2 Моделирование

1.1 Понятие моделирования и модели

Моделирование = метод научного познания: изучение некоторого объекта посредством его модели

Модель = объект-заменитель объекта оригинала (в определенном соответствии с оригиналом; обеспечивает изучение некоторых его свойств)

Моделирование ИС

1. зачем изучать ИС ?

(разработчик должен точно знать, ЧТО требуется сделать)

2. в чем сложность изучения?

ИС еще нет => есть 2 виртуальные системы: заказчика и разработчика => нужны 2 модели:

1. “что хочет заказчик”
2. “что может разработчик”

Свойства модели

- Позволяет акцентировать наиболее важные аспекты объекта
- Всегда хуже оригинала = его упрощение

Модель

= абстрактное описание на некотором формальном языке некоторых аспектов системы, важных с точки зрения моделирования

Цели моделирования

- понять ПрО
- проанализировать поведение системы во времени
- записать принятое проектное решение

=> Различные модели для различных целей

Принципы моделирования

1. Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть решение (парадигмы: логическая, функциональная, объектная...)

Принципы моделирования

2. Каждая модель может быть воплощена с разной степенью абстракции

- поверхностное описание = общение с заказчиком;
- битовое представление и обработка = спецификация

Принципы моделирования

3. Лучшие модели = те, которые ближе к реальности

Но детальность модели должна соответствовать цели моделирования !

Принципы моделирования

4. Одной модели недостаточно

= несколько моделей с акцентами на различные стороны системы

Классификации моделей

- материальные (модель авто в аэродинамической трубе)
- идеальные (описательные на некотором языке)

= изучаем только идеальные

Классификации моделей

По точке зрения на систему:

- Статические, СМ (структурные свойства)
- Динамические, ДМ (поведенческие свойства)
- Функциональные, ФМ (функциональные свойства)

Ортогональные взгляды на систему

Статическая модель

- **Описывает составные части системы, их структуру, связи между ними, операции**
- **Операции:**

= события ДМ

= функции ФМ

Динамическая модель

- **Описывает последовательность выполнения шагов в процессе функционирования системы**
- **Объясняет:**

= вызовы операций СМ;

= вычисления функций ФМ

Функциональная модель

- **Описывает преобразования, осуществляемые системой**
- **Раскрывает содержание:**

= операций СМ;

= событий ОМ.

Важность моделей для ПрО

- **Не интерактивные вычислительные задачи => ФМ**
- **Интерактивные ИС => ДМ**
- **Системы с нетривиальными структурами данным => СМ**

Классификация Г.Буча (для программных систем и ИС)

Модель прецедентов

Прецедент = описывает наблюдаемое поведение системы (пользователи, аналитики, тестировщики)

Модель проектирования

Классы, интерфейсы и кооперации = словарь задачи и ее решения

Поддерживает функциональные требования к системе

Модель процессов

Потоки и процессы, формирующие механизмы параллелизма и синхронизации в системе

Модель реализации

Компоненты и файлы для сборки и выпуска конечного продукта

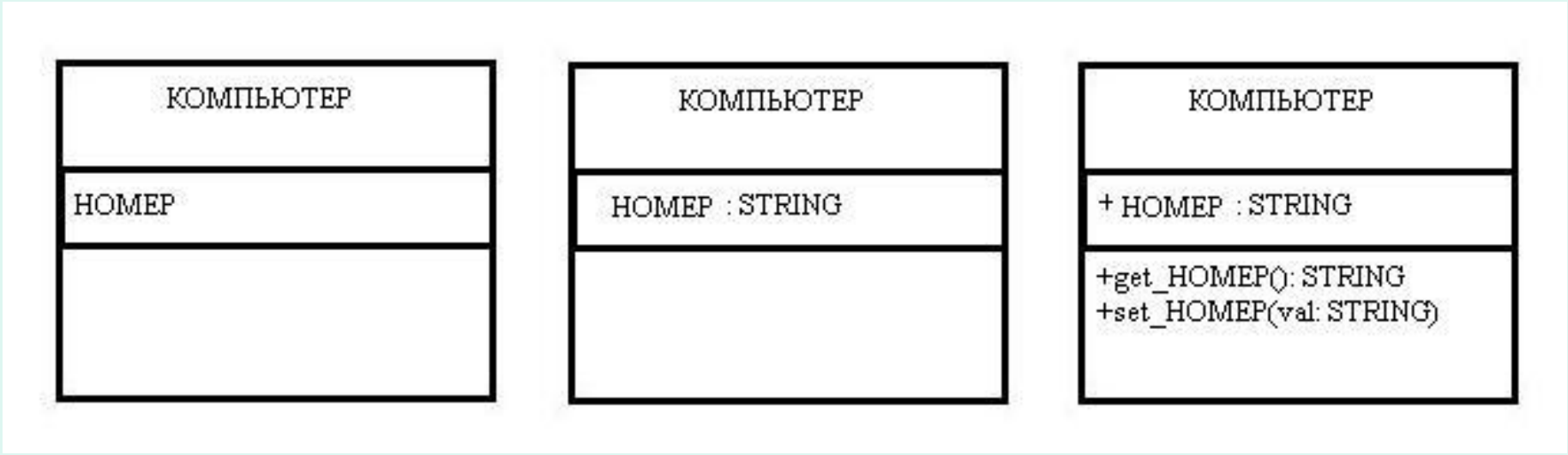
Модель развертывания

Узлы, формирующие топологию аппаратных средств системы, на которых она выполняется

Классификация по степени абстракции (второй принцип моделирования)

1. Концептуальные модели (высокоуровневые, в терминах ПрО)
2. Модели спецификации (внешний вид и внешнее поведение системы)
3. Модели реализации (внутреннее устройство системы, конкретный способ реализации внешнего облика и наблюдаемого поведения)

Пример трех моделей Компьютера



О важности моделей

Концептуальная и модель спецификации – необходима всегда

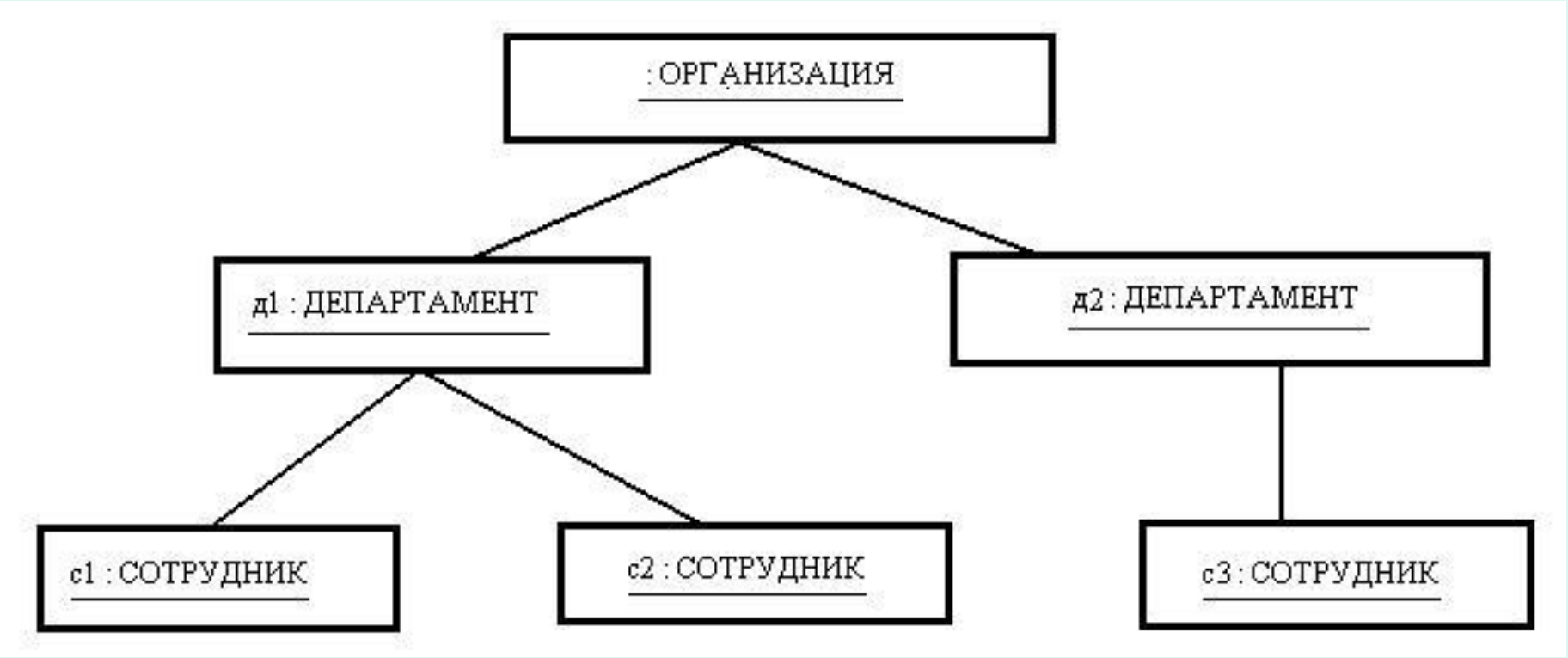
Модель реализация трудоемкая – имеет смысл для иллюстрации нестандартного решения

Метамоделирование

Метамодель –
модель модели (**<= неверно !**)

Метамодель – модель языка моделирования

Модель организации (вариант 1)

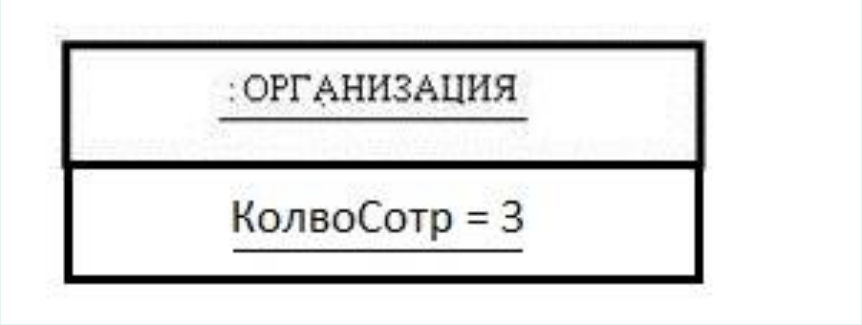


Модель организации

- Обозначения:

- прямоугольники = экземпляры классов (**объекты**);
- имена этих классов записаны после символа **:** **(например, Департамент);**
- имена объектов этих классов записаны перед символом **:** **(например, д1);**
- **линии** = связи между объектами

Модель организации (варианты 2 и 3)



Простые модели

Модели организации (варианты 1 - 3) = простые модели, они моделируют:

- ОРГАНИЗАЦИЮ;
- друг друга (с различной степенью абстракции).

Метамодель (вариант 1)

Выделяем на основе анализа простых моделей 3 элемента: **объект, связь и слот**

Рис. 2.1 - Метамодель

Метамодель (вариант 1)

- Каждый объект принадлежит определенному классу и характеризуется набором свойств (слотов)
- Для идентификации должны иметь **Имя**:

- каждый объект;
- каждый класс;
- каждый слот.

- Слот также имеет **Значение**

Метамодель (вариант 1)

- Сплошная линия обозначает ассоциацию между элементами
- На линии ассоциации размещают символы кратности (множественности):

- числа = конкретные значения (**1** или **2**);
- * - произвольное количество

Метамодель (вариант 1)

- Ассоциация между ***Объектом*** и ***Слотом***:

“У каждого Объекта может быть произвольное количество слотов”

- Ассоциация между ***Объектом*** и ***Связью***:

“Между двумя объектами может быть произвольное количество связей”

Имеем язык, который можно смоделировать => на рис. 2.1 - **метамодель**

Метамодель (вариант 2)

Выделяем на основе анализа простых моделей 3 понятия: **организация, департамент и сотрудник**

Рис. 2.2 - Метамодель

Метамодель (вариант 2)

- Использована нотация диаграмм классов (язык UML)
- Каждый класс представлен прямоугольником с тремя секциями:

- имя (обязательная);

- атрибуты;

- операции.

Метамодель (вариант 2)

- Использована нотация диаграмм классов (язык UML)
- Линии с ромбами на конце обозначают частный случай ассоциации между классами (отношение **“часть - целое”**):

  - ромб размещается у класса **“часть”:**

“Каждая Организация состоит из произвольного количества Департаментов”;

“Произвольное количество Сотрудников могут входить в состав Организации (Департамента)”.

Метамодель (вариант 2)

- Закрашенный ромб обозначает более сильную связь:

  “При уничтожении Организации ее Департаменты прекращают существование”;

  “При уничтожении Организации (Департамента) их Сотрудники продолжают существовать”.

