовые операции из раздела информатики, которые вообще вряд ли пригодятся во первых, а во вторых это отдельная тема, которую очень долго изучать. В итоге уже задач таких накопилось ппц много, но их бессмысленно решать по моему

Ответить

0

**Илья** Уровень 35, Москва, Россия

7 февраля 2020, 15:57

*/\* Комментарий удален \*/*

Ответить

+3

**Eiffil** Уровень 41, Москва

13 апреля 2020, 00:29

Товарищ в этом видео рассказывает, что контроллер после манипуляций с моделью (например, добавление айфона в корзину) обновляет данные представления. Но в статье выше явно указано, что контроллер этого не делает. Либо само view запрашивает данные у модели. либо у модели есть слушатели, которые получают сообщение об изменении модели. Пока не понимаю, кто прав.

Ответить

0



Показать еще комментарии

ОБУЧЕНИЕ

Курсы программирования

Курс Java

Помощь по задачам

Подписки

СООБЩЕСТВО

Пользователи

Статьи

Форум

Чат

КОМПАНИЯ

О нас

Контакты

Отзывы

FAQ

НАЧАТЬ ОБУЧЕНИЕ



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

 Русский

▼

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

 ДОСТУПНО В  
Google Play

 Загрузите в  
App Store

