

# Визуальное моделирование, UML

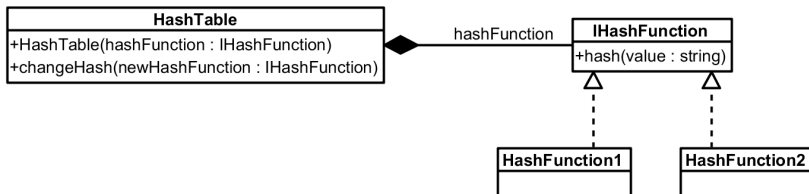
Юрий Литвинов  
yurii.litvinov@gmail.com

15.05.2018г

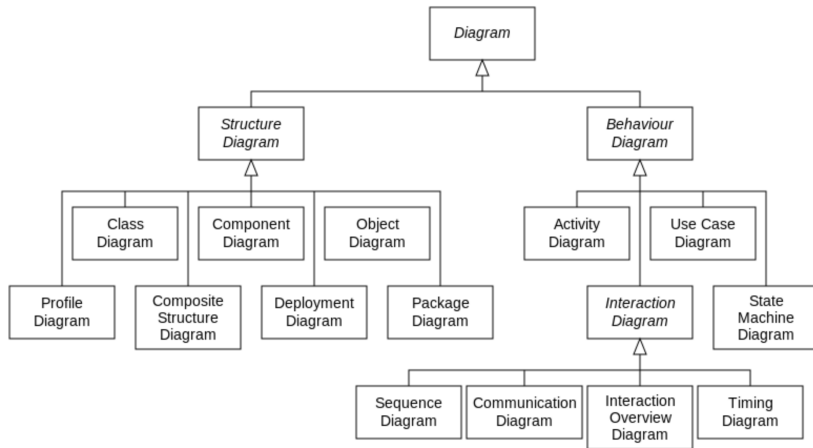
# Визуальное моделирование

- ▶ Модели
- ▶ Метафора моделирования
- ▶ Цель моделирования

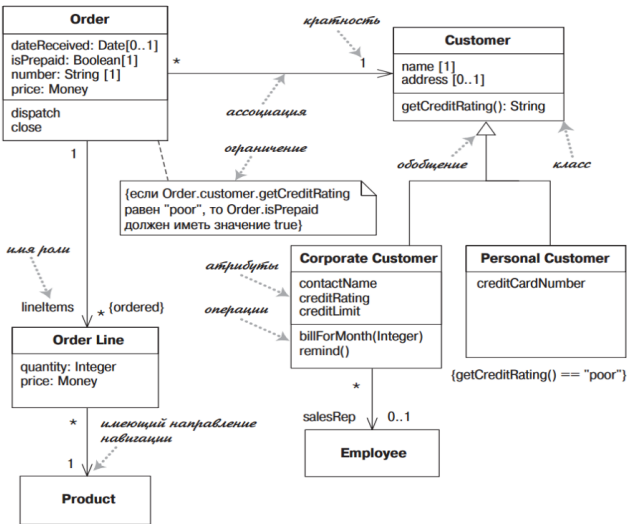
# Пример модели



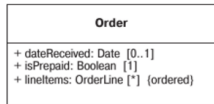
# Язык UML



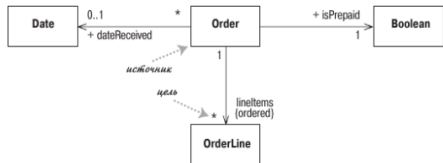
# Диаграммы классов UML



# Свойства



## Атрибуты



## Ассоциации

### Синтаксис:

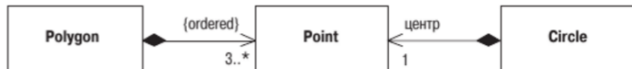
- ▶ видимость имя: тип кратность = значение по умолчанию {строка свойств}
- ▶ Видимость: + (public), - (private), # (protected), ~(package)
- ▶ Кратность: 1 (ровно 1 объект), 0..1 (ни одного или один), \* (сколько угодно), 1..\*, 2..\*

## Агрегация и композиция

Агрегация – объект “знает” о другом (не управляет его временем жизни, имеет на него ссылку или указатель)