Имеем язык, который можно смоделировать => на рис. 2.2 - **метамодель**

Модели и метамодели

- Имеем 3 модели и 2 метамодели
- Модели похожи и отличаются степенью детализации
- **Метамодели различные:**

- Вариант 1 описывает предметно-независимый язык;
- Вариант 2 описывает предметно-зависимый язык.

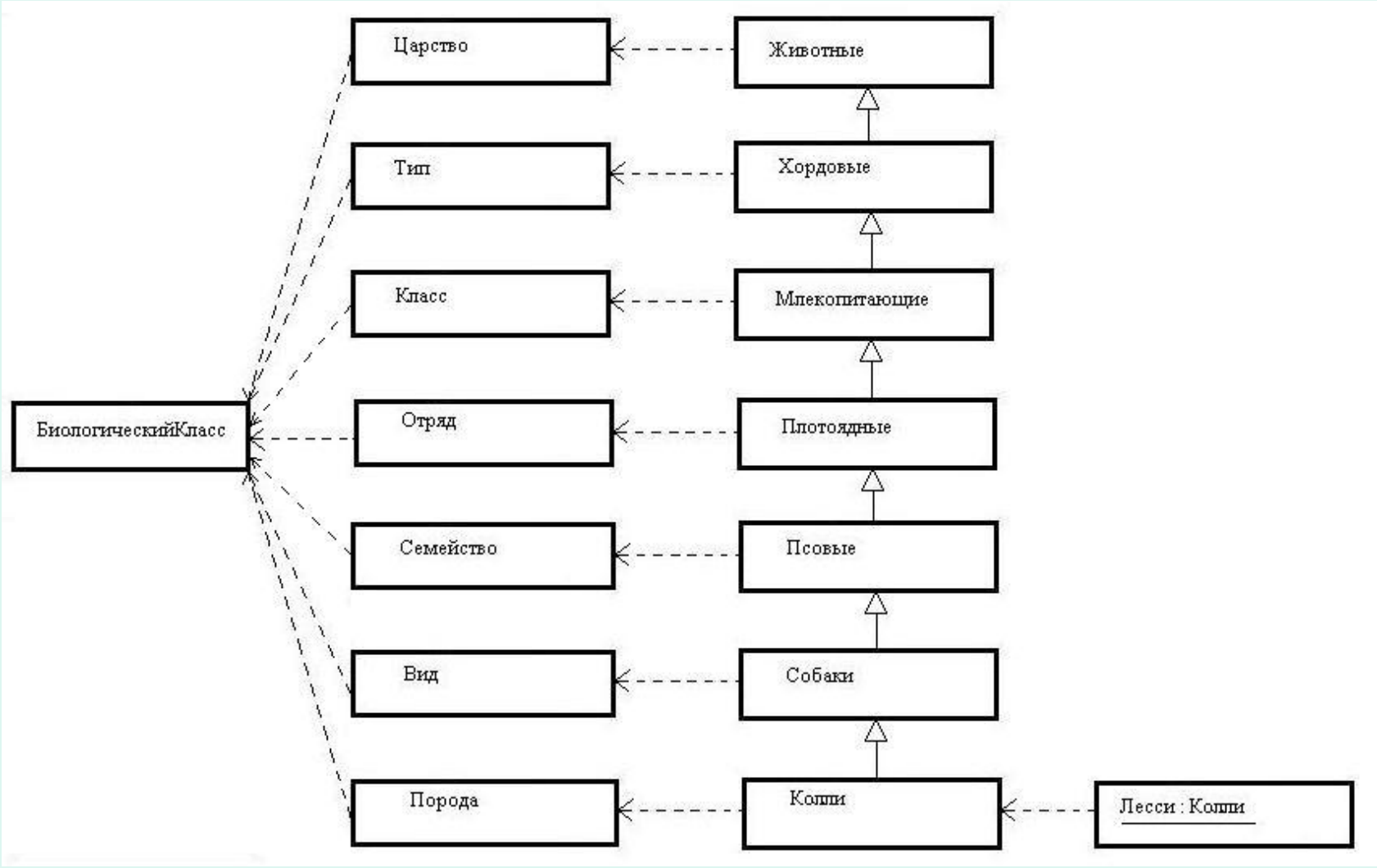
Модели и метамодели

- **Лингвистическая метамодель** = метамодель, которая описывает предметно-независимый язык моделирования.
- **Онтологическая метамодель** = метамодель, которая описывает предметно-зависимый язык моделирования.

Метамодели рис. 2.1 и 2.2

- модели описаны на одном языке (**классы, ассоциации, атрибуты**) => это **онтологическая метамодель**
- графическая нотация моделей (**прямоугольники, дуги, ромбы, надписи**) => это **лингвистическая метамодель**

Пример многоуровневого онтологического моделирования = классификация в биологии



- Пунктирная стрелка обозначает отношение **“Класс - Экземпляр”** и указывает на класс
- Треугольник на сплошной стрелке обозначает **обобщение** и указывает на более общее понятие
- Нижний (первый) уровень – **объект** (конкретная собака **Лесси**)
- Второй уровень – **часть** классификации
- Третий уровень – **основания** классификации
- Четвертый уровень – **самый общий класс**, все экземпляры которого имеют отношение к биологии

- Каждый уровень определяет мини-язык
- Каждый нижестоящий уровень описывается в терминах вышестоящего уровня:

= **Колли** – конкретный экземпляр **Породы**

= **Псовые** – одно из **Семейств**

- **«Лесси» – это экземпляр класса «Колли»**
- **«Колли» – это класс**
- **«Порода» – это метакласс**
- **«БиологическийКласс» – это мета-метакласс**

- **Метакласс** = класс, экземплярами которого являются другие классы.
- **Класс** = совокупность объектов, обладающих схожей структурой, поведением и семантикой.
- **Семантика** = смысл, суть, значение некоторого предмета или явления

Классификация ИС по уровню и составу моделей

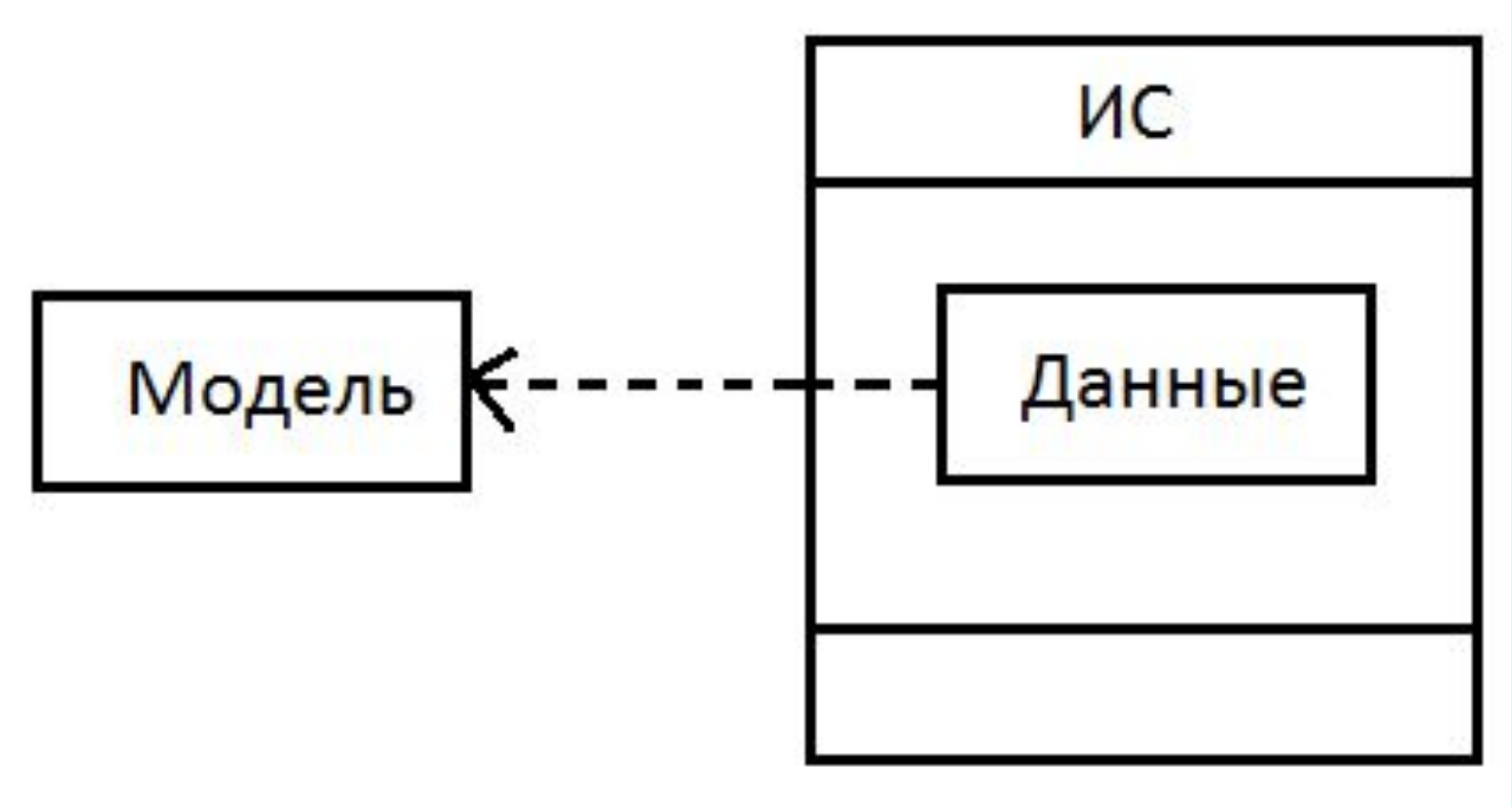
Классическая четырехуровневая иерархия моделей

- **Уровень M0** = данные, описывающие состояние предметной области
- **Уровень M1** = онтологическая метамодель для уровня M0 и содержит модель предметной области
- **Уровень M2** = лингвистическая метамодель для уровней M1 и M0

(на этом уровне модель языка моделирования, с которыми работают аналитики, CASE-средства...)

- **Уровень M3** = язык, на котором описываются метамодели уровня M2 и который обычно описывается на самом себе

1) Традиционные информационные системы



1) Традиционные информационные системы

- **Данные:**

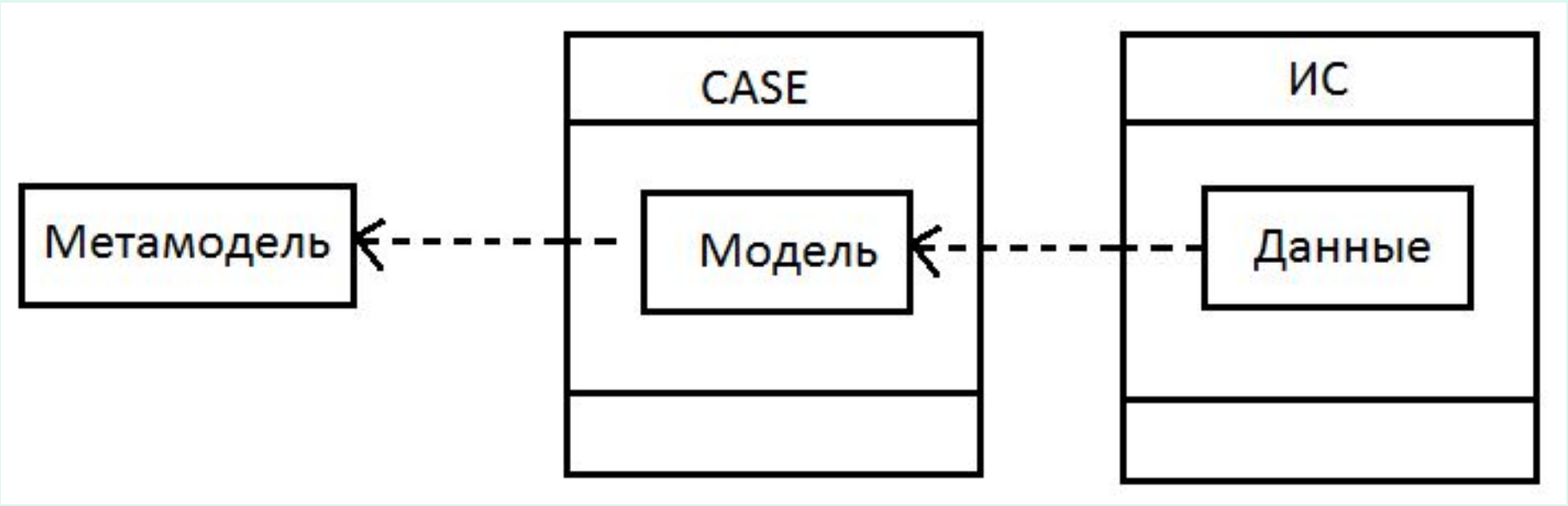
- находятся внутри системы;
- они описывают состояние ПрО;
- они соответствуют некоторой модели ПрО, которая может быть описана на любом языке.

- **Модель ПрО:**

- разрабатывается аналитиками;
- разработчики реализуют ее с помощью средств программирования.

- **При изменении модели** надо переписывать и перекомпилировать исходные коды системы

2) Традиционные CASE-технологии



2) Традиционные CASE-технологии

- **Модель ПрО:**

- определяется формально;
- находится внутри CASE–средства;
- описывается в терминах метамодели, которая может быть определена на любом языке.

- **Метамодель ПрО:**

- разрабатывается аналитиками;
- разработчики реализуют ее с помощью средств программирования.

- **При изменении метамодели** надо переписывать и перекомпилировать код CASE–средства

2) Традиционные CASE-технологии

- **CASE–средство**:

- предоставляет инструменты для создания редактирования моделей;
- позволяет частично сгенерировать код ИС;

- **Полученная ИС:**

- реализует необходимые структуры данных;
- обеспечивает доступ к БД;
- предоставляет стандартный интерфейс пользователя;
- ДМ и ФМ дописываются вручную.

2) Традиционные CASE-технологии

- **После повторной генерации:**

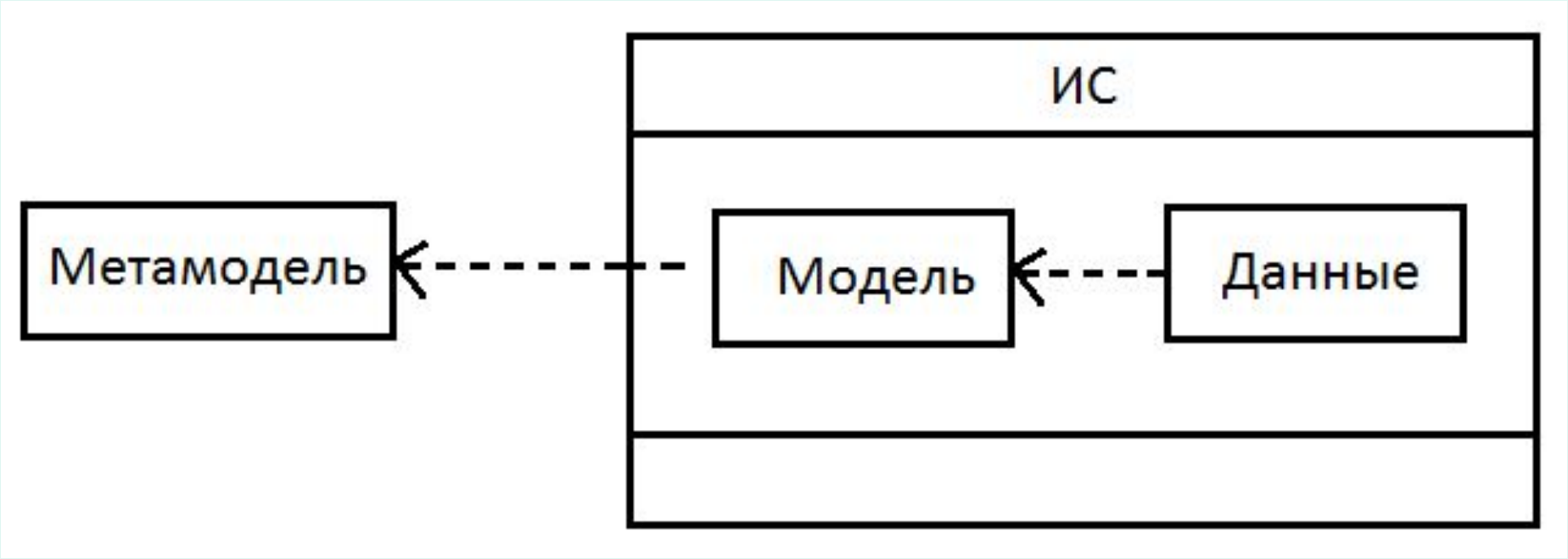
требуется доработка кода вручную

- **Достоинства:**

- экономится время на начальных этапах разработки;

- поддерживается соответствие между системой и моделью.

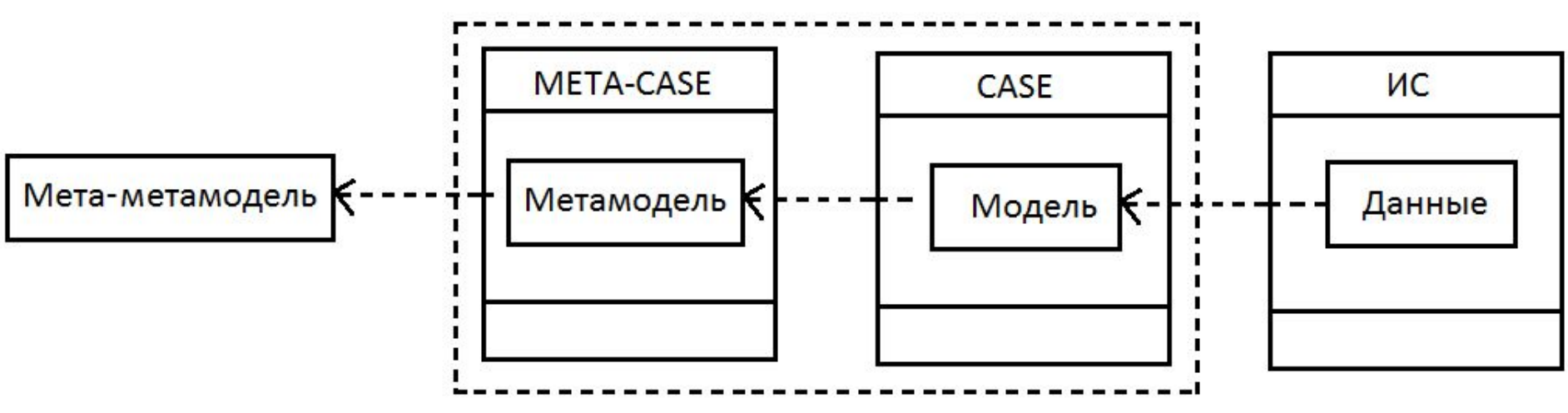
3) Информационные системы, управляемые метаданными



3) Информационные системы, управляемые метаданными

- **Модель ПрО:**
- находится внутри ИС;
- выступает в роли «управляющей программы».
- **ИС:** выступает в роли интерпретатора;
- **Недостатки:** нельзя дописать код вручную (снижается универсальность)
- **Достоинства:** При изменении модели не требуется повторять кодирование и компиляцию => ИС работает с новой моделью
- **Пример:** ***1С: Предприятие***

4) Технология DSM с генерацией кода



- **DSM** (*Domain Specific Modeling*, моделирование в терминах предметной области)

4) Технология DSM с генерацией кода

- **Для решения каждой задачи**:

- применяется свой язык моделирования для решения каждой задачи (с понятиями и отношениями ПрО)

4) Технология DSM с генерацией кода

- **Мета-метамодель** – реализуется META-CASE-средством
- **Метамодель**:

– описывается META-CASE-средством;

– она определяет DSL (Domain Specific Language);

– на ее основе генерируется CASE-средство

- **Модель ПрО**:

– описывается CASE-средством;

– на ее основе генерируется **ИС**.

- META-CASE- и CASE-средства могут быть объединены

4) Технология DSM с генерацией кода

- Использование **DSL** позволяет упростить процесс создания моделей ПрО, в котором могут принимать участие эксперты заказчика
- **Достоинства и недостатки** = традиционной CASE-технологии

5) Технология DSM с интерпретацией метаданных

5) Технология DSM с интерпретацией метаданных

- Комбинация подходов **3)** и **4)**
- Внутри ИС находятся **метамодель, модель и данные**
- **Мета-метамодель** должна быть максимально выразительной
- **Недостаток:** потеря производительности из-за интерпретации сразу двух уровней метамоделей
- **Достоинство:** гибкость системы (при достаточной выразительности мета-метамодели).

Домашнее задание

- Выполнить многоуровневое онтологическое моделирование для объекта:
- **«компьютерный вирус Worm»**
- **«антивирусная программа Comodo Antivirus»**