

Анализ и концептуальное моделирование систем

профессор, д.т.н. Смоленцева Т.Е.:

smolenceva@mirea.ru:

ЛЕКЦИЯ 5.

Модели анализа, проектирования, реализации.

СОДЕРЖАНИЕ ЛЕКЦИИ:

- Способы использования UML.
- Построение модели системы.
- Модели анализа, проектирования.

1. Способы использования UML.

1

Рисование картинок. Графические средства UML можно и нужно использовать безотносительно ко всему остальному. Даже рисование диаграмм карандашом на бумаге позволяет упорядочить мысли и зафиксировать для себя существенную информацию о моделируемом приложении или иной системе.

2

Обмен информацией. Сообщество людей, применяющих и понимающих UML стремительно растет. Если вы будете использовать UML, то вас будут понимать другие и вы будете понимать других с полувзгляда.

3

Спецификация систем. Это важнейший способ использования UML.

4

Повторное использование архитектурных решений. Повторное использование ранее разработанных решений — ключ к повышению эффективности.

5

Генерация кода. Генерировать код нужно и можно, но возможности имеющихся инструментов не стоит переоценивать.

6

Имитационное моделирование. Возможности построения моделей UML, из которых путем вычислительных экспериментов можно было бы извлекать информацию о моделируемом объекте, пока что уступают возможностям специализированных систем, сконструированных для этой цели.

7

Верификация моделей. Было бы замечательно, если бы по модели можно было бы делать формальные заключения о свойствах: модель непротиворечива, согласована, эффективна.

Инструментальная поддержка

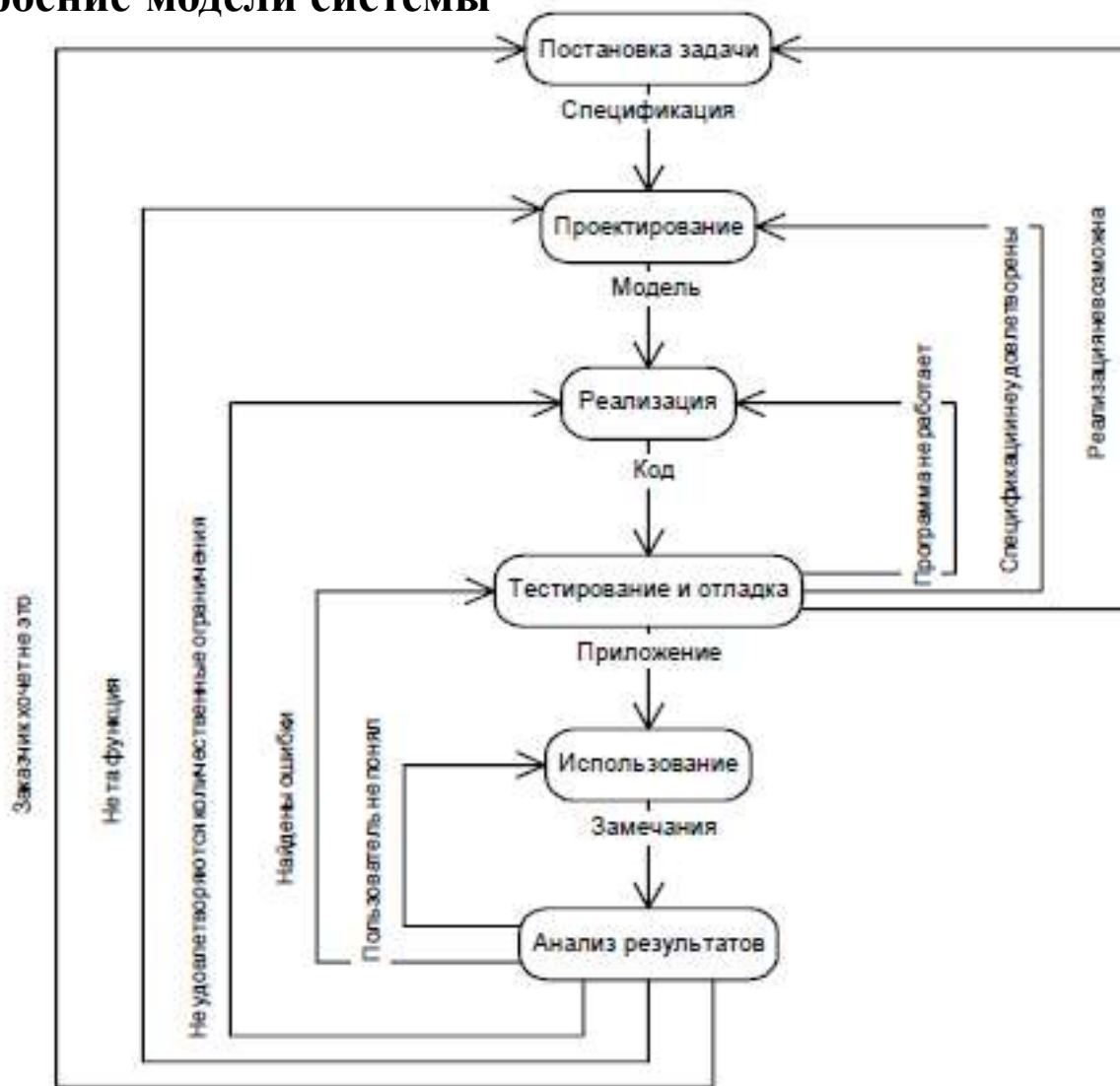


В описании UML используются три языковых уровня.

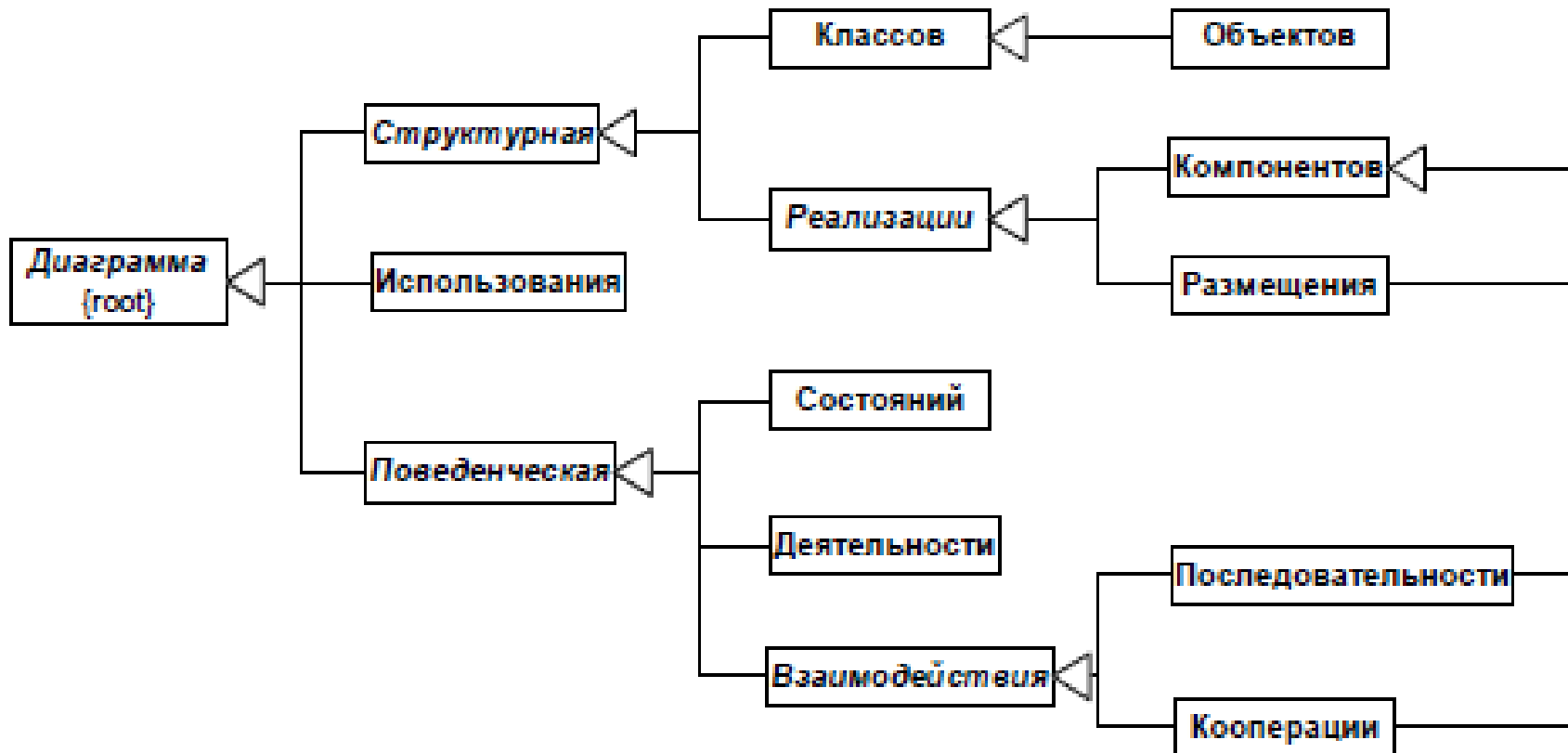
- *Мета-метамодель, то есть описание языка, на котором описана метамодель.*
- *Метамодель, то есть описание языка, на котором описываются модели.*
- *Модель, то есть описание самой моделируемой предметной области.*

2. Построение модели системы

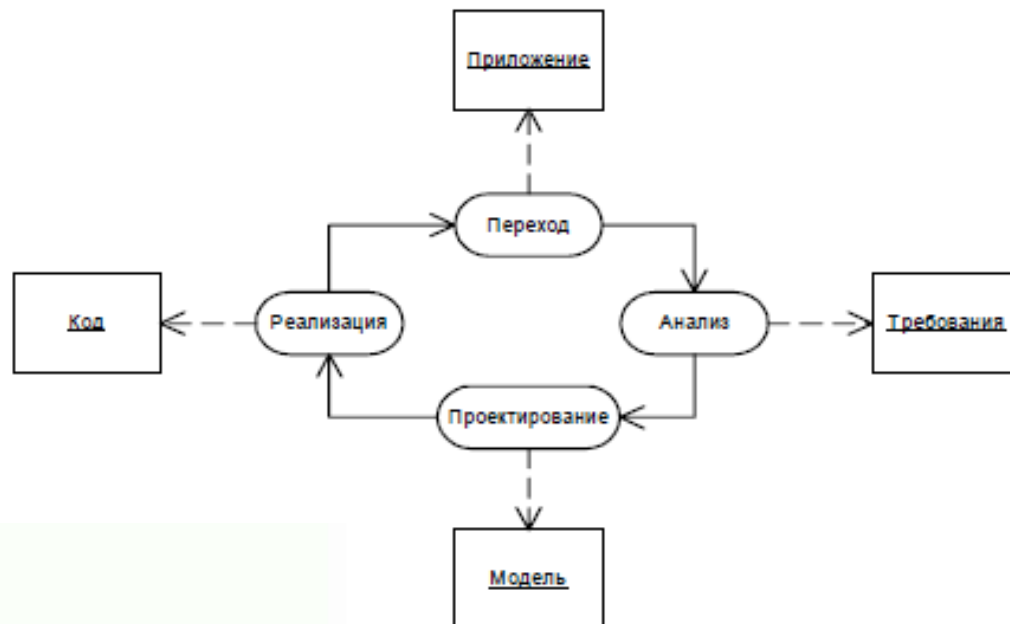
Жизненный цикл приложения



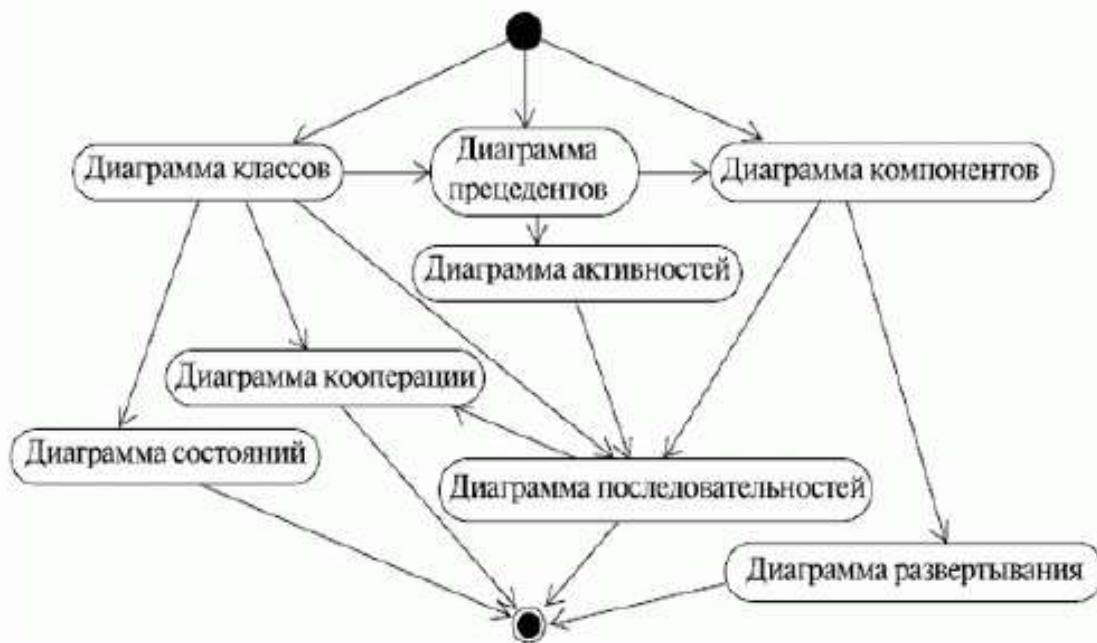
Иерархия типов диаграмм



Итеративный процесс разработки



Построение модели системы



Документирование этапов разработки с помощью UML



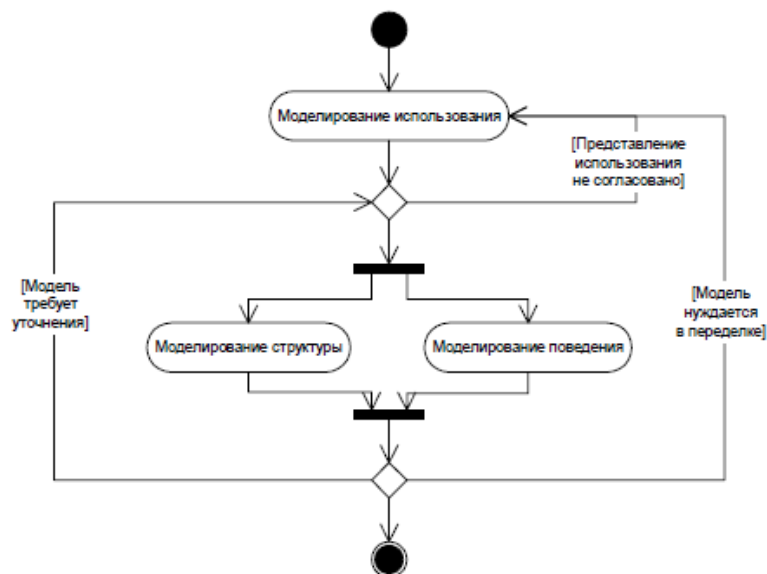
Корпоративные приложения часто требуют для своей работы некоторой ИТ-инфраструктуры, хранят информацию в базах данных, расположенных где-то на серверах компании, вызывают веб-сервисы, используют общие ресурсы и т. д.



Представления модели и диаграммы в языке UML

Представления	Диаграммы	Основные концепции
Статическое представление	Диаграмма классов	Класс, ассоциация, обобщение, зависимость, реализация, интерфейс
Представление использования	Диаграмма использования	Вариант использования, действующее лицо, ассоциация, расширение, включение, обобщение вариантов использования
Представление реализации	Диаграмма компонентов	Компонент, интерфейс, зависимость, реализация
Представление размещения	Диаграмма размещения	Узел, компонент, зависимость, расположение
Представление конечных автоматов	Диаграмма состояний	Состояние, событие, переход, действие
Представление деятельности	Диаграмма деятельности	Состояние, деятельность, переход по завершении, развилка, слияние
Представление взаимодействия	Диаграмма последовательности	Взаимодействие, объект, сообщение, активация
	Диаграмма кооперации	Кооперация, взаимодействие, роль в кооперации, сообщение
Представление управления моделью	Диаграмма классов	Пакет, подсистема, модель

Процесс моделирования



3. Модель анализа, проектирования

При разработке модели анализа рекомендуется построить следующие диаграммы (основные артефакты):

- *классов анализа;*
- *последовательности;*
- *кооперации.*

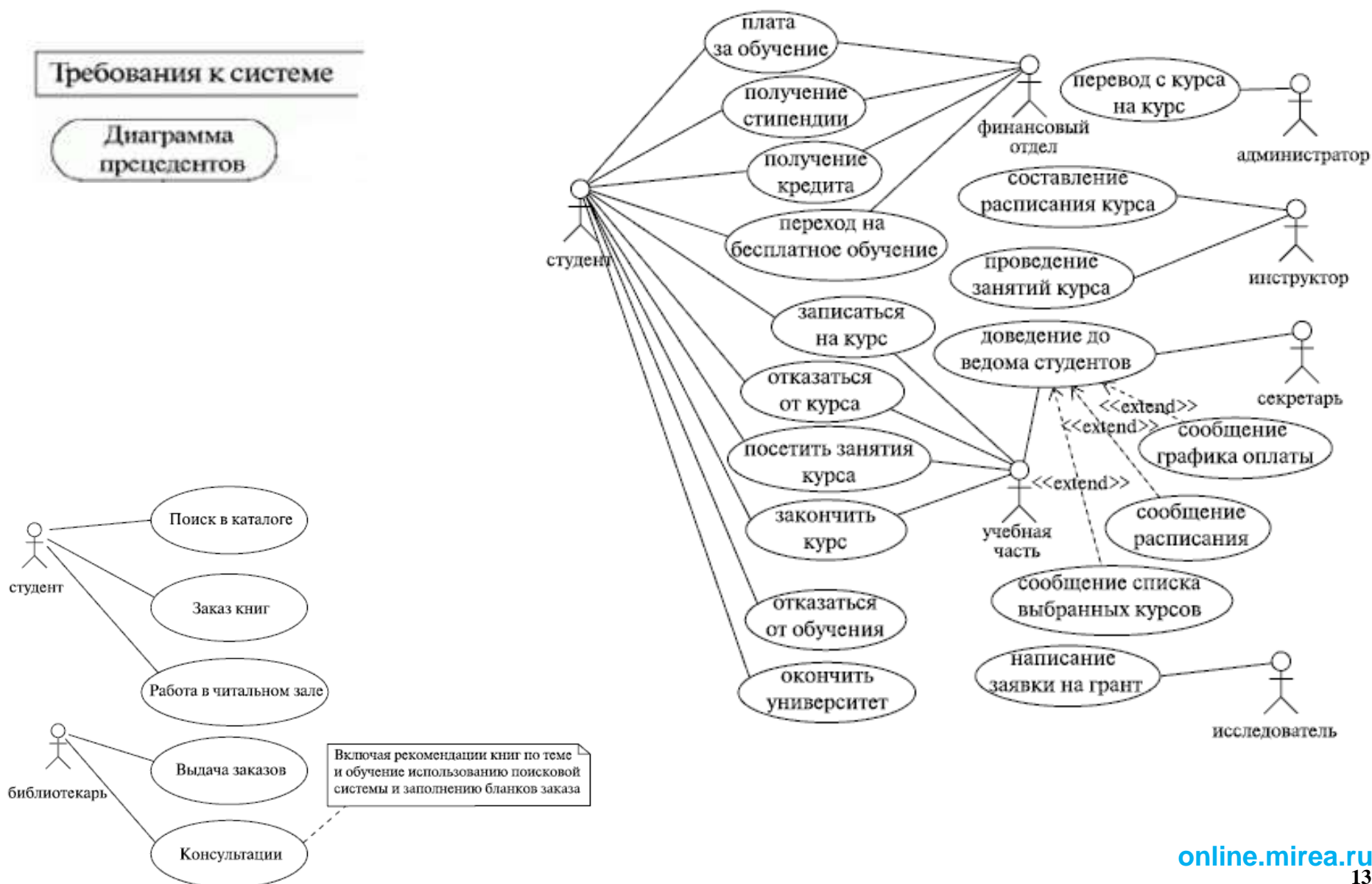
В качестве дополнительных артефактов в данную модель также входят согласованные между заказчиком и разработчиком нефункциональные требования к системе.

Группы требований (параметры качества):

- *требования практичности;*
- *требования надежности;*
- *требования производительности.*

Требования к системе

Диаграмма прецедентов



Информационная система «Отдел кадров» (ОК) предназначена для ввода, хранения и обработки информации о сотрудниках и движении кадров. Система должна обеспечивать выполнение следующих основных функций.

1. Прием, перевод и увольнение сотрудников.
2. Создание и ликвидация подразделений.
2. Создание вакансий и сокращение должностей.

прием сотрудника

перевод сотрудника

увольнение сотрудника

создание подразделения

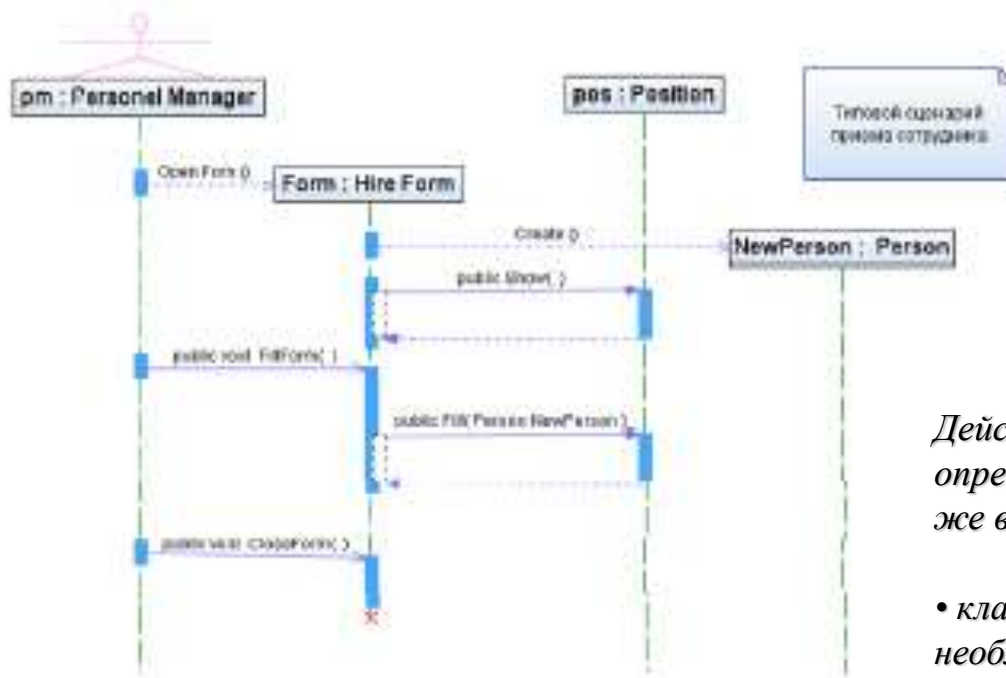
ликвидация
подразделения

создание вакансии

сокращение должности

Составление диаграмм использования — это первый шаг моделирования. Основное назначение диаграммы использования — **показать, что делает система** во внешнем мире. Диаграмма использования **не зависит от программной реализации системы** и поэтому не обязана соответствовать структуре классов, модулей и компонентов системы. Идентификация действующих лиц и вариантов использования — ключ к дальнейшему проектированию. В зависимости от выбранной парадигмы проектирования и программирования применяются различные способы реализации вариантов использования.

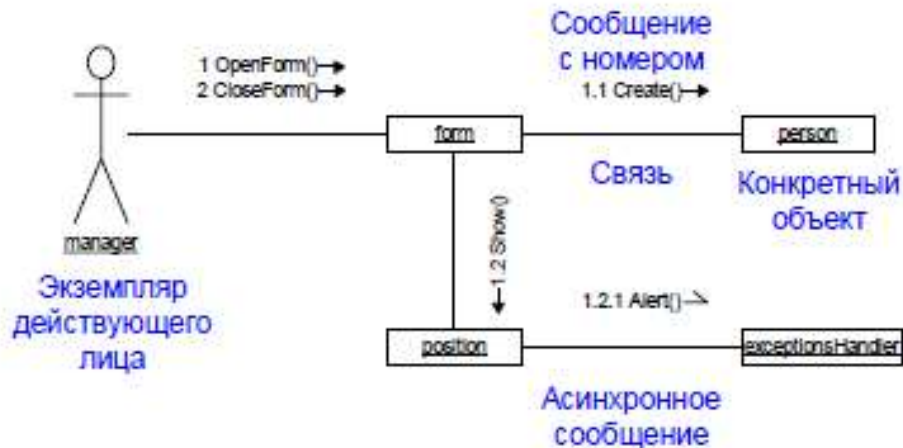
Диаграмма последовательности для типового сценария приема сотрудника информационной системы отдела кадров



Действующее лицо *PersonnelManager* уже было определено при моделировании использования, здесь же в модели появились новые сущности:

- класс *HireForm*, ответственный за интерфейс, необходимый для выполнения варианта использования прием сотрудника;
- класс *Person*, ответственный за хранение данных о конкретном человеке;
- класс *Position*, ответственный за хранение данных и выполнение операций с конкретной должностью.

Диаграмма кооперации для исключительной ситуации при приеме сотрудника информационной системы отдела кадров



Построение этой диаграммы выявило необходимость включения в модель (по крайней мере) еще одного класса — `ExceptionHandler`, который несет ответственность за обработку исключительных ситуаций.

МОДЕЛЬ ПРОЕКТИРОВАНИЯ

Назначение модели проектирования заключается в создании полного детализированного описания *внутренней архитектуры и алгоритмов работы* системы.

Рекомендуется разрабатывать данную модель без привязки к конкретным языкам программирования, с помощью которых будет создаваться программный продукт, т. е. разрабатывать логическую модель.

Построение этой модели необходимо:

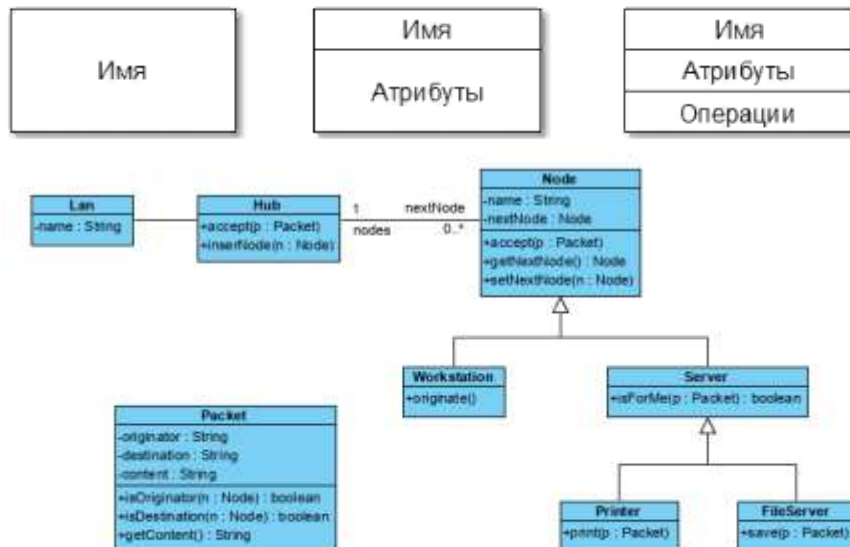
- для уточнения внутренней архитектуры и вариантов использования системы;
- для уточнения требований;
- для определения детализированных алгоритмов работы системы в целом и ее отдельных элементов.

Назначение структурного моделирования

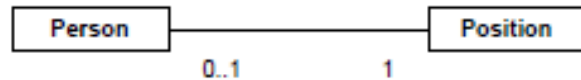
структура связей между объектами во время выполнения программы;

- *структура хранения данных;*
- *структура программного кода;*
- *структура компонентов в приложении;*
- *структура используемых вычислительных ресурсов;*
- *структура сложных объектов, состоящих из взаимодействующих частей;*
- *структура артефактов в проекте;*
- *Диаграммы компонентов.*

Диаграмма классов (Class diagram)



Состав **диаграммы классов** аналогичен составу **диаграммы классов анализа**. В то же время классы анализа должны пройти процедуру строгой экспертизы на предмет их возможной декомпозиции на более мелкие и специализированные классы. При построении диаграммы окончательно должны быть определены **атрибуты и операции классов**.



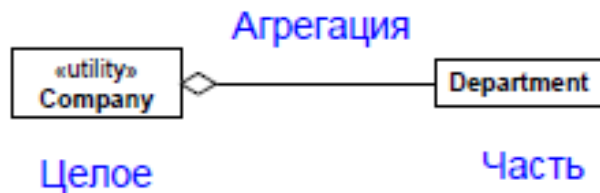
Кратность
задана
диапазоном

Кратность
задана
числом

Роль полюса ассоциации, называемая также спецификатором интерфейса — это способ указать, как именно участвует классификатор (присоединенный к данному полюсу ассоциации) в ассоциации. В общем случае данное дополнение имеет следующий синтаксис:

видимость ИМЯ : тип

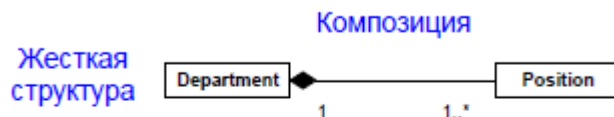
Имя является обязательным, оно называется именем роли и фактически является собственным именем полюса ассоциации, позволяющим различать полюса. Если рассматривается одна ассоциация, соединяющая два различных класса, то в именах ролей нет нужды: полюса ассоциации легко можно различить по именам классов, к которым они присоединены. Однако, если это не так, т. е. если два класса соединены несколькими ассоциациями, или же если ассоциация соединяет класс с самим собой, то указание роли полюса ассоциации является необходимым.



Агрегация

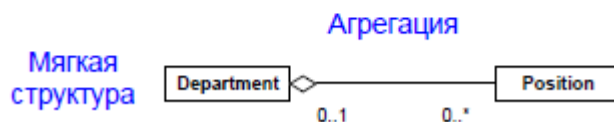
Целое

Часть



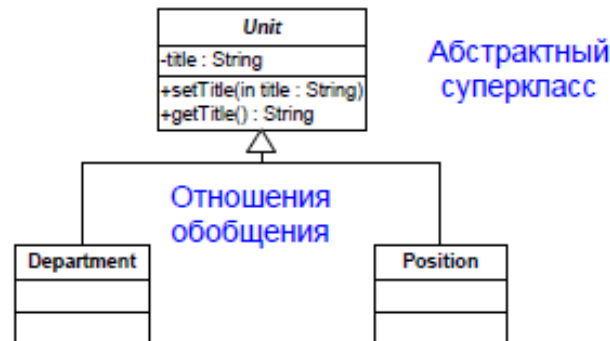
Композиция

Жесткая
структура



Агрегация

Мягкая
структура



Абстрактный
суперкласс

Отношения
обобщения

Конкретные подклассы

Метамодель ассоциации

online.mirea.ru

Спасибо за внимание!