

1

Архитектура ПО без Мистики

Приступим!



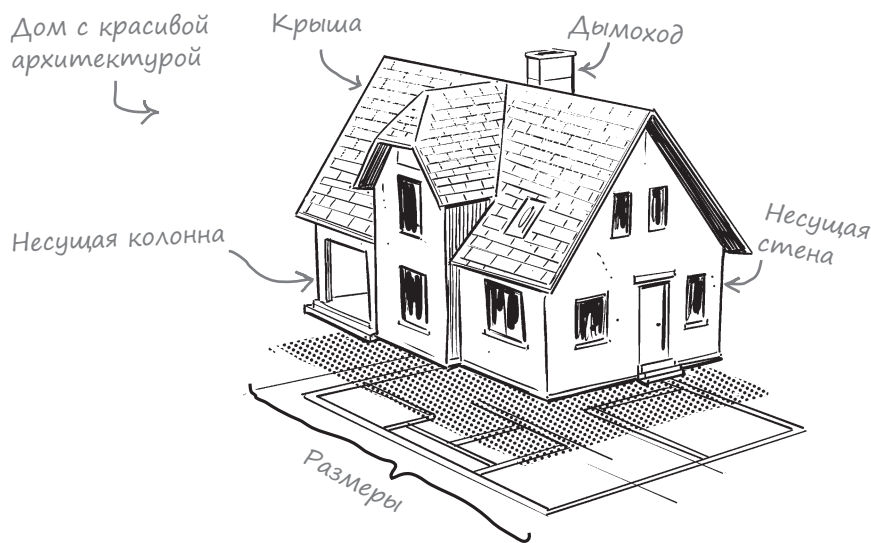
Архитектура программного обеспечения лежит в основе успешной работы системы. Эта глава срывает покров таинственности

с архитектуры ПО и рассказывает об архитектурных измерениях и различиях между архитектурой и дизайном. Об этом важно знать, поскольку понимая архитектурные практики и применяя их, вы сможете создавать эффективные системы, которые не просто лучше функционируют, но и отвечают нуждам и чаяниям бизнеса и продолжают работать, когда техническая и бизнес-среда претерпевают постоянные изменения.

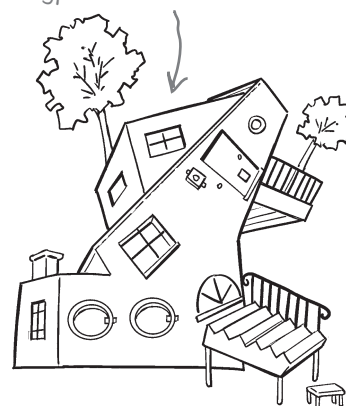
Основы архитектуры программного обеспечения

Чтобы лучше понять, что такое архитектура ПО, вообразите типичный дом. Структура дома — это его *архитектура*: такие его параметры, как форма, количество комнат и этажей, размеры и т. д. Структура обычно представлена в виде плана здания со всеми линиями и рамками, помогающими понять, как его построить. Структурные элементы, подобные показанным ниже, впоследствии будет сложно и дорого изменить, и все они являются *важными* элементами дома.

Метафора здания очень популярна при знакомстве с архитектурой ПО.



Этот дом не только уродлив, но и неудобен



Строительство дома основывается на его архитектуре. Без нее может получиться такой дом, как на рисунке справа.

Архитектура важна и для программных систем. Встречались ли вам системы, которые не масштабируются, ненадежны или сложны в сопровождении? Почти наверняка архитектуре таких систем уделялось недостаточно внимания.