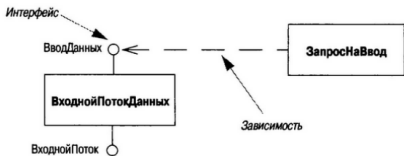
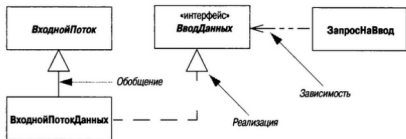


Композиция — объект владеет другим объектом (управляет его временем жизни, хранит его по значению или по указателю, делая delete)

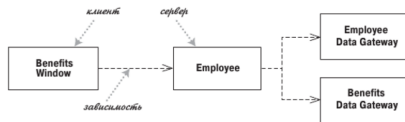


Уточнение обычной ассоциации, используется только если очень надо

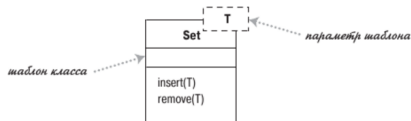
## Интерфейсы



## Зависимости



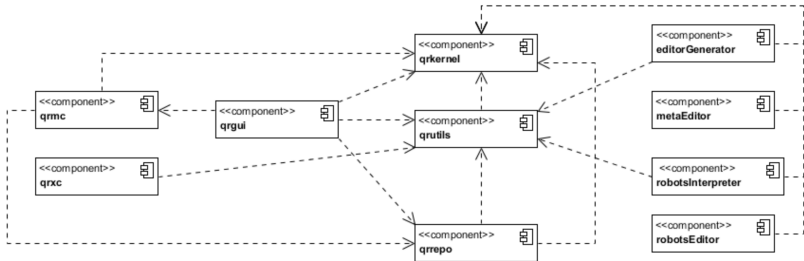
## Шаблоны





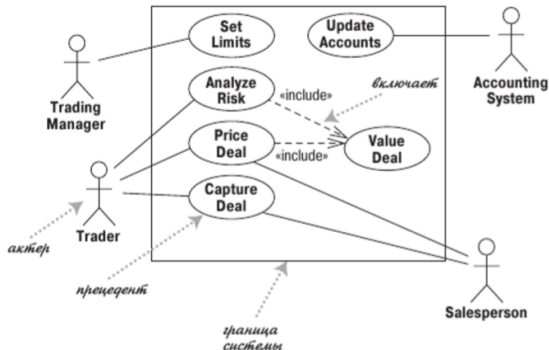
# Диаграммы компонентов

## Component diagrams



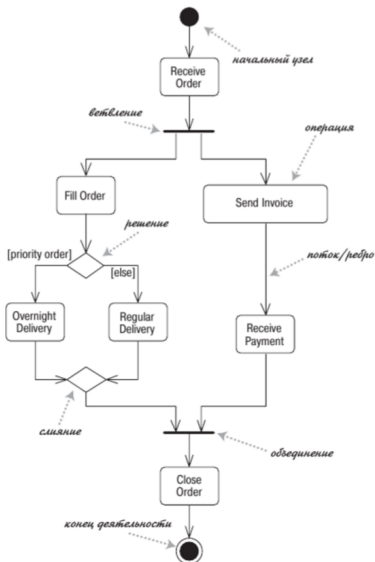
# Диаграммы случаев использования

## Use case diagrams

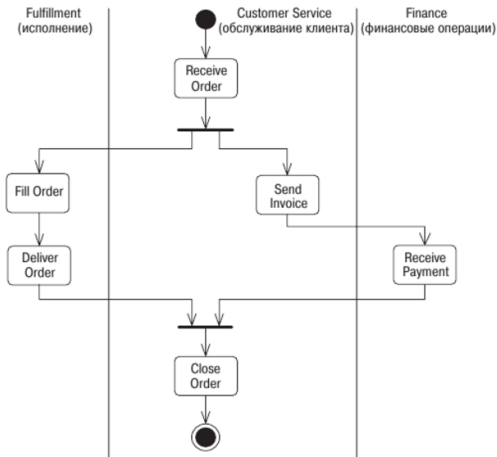


# Диаграммы активностей

## Activity diagrams

