

Технологии проектирования ИС и ИТ

ФИО преподавателя: Смирнов Михаил Вячеславович

e-mail: smirnov.mirea@gmail.com

Лекция 7

Объектно-ориентированный анализ и проектирование. Основные спецификации UML.

Содержание

- ▶ Определение и способы применения диаграмм UML
- ▶ Нотация и метамодель
- ▶ Классификация диаграмм UML
- ▶ Общее описание диаграмм UML
- ▶ Допустимый UML и его расширяемость
- ▶ Диаграмма классов UML
- ▶ Диаграмма прецедентов UML
- ▶ Диаграмма последовательности UML

Что такое UML?

UML (Unified Model Language) – семейство графических нотаций в основе которого лежит единая метамодель.

Создан, чтобы помогать в описании и проектировании информационных систем любой сложности.

Считается графическим языком моделирования.

Используется в качестве инструмента обратного и прямого инжиниринга в CASE-средствах разработки информационных систем.

Способы применения UML

Существует три основных **режима** использования UML диаграмм:

- режим эскиза
- режим проектирования
- режим языка программирования

У режима языка программирования есть две опции: **прямой** (диаграммы до кода) и обратный (диаграммы на основании кода) **инжиниринг**.

Исполняемый UML и MDA

MDA (Model Driven Architecture) – архитектура, управляемая моделью.

Основой является UML диаграммы.

В MDA существует **два состояния**:

PIM (Platform Independent Model) – модель, не зависящая от платформы.

PSM (Platform Specific Model) – модель, зависящая от платформы.

В “**исполняемом UML**” PSM заменена на компилятор модели.

Группа OMG и стандартизация UML

OMG – организация, занимающаяся созданием методологии на основе существующих стандартов. С момента появления первой версии UML, методологии UML создавались в OMG.



<https://www.omg.org/UML/>

Понятие нотации

Нотация – это совокупность графических элементов, которые применяются при моделировании.

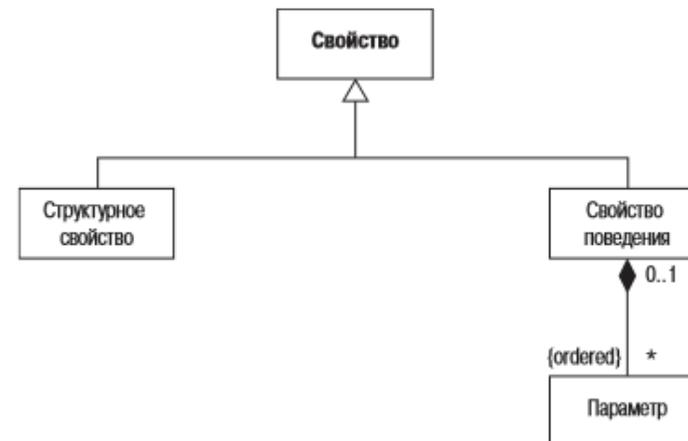
Нотация – синтаксис языка моделирования.

Самая по себе нотация (в том числе и UML) является **нестрогим** набором правил моделирования.

Объекты, создаваемые в языке UML с использованием нотаций называются **диаграммы**.

Понятие метамодели

Большей строгости и соответствия стандарту моделирования (зачастую жертвуя практической полезностью) достигают определением метамодели (ниже показан общий вид метамодели для диаграммы классов).



Классификация типов диаграмм UML

Актуальная версия UML 2.5.1. описывает 15 типов используемых диаграмм с соответствующими им нотациями.

Все диаграммы разделены на две основные группы – Диаграммы структуры и диаграммы поведения.