Упражнение

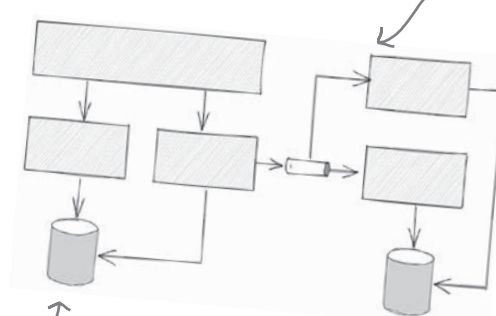
Садоводство — еще одна метафора, полезная для описания архитектуры ПО. Ниже напишите, как планирование сада может быть связано с архитектурой ПО. Наше решение вы сможете увидеть в конце главы.

→ Решение на с. 61

План строительства и архитектура ПО

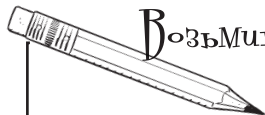
Как же связаны план строительства дома и архитектура ПО? И то и другое представляет строящуюся конструкцию. А как выглядит «план строительства» программной системы? Как множество линий и прямоугольников, конечно.

План здания определяет его структуру: комнаты, стены, лестницы и т. д. Архитектурный план ПО тоже определяет структуру (пользовательские интерфейсы, сервисы, базы данных и протоколы связи). Оба плана отражают рекомендации и ограничения, а также видение конечного результата.



Интересный факт: раньше план здания называли чертежом, но сейчас этот термин устарел (по крайней мере, в строительстве).

На обеих схемах показаны планы строительства.



Возьмите в руку карандаш

Какие особенности дома вы можете назвать *структурными* и связанными с *архитектурой*? Наш ответ вы найдете в конце главы.

Запишите здесь свои мысли.



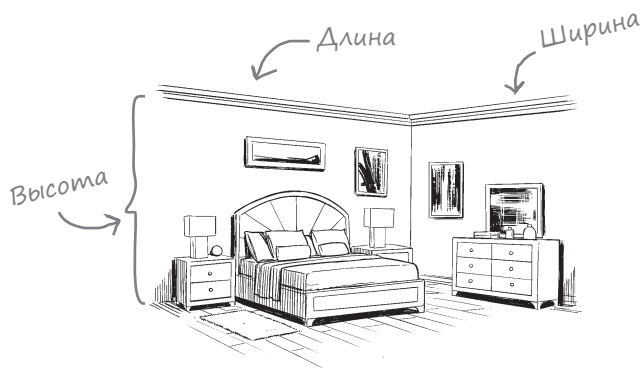
→ Решение на с. 61

Вы заметили, что на плане дома выше не указаны подробные сведения о комнатах, такие как тип напольного покрытия (ковролин или паркет), цвет стен и место, где в спальне будет стоять кровать? Дело в том, что эти подробности не являются *структурными*. Иначе говоря, они определяют не *архитектуру* дома, а его *дизайн*.



Не волнуйтесь! О различиях между ними вы узнаете далее в этой главе. А пока просто сосредоточьтесь на *структуре*, то есть на архитектуре.

Измерения в архитектуре ПО



Многие окружающие нас вещи многомерны. Например, конкретную комнату можно описать, сказав, что ее длина — 5 метров, ширина — 4 метра, а высота потолка — 2,5 метра. Обратите внимание: чтобы правильно описать комнату, необходимо указать все три измерения: высоту, длину и ширину.

Таким же образом можно описать архитектуру ПО по ее измерениям. Разница лишь в том, что в ней *четыре измерения*.

1 Архитектурные характеристики

Это измерение описывает, какие аспекты системы: масштабируемость, тестируемость, доступность и т. д. — поддерживает архитектура.

