

# Паттерны проектирования

**Финал!**



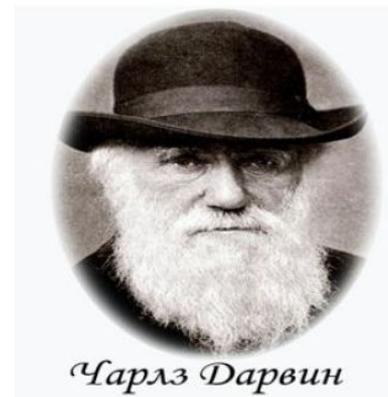
# Экономический дарвинизм

Происходит глобализация на всех уровнях.

Бизнес или его части перемещаются в Интернет

Становится все больше конкурентов на одного клиента.

У клиентов более высокие ожидания и меньше лояльности.

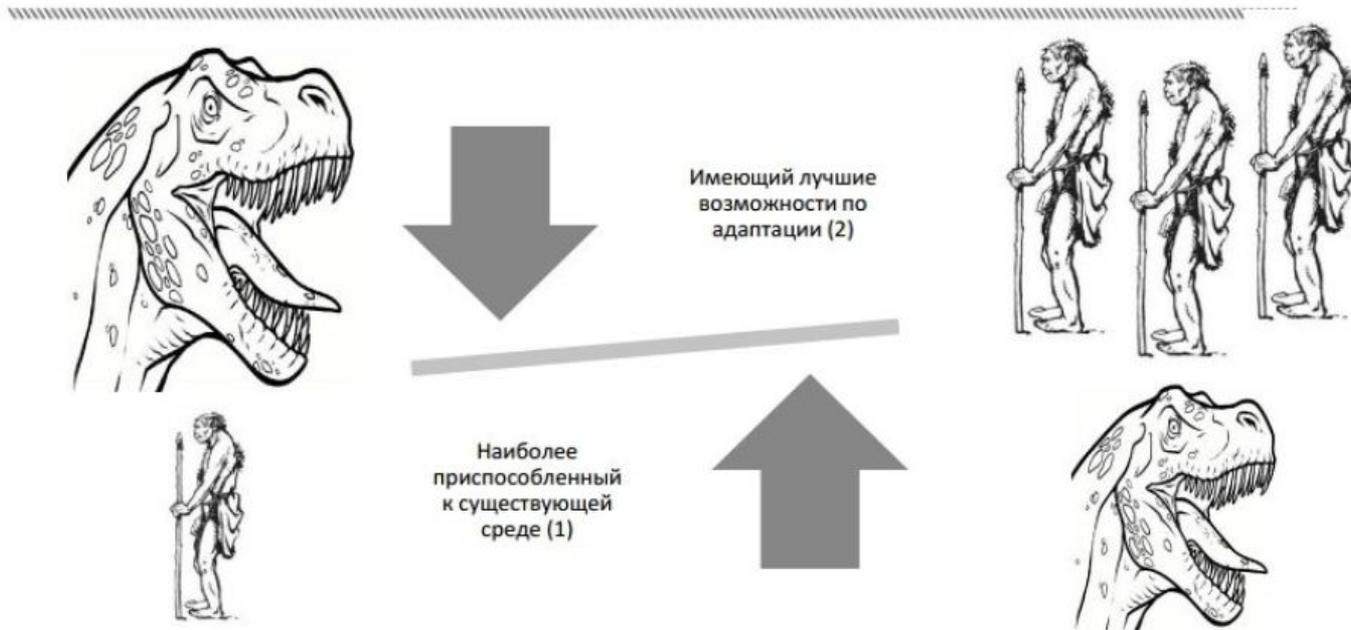


*Чарлз Дарвин*

Основной закон эволюции звучит следующим образом:

- Во-первых, выживает наиболее приспособленный к существующей среде.
- Во-вторых, для выживания необходимо иметь лучшие возможности по адаптации, то есть, приспособленность к будущим изменениям.

## Успех - > Выбор правильного баланса (компромисс)



Успех зависит от выбора правильного баланса, и зачастую — это компромиссное решение. ( Необходимо быть достаточно приспособленным к существующей среде, чтобы выжить. Нужно иметь такие возможности по адаптации, чтобы пережить те изменения, которые могут произойти в этой среде. )

**Чем выше приспособленность к текущим условиям,  
тем меньше готовность к адаптации.**

*Чарльз Дарвин*

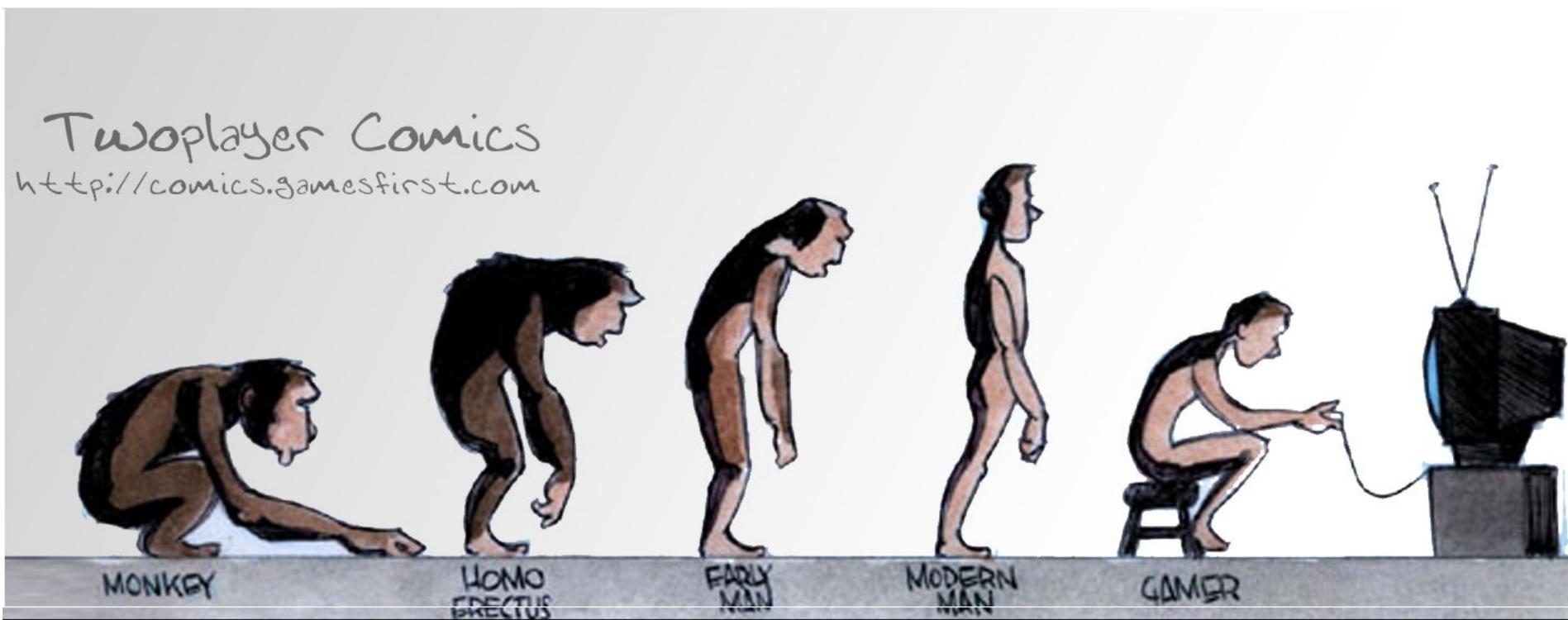
**Второй принцип — Чем выше приспособленность к текущим условиям, тем меньше готовность к адаптации.**

Требования — это условия выживания, обусловленные конкурентной средой. Требования часто являются результатом компромиссного решения.

При этом необходимо подчеркнуть, что если мы с вами хотим управлять своим развитием, то речь идет не о естественном отборе, в результате которого просто выживут те, у кого баланс характеристик приспособляемости будет лучше, тогда решение принимаем не мы. Речь идет об осознанном выборе решений при построении нашей компании или системы, которую мы создаем.

# Каким образом дарвинизм связан с информационными технологиями?

Информационная технология — это такая технология, которая работает исключительно с информацией. Как следствие работы с информацией, эта деятельность может быть передана на обработку вычислительной системе.



# Программа



---

Набор детальных инструкций для вычислительной системы:

- Где взять информацию
- Как преобразовать информацию
- И куда ее отдать

Для того чтобы информацией можно было пользоваться добавляются следующие операции:

- Хранение
- Поиск

# Роль информационных технологий

- Из вспомогательного инструмента к ... центральной нервной системе бизнеса
  - Даже короткие простои ИТ считаются критическим
  - Изменения бизнеса стали невозможны без поставщиков ИТ
- Цифровые или Digital продукты:
  - Полностью цифровые (информационные) продукты ...
  - Во многие «off-line» продукты внедряются ИТ



## **Нужно быть готовым к предстоящим изменениям (ИТ)**

---

- Чем сложнее субъект, тем сложнее осуществлять изменения без потери управления.**
  - Необходимо разделить сложный объект на части таким образом, чтобы изменения в одной части не приводили к нарушению работоспособности в других частях.**
  
- Необходимо поддерживать осведомленность о требованиях других частей, чтобы при изменениях сохранить их работоспособность, или осуществлять изменения синхронно**

# Принцип развития – мир имеет тенденцию **усложняться**

Поэтому одной из важнейших частей готовности к предстоящим изменениям — это, как на подводной лодке-корабле, создать такое разделение на модули, чтобы изменения в одной части не приводили к нарушению работоспособности в других частях.

Поэтому второй вывод, который мы можем сделать, связывая информационные технологии и теорию эволюции — это необходимость поддерживать осведомленность о требованиях других частей для того, чтобы при изменениях сохранить их работоспособность.

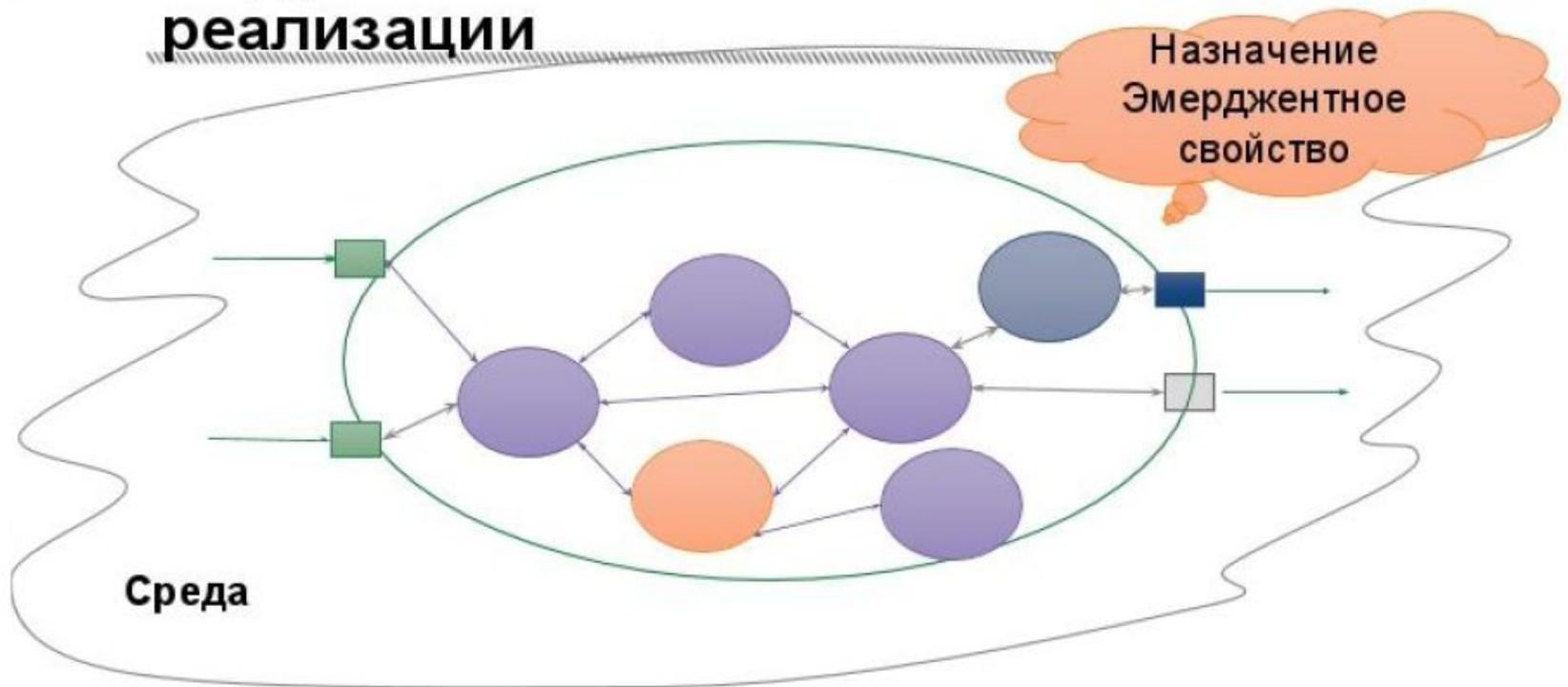
Или производить изменения синхронно в двух частях.



Простейший пример, с которым мы все с вами неоднократно сталкивались — программа вдруг начинает вести себя неадекватно. Вы обновили телефон, вы обновили программу, возможно, даже не та программа, которой вы хотите пользоваться, но она вдруг перестает работать. И вот как раз эта ситуация, когда внезапно то, что работало для вас на протяжении нескольких иногда месяцев, иногда лет, вдруг перестало работать — это как раз и связано с потерей контроля, связано с тем, что кто-то не проработал требований или не учел требования по взаимодействию с другими системами.

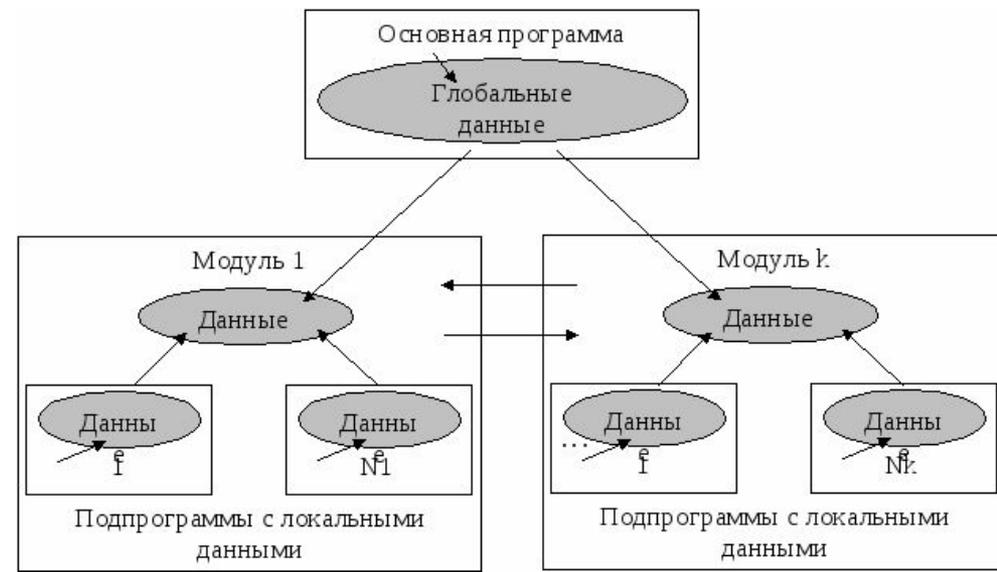
# Для каждого ПО необходима:

## Модульность и независимость от реализации



Независимость от реализации позволяет нам модифицировать объекты входящие внутрь системы таким образом, что это не сказывается на назначении и функционировании системы в целом. То есть, мы можем улучшать некоторые объекты или менять саму систему, меняя ее характеристики в лучшую сторону, при этом не меняя в целом экосистему или не влияя на среду. Если наши интерфейсы взаимодействия сохраняются, соответственно, мы получаем возможность развития системы.

Модульность — это тот самый элемент который позволяет нам, помимо изменений, еще и добавлять новые характеристики. То есть, добавляя отдельные компоненты, которые создают независимое качество, мы соответственно, можем поменять характеристики системы в целом и, меняя между модули, добиваться различных характеристик системы в целом.



# Паттерны

## Порождающие

1. **Abstract Factory (Абстрактная Фабрика)** 5
2. **Builder (Строитель)** 2
3. **Factory Method (Фабричный Метод)** 5
4. **Prototype (Прототип)** 3
5. **Singleton (Одиночка)** 4

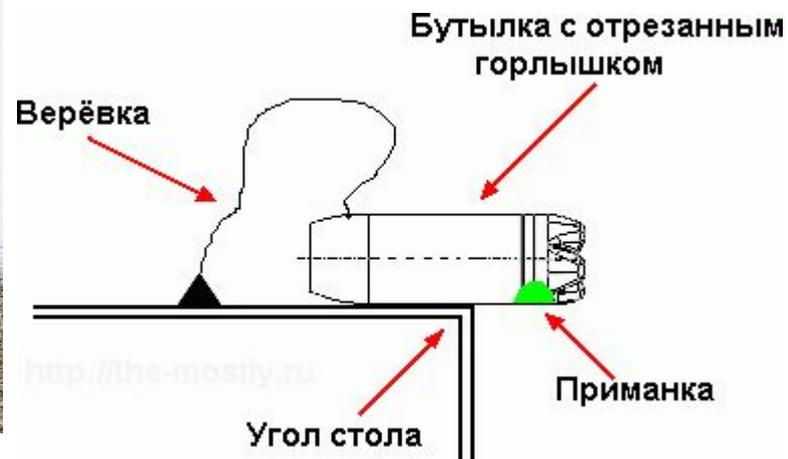
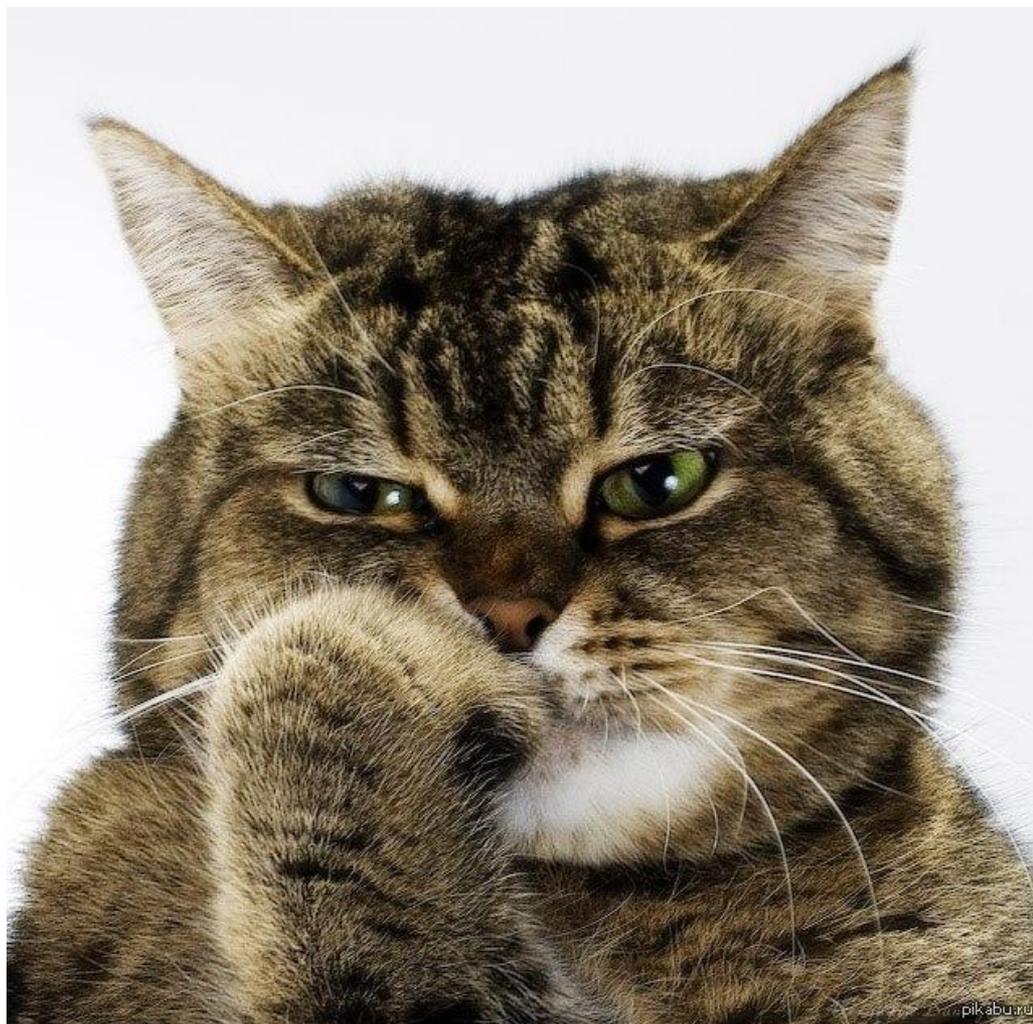
## Структурные

1. **Adapter (Адаптер)** 4
2. **Bridge (Мост)** 3
3. **Composite (Компоновщик)** 4
4. **Decorator (Декоратор)** 3
5. **Facade (Фасад)** 5
6. **Flyweight (Приспособленец)** 1
7. **Proxy (Заместитель)** 4

## Поведенческие

1. **Chain of Responsibility (Цепочка Обязанностей)** 2
2. **Command (Команда)** 4
3. **Interpreter (Интерпретатор)** 1
4. **Iterator (Итератор)** 5
5. **Mediator (Посредник)** 2
6. **Memento (Хранитель)** 1
7. **Observer (Наблюдатель)** 5
8. **State (Состояние)** 3
9. **Strategy (Стратегия)** 4
10. **Template Method (Шаблонный Метод)** 3
11. **Visitor (Посетитель)** 1

# Жизнь сложнее готовых схем



# Составные паттерны. MVC



# Стратегия



Изображение состоит из вложенных окон, панелей, кнопок, надписей и т. д. Каждый компонент является комбинацией (окно) или листом (кнопка). Когда контроллер приказывает представлению обновиться, он обращается к верхнему компоненту, а паттерн Компоновщик делает все остальное.

Модель реализует паттерн Наблюдатель для оповещения заинтересованных объектов об изменениях состояния. Паттерн Наблюдатель обеспечивает полную независимость модели от представлений и контроллеров. Он позволяет использовать разные представления с одной моделью, или даже несколько представлений одновременно.

# Наблюдатель

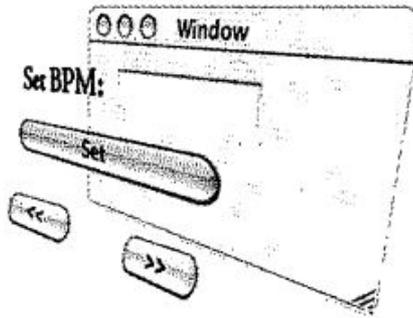


**Компоновщик**



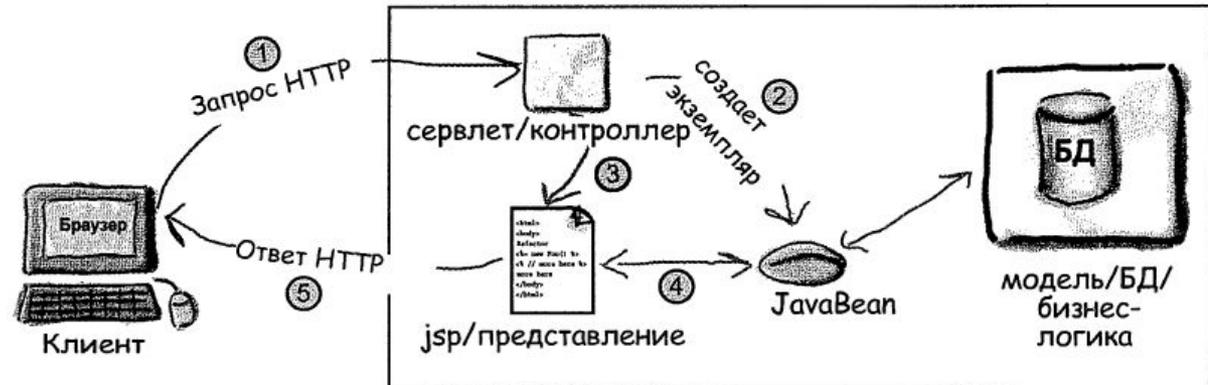
**Представление**

paint()



Представление является комбинацией компонентов GUI (надписи, кнопки, текстовые поля и т. д.). Компонент верхнего уровня содержит другие компоненты, которые содержат другие компоненты и т. д. вплоть до листовых узлов.

# MVC и веб



- 1** **Запрос HTTP принимается сервлетом.**  
Пользователь в браузере выдает запрос HTTP. Обычно при этом передаются данные формы (например, имя и пароль). Сервлет принимает данные формы и разбирает их.
- 2** **Сервер выступает в роли контроллера.**  
Сервер играет роль контроллера и обрабатывает ваш запрос — вероятно, с выдачей запросов к модели (обычно к базе данных). Результат обработки запроса обычно упаковывается в форме JavaBean.
- 3** **Контроллер передает запрос представлению.**  
Представлению соответствует код JSP, единственная задача которого — сгенерировать страницу с представлением модели (4) и всеми элементами, необходимыми для дальнейших действий.
- 5** **Представление возвращает страницу браузеру через HTTP.**  
Страница возвращается браузеру для отображения. Пользователь создает дальнейшие запросы, которые обрабатываются по той же схеме.

Удачи! 😊