<https://www.uml-diagrams.org/uml-25-diagrams.html>

Допустимый UML и эскизы

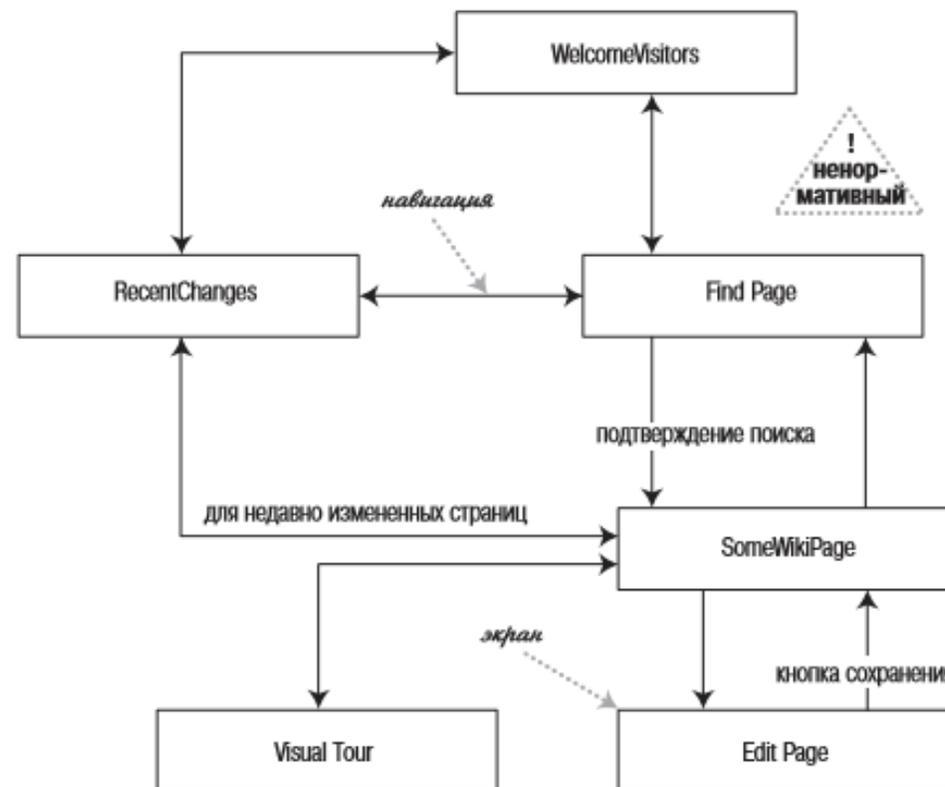
Существует две версии правил UML:

- предписывающие (официальная основа)
- описательные (применение на практике)

При применении UML в качестве средства UML правила рассматриваются строго как предписывающие.

Расширяемость UML, дополнения

Нормальной практикой является, при необходимости, дополнять существующие диаграммы UML (описательные) ненормативными диаграммами.



Расширяемость UML, дополнения

Так выглядит ненормативная таблица решений, позволяющая пояснить логику обработки задачи из нескольких переменных.

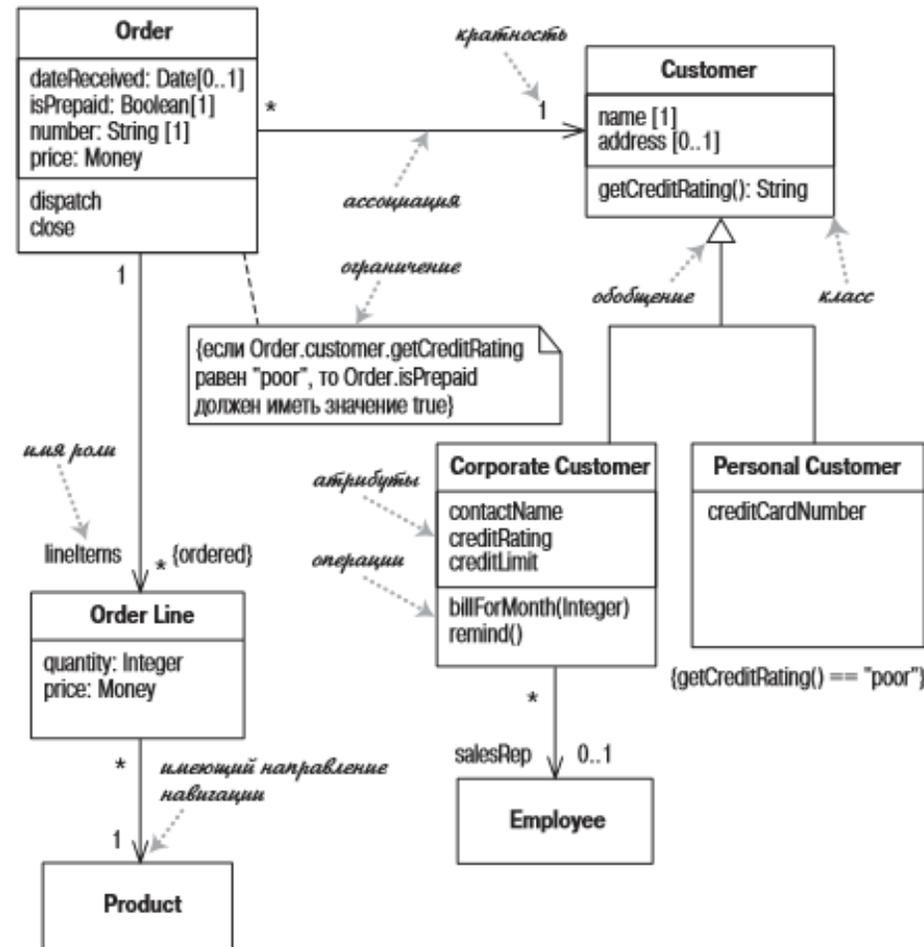
| | | | | | | |
|--------------------------|-------|-------|------|------|------|------|
| Специальный клиент | X | X | Y | Y | N | N |
| Приоритетный заказ | Y | N | Y | N | Y | N |
| Международный заказ | Y | Y | N | N | N | N |
| Плата | \$150 | \$100 | \$70 | \$50 | \$80 | \$60 |
| Предупредительный сигнал | . | . | . | | | |

Диаграммы классов

Описывает типы объектов системы и различного рода **статические** отношения, которые существуют между ними.

Помимо самой структуры классов и их названий, на диаграмме отображаются **атрибуты классов, методы классов и ограничения классов.**

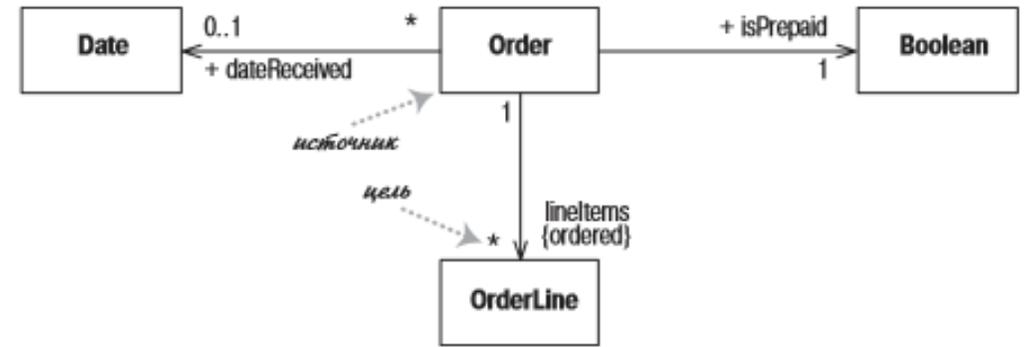
Пример диаграммы классов



Ассоциации классов

Ассоциации обозначаются сплошными линиями, стрелками, направленными в одном или двух направлениях. Каждая ассоциация описывается **кратностью**.

Кратность – это количество объектов, которые могут заполнять данное свойство. Как правило, кратности обозначаются следующими символами: **1, 0..1, ***



Методы классов

Структура: видимость имя (список параметров) : возвращаемый тип {свойства}, где:

- видимость, показывает открытая или закрытая операция;
- список параметров, параметры операции;
- возвращаемый тип, тип возвращаемого значения (если оно вообще есть);
- строка свойств, свойства применяемые к операции

Например, проверка баланса: + balanceOn (date: Date) : money

Диаграмма прецедентов (use-case)

Статичная диаграмма, показывающая взаимодействия между пользователями системы и самой системой, в рамках которого пользователи реализуют функционал системы.

Это один из наиболее распространенных способов определения функциональных требований к системе на этапе ее разработки и ознакомления заказчика с ними.

Содержит границы системы, актеров и прецеденты.

Этапы создания диаграммы прецедентов

Для комфортного создания диаграммы имеет смысл придерживаться последовательности трех этапов:

- составление сценария;
- составление текста прецедента с расширениями;
- непосредственно изображение прецедентов на диаграмме.

Сценарий для диаграммы

Последовательность шагов, описывающих взаимодействия пользователя и системы.

Например, в сценарии “Покупка товара”:

Покупатель просматривает каталог и помещает выбранные товары в корзину. При желании оплатить покупку он вводит информацию о кредитной карте и производит платеж. Система проверяет авторизацию кредитной карты и подтверждает оплату товара формой, дублируя сообщение электронной почтой.

Текст прецедента

В тексте описывается главный успешный сценарий прецедента.

После этого приводятся расширения (ответвления или вариации) прецедента.

Главный успешный сценарий:

1. Покупатель просматривает каталог и выбирает товары для покупки.
2. Покупатель оценивает стоимость всех товаров.
3. Покупатель вводит информацию, необходимую для доставки товара (адрес, доставка на следующий день или в течение трех дней).
4. Система предоставляет полную информацию о цене товара и его доставке.
5. Покупатель вводит информацию о кредитной карточке.
6. Система осуществляет авторизацию счета покупателя.
7. Система подтверждает оплату товаров немедленно.
8. Система посылает подтверждение оплаты товаров по адресу электронной почты покупателя.

Расширения:

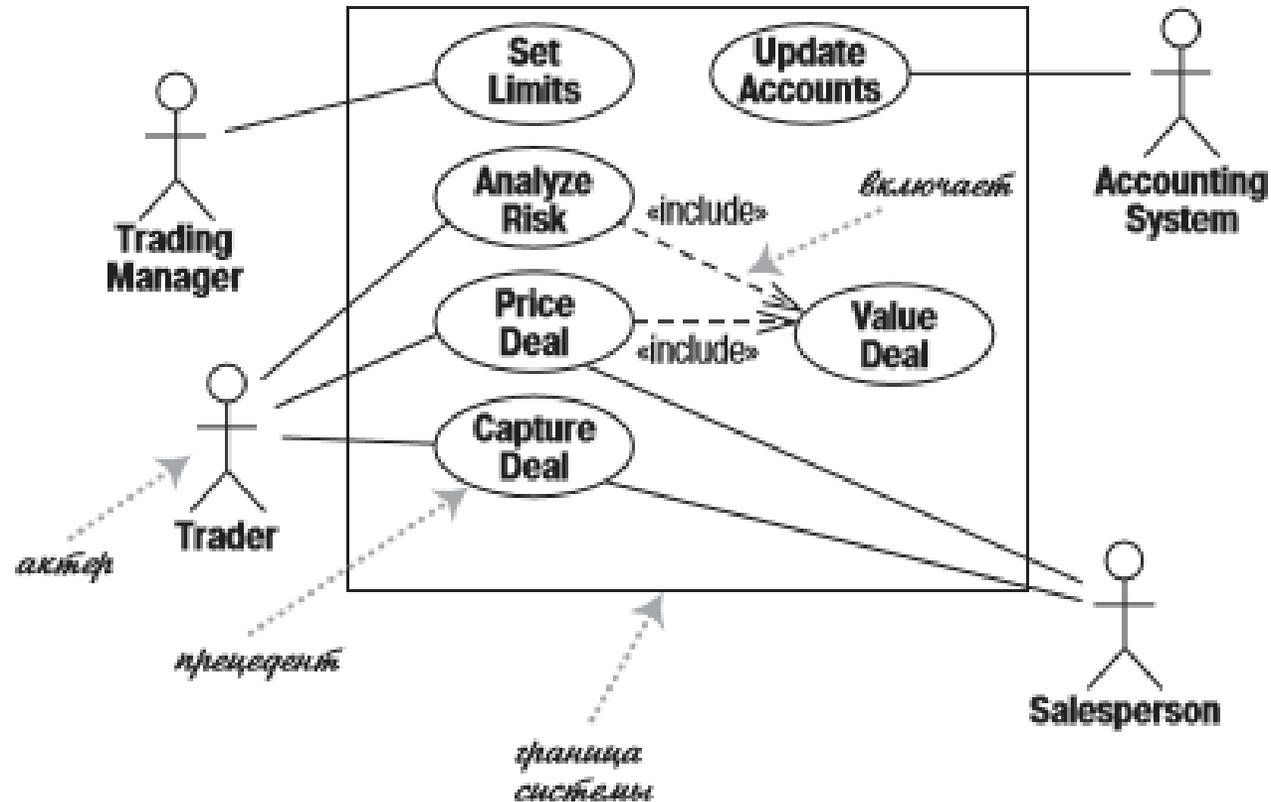
3а. Клиент является постоянным покупателем.

- .1: Система предоставляет информацию о текущей покупке и ее цене, а также информацию о счете.
- .2: Покупатель может согласиться или изменить значения по умолчанию, затем возвращаемся к шагу 6 главного успешного сценария.

6а. Система не подтверждает авторизацию счета.

- .1: Пользователь может повторить ввод информации о кредитной карте или закончить сеанс.

Пример диаграммы прецедентов



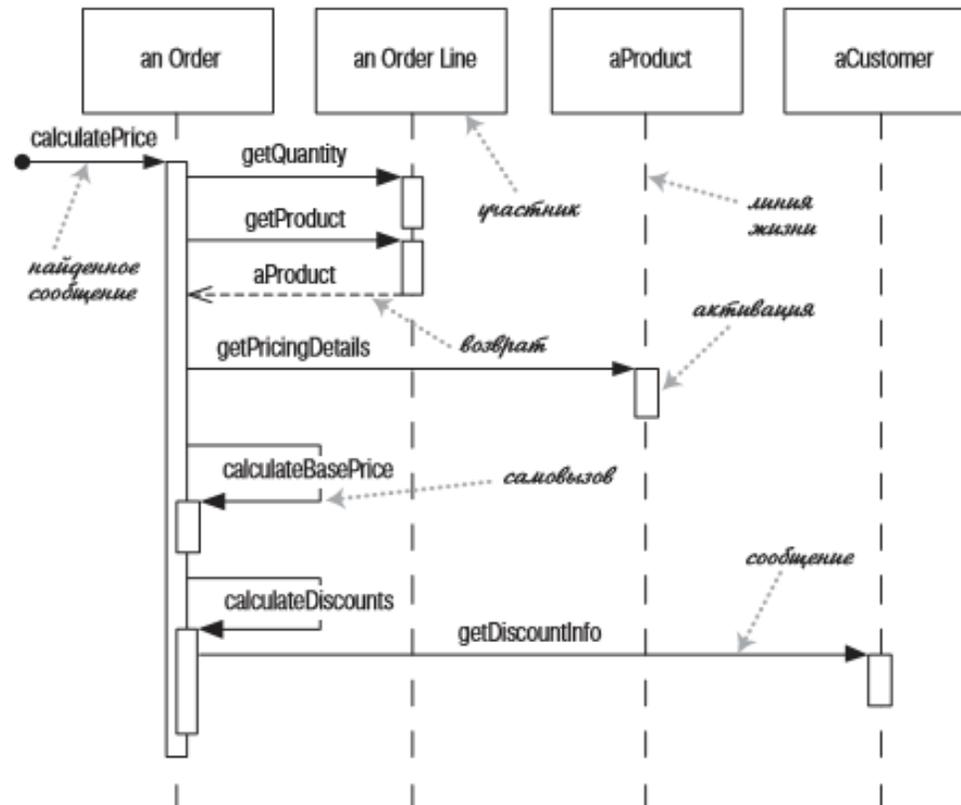
Диаграммы последовательности

Динамические диаграммы, описывающие взаимодействие групп объектов ИС (преимущественно классов) в различных условиях их поведения.

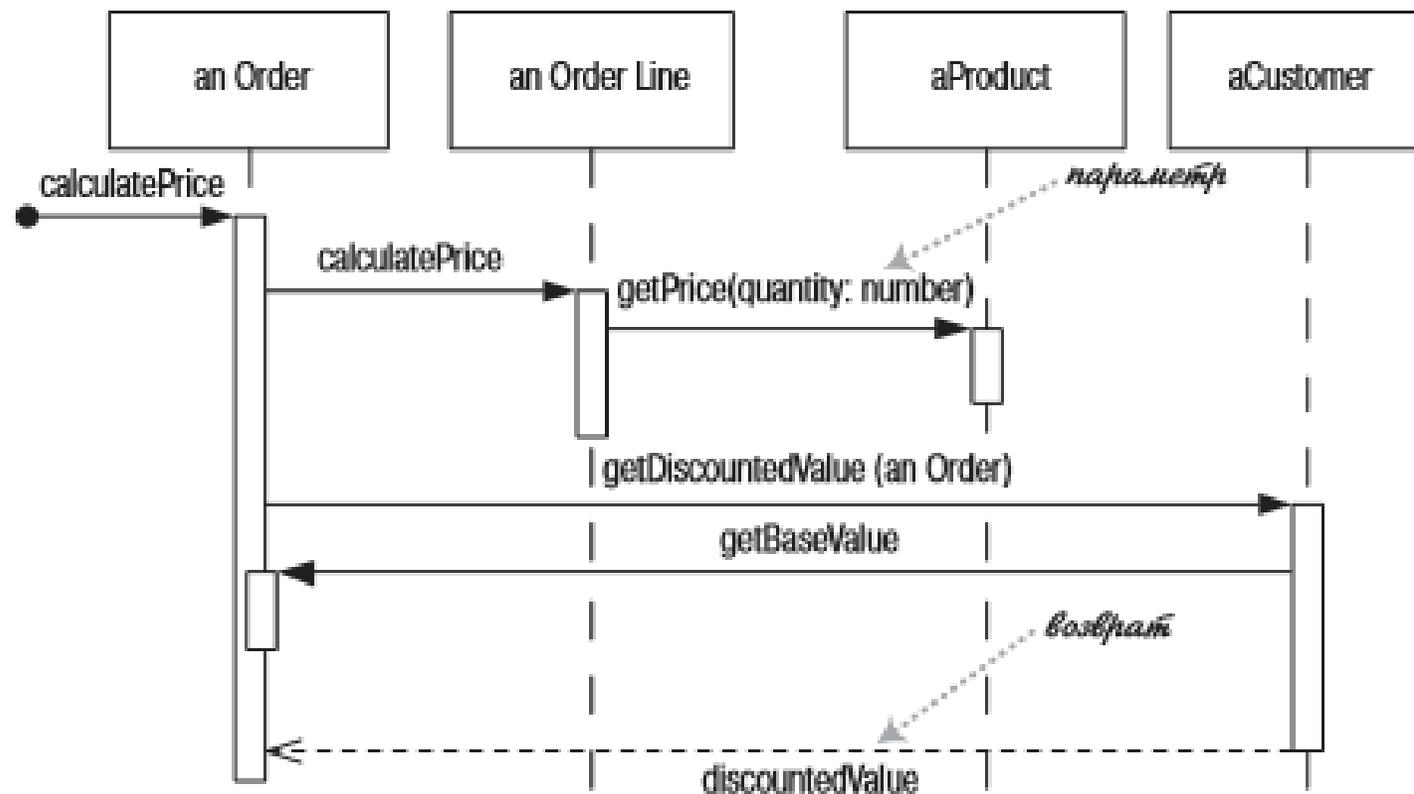
Правила отображения: один сценарий – одна диаграмма.

Ключевые элементы: участники, линии жизни, сообщения, активации.

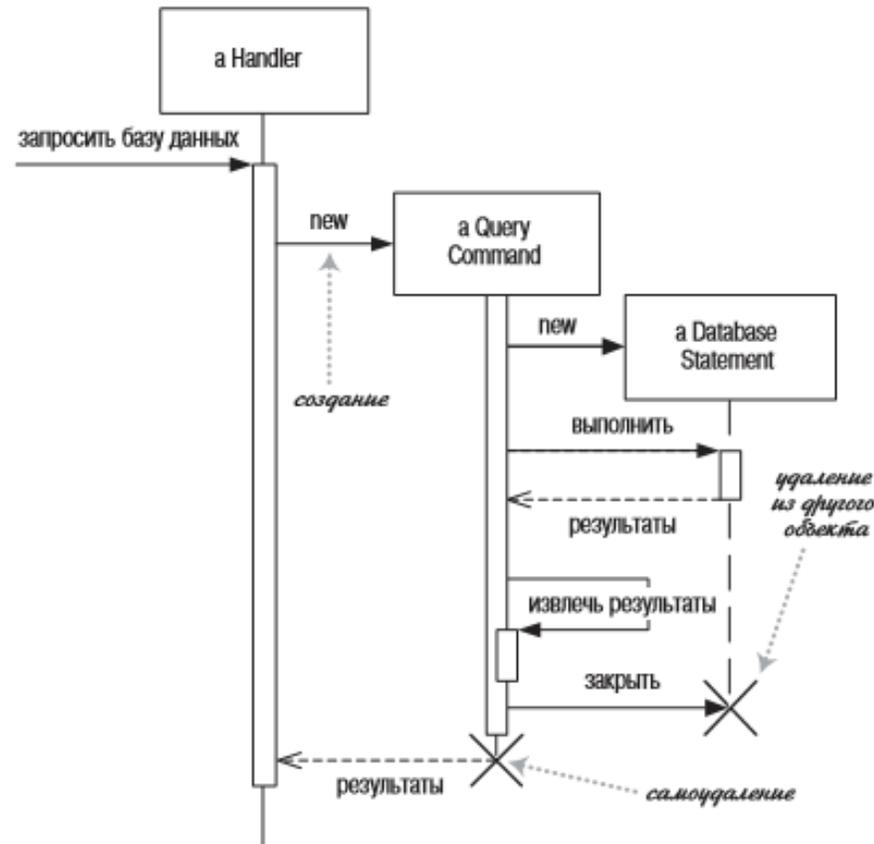
Диаграммы последовательности (централизованное управление)



Диаграммы последовательности (распределенное управление)



Создание и удаление участников диаграммы последовательности



Фреймы взаимодействия

```

foreach (lineitem)
  if (product.value > $10K)
    careful.dispatch
  else
    regular.dispatch
  end if
end for
if (needsConfirmation) messenger.confirm
end procedure
  
```

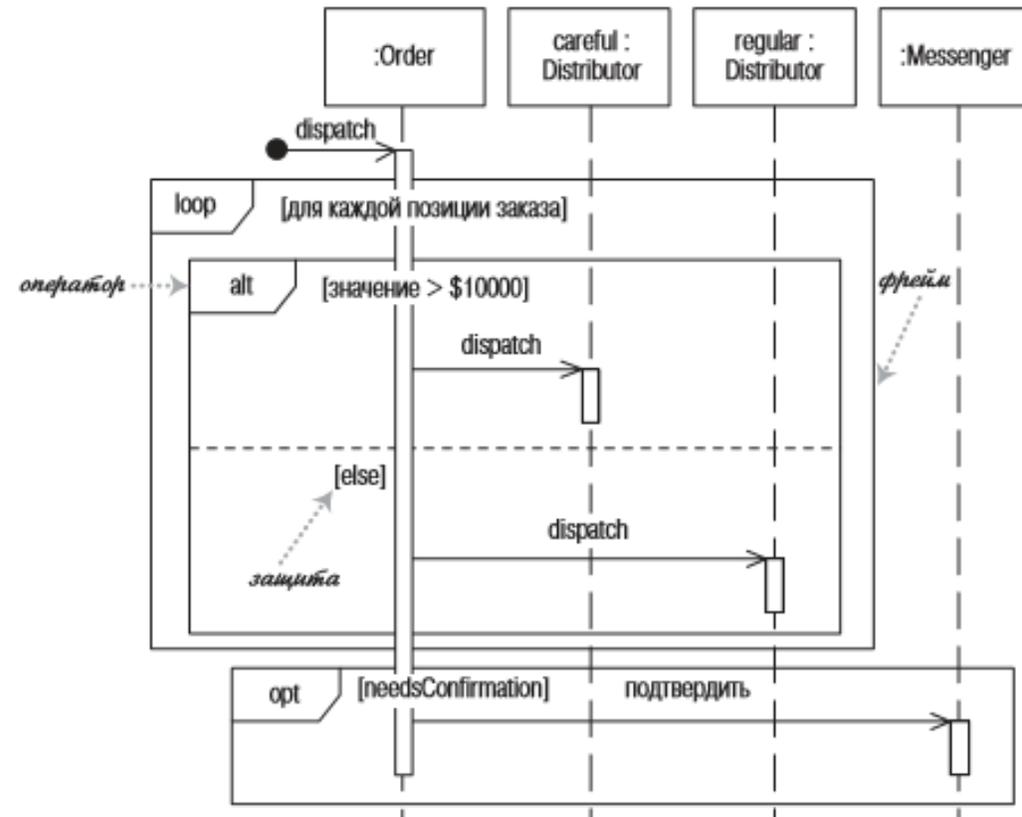


Таблица фреймов

| | |
|------|--|
| Alt | Из нескольких выполняется только истинный |
| Opt | Необязательный фрагмент, выполняющийся только при истинном условии |
| Par | Все фрагменты выполняются параллельно |
| Loop | Классический цикл, тело итерации |
| Neg | Неверное взаимодействие |
| Ref | Ссылка на взаимодействие, выполняющееся на другой диаграмме |

Статьи для самостоятельного изучения

- Материалы лекции 7 на сайте <http://msuniversity.ru>
- М. Фаулер, Справочник – UML. Основы. <http://msuniversity.ru>
- Технологии разработки программного обеспечения. Глава 8. Основы объектно-ориентированного представления программных систем, стр. 187-215 <http://msuniversity.ru>
- Технологии разработки программного обеспечения. Глава 9. Объектно-ориентированная разработка требований, стр. 218-267 <http://msuniversity.ru>
- Актуальная спецификация UML на сайте OMG <https://www.omg.org/spec/UML/>
- Краткое описание диаграмм стандарта UML 2.5 (англ.) <https://www.uml-diagrams.org/uml-25-diagrams.html>
- Хабростатья. Отношения классов – от UML к коду. <https://habr.com/ru/post/150041/>

Спасибо за внимание!