2 Архитектурные решения

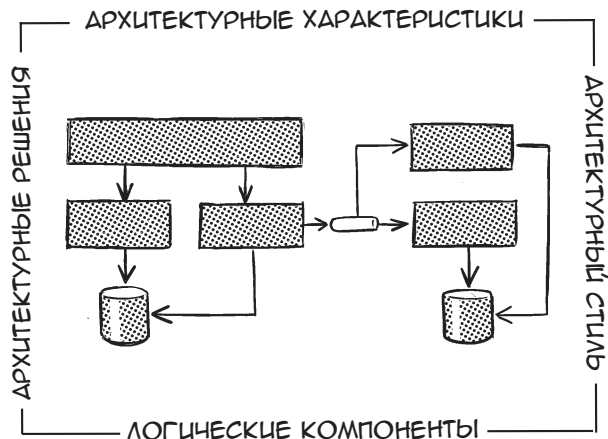
Это измерение содержит важные решения, имеющие долгосрочные или значительные последствия для системы, — например, тип базы данных, количество сервисов и их способ взаимодействия.

3 Логические компоненты

Это измерение описывает функциональные блоки и их взаимодействие. Например, система электронной коммерции может иметь компоненты для управления запасами, обработки платежей и т. д.

4 Архитектурный стиль

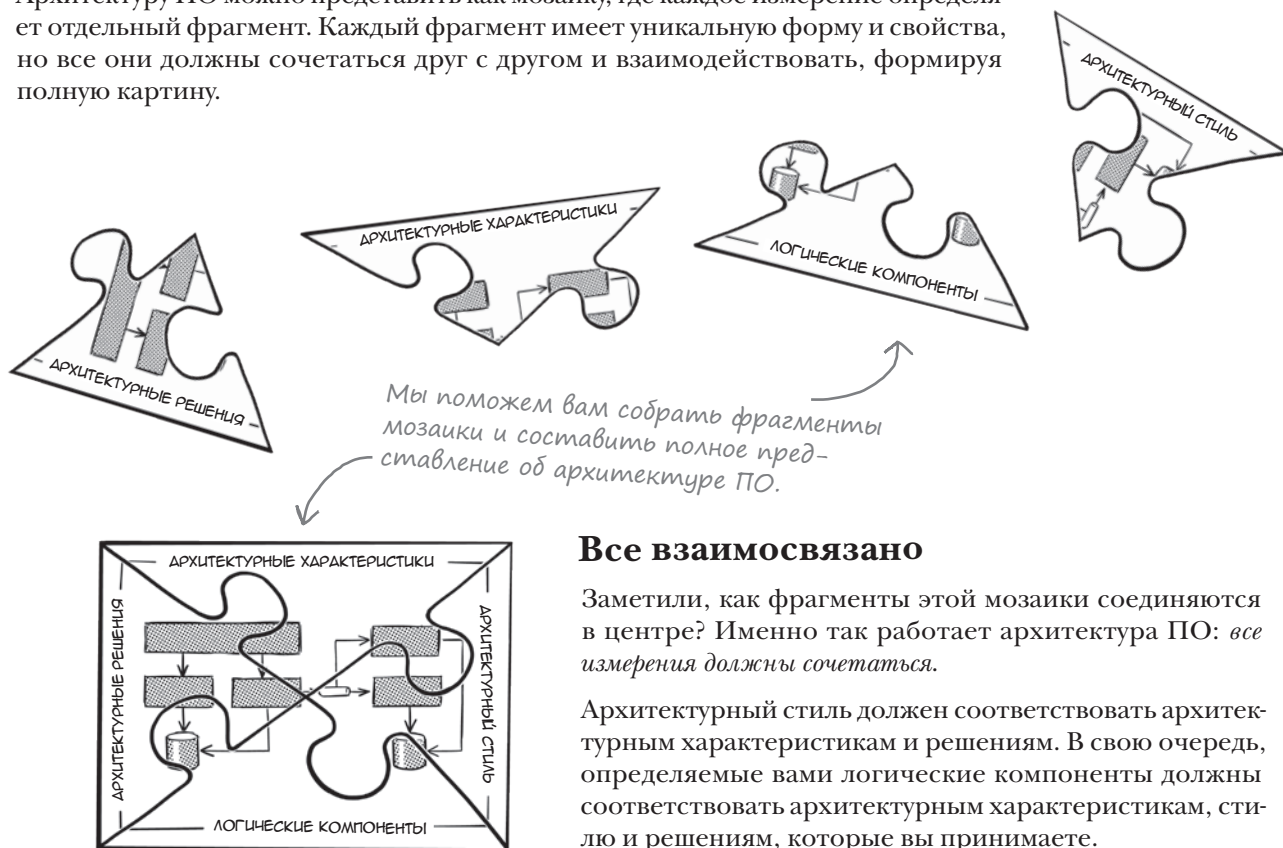
Это измерение определяет общую форму и структуру программной системы точно так же, как план здания определяет общую форму и структуру дома.



Далее в книге вы узнаете о пяти наиболее распространенных архитектурных стилях.

Мозаика из измерений

Архитектуру ПО можно представить как мозаику, где каждое измерение определяет отдельный фрагмент. Каждый фрагмент имеет уникальную форму и свойства, но все они должны сочетаться друг с другом и взаимодействовать, формируя полную картину.



Все взаимосвязано

Заметили, как фрагменты этой мозаики соединяются в центре? Именно так работает архитектура ПО: *все измерения должны сочетаться.*

Архитектурный стиль должен соответствовать архитектурным характеристикам и решениям. В свою очередь, определяемые вами логические компоненты должны соответствовать архитектурным характеристикам, стилю и решениям, которые вы принимаете.

Часто задаваемые вопросы

В: Все ли четыре измерения нужны при создании архитектуры или какие-то можно не учитывать, если времени не хватает?

О: К сожалению, ни одно измерение нельзя не учитывать — все они необходимы для создания и описания архитектуры. Начинающие архитекторы часто ошибаются, используя только одно или два измерения для описания архитектуры. Фраза «Наша архитектура — это микросервисы» описывает одно измерение — архитектурный стиль, но оставляет много вопросов без ответов. Например, какие архитектурные характеристики важны для успешной работы системы? Из каких логических компонентов (функциональных строительных блоков) она состоит? Какие важные решения приняты относительно реализации архитектуры?

Измерение первое: архитектурные характеристики



Архитектурные характеристики – основа архитектуры программной системы. Без них невозможно принять архитектурные решения или проанализировать компромиссы.

Представьте, что выбираете один из двух домов. Один большой, но находится рядом с оживленной автомагистралью. Другой намного меньше, зато расположен в красивом и тихом районе.

Что для вас важнее: размер дома или тишина и вид из окна? Не зная этих характеристик, вы не сможете сделать правильный выбор.

То же относится к архитектуре. Допустим, вам нужно решить, какую базу данных использовать в новой системе: реляционную, простую типа «ключ – значение» или сложную графовую. Выбор будет зависеть от архитектурных характеристик, имеющих для вас решающее значение. Например, можно выбрать графовую БД, если нужен быстрый поиск (назовем это *производительностью*), или традиционную реляционную, если нужно сохранить взаимосвязи между данными (назовем это *целостностью данных*).

производительность

время, необходимое системе для обработки запроса

доступность

время безотказной работы системы; обычно измеряется в «девятках» (так, 99,9% – это три «девятки»).

масштабируемость

способность системы поддерживать постоянное время отклика и количество ошибок по мере увеличения количества пользователей или запросов

Это наиболее распространенные архитектурные характеристики. Вы познакомитесь с ними в главе 2.



Упражнение

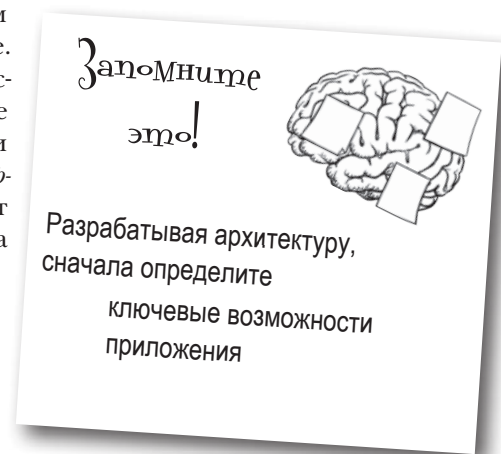
Отметьте описания, которые, по вашему мнению, можно считать архитектурными характеристиками, поддерживающими *структуру* программной системы.

- Изменение размера шрифта в пользовательском интерфейсе.
- Простота внесения изменений.
- Одновременное обслуживание тысяч пользователей.
- Шифрование паролей перед сохранением в базе данных.
- Взаимодействие с множеством внешних систем в целях обработки запроса.

→ Решение на с. 62

Термин «*архитектурные характеристики*» может быть вам незнаком, но это не значит, что вы не слышали его раньше. В совокупности такие аспекты, как производительность, масштабируемость, надежность и доступность, известны также как нефункциональные требования, атрибуты качества и просто «возможности». Нам нравится термин «*архитектурные характеристики*», поскольку все эти аспекты помогают определить характер архитектуры и понять, что она должна поддерживать.

Архитектурные характеристики — это возможности, критически важные или значимые для успешной работы системы.



Что чему соответствует ?

Вам выпал шанс оценить, как много вы уже знаете о типичных архитектурных характеристиках. Сопоставьте характеристики (слева) с определениями (справа). Обратите внимание, что определений больше, чем характеристик, поэтому будьте осторожны — характеристика соответствует не каждому определению.

Расширяемость

Пример от нас.

Гибкость

Интегрируемость

Отказоустойчивость

Осуществимость

При выборе архитектурных решений учитываются сроки, бюджет и навыки разработчиков.

Способность системы сохранять другие части работоспособными при появлении фатальных ошибок.

Простота расширения системы в целях поддержки дополнительных возможностей.

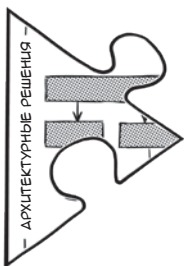
Время, необходимое для отправки ответа пользователю.

Возможность быстро вносить изменения в систему (удобство сопровождения, тестируемость и готовность к развертыванию).

Способность взаимодействовать с другими системами для обработки бизнес-запроса.

Решение на с. 62

Измерение второе: архитектурные решения



Архитектурные решения позволяют выбрать структурные аспекты, имеющие долгосрочные или значительные последствия. Служа своего рода ограничениями, они будут задавать направления работы при планировании и создании системы.

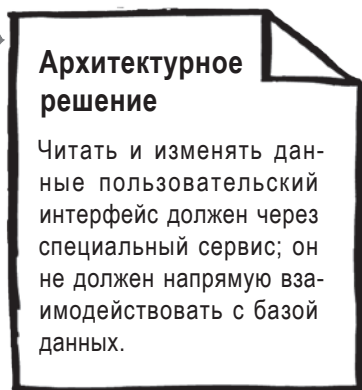
Сколько этажей должно быть в доме — один или два? Какой должна быть крыша — плоской или куполообразной? А может, дом должен быть просторным в стиле ранчо? Это хорошие примеры архитектурных решений, которые затрагивают *структурный* аспект дома.

Как должен выглядеть дом? Это архитектурное решение.



Вы можете решить, что пользовательский интерфейс системы не должен напрямую взаимодействовать с базой данных, а получать и изменять данные он может с помощью базовых сервисов. Это архитектурное решение накладывает особые ограничения на разработку пользовательского интерфейса, а также помогает разработчикам узнать, как другие компоненты должны получать данные из базы и обновлять их.

Это пример архитектурного решения.



Это решение накладывает ограничение и указывает направления работы.



Так изображаются сервисы. Они часто будут встречаться в книге.



Это база данных (БД).

В главе 3 вы узнаете много нового об архитектурных решениях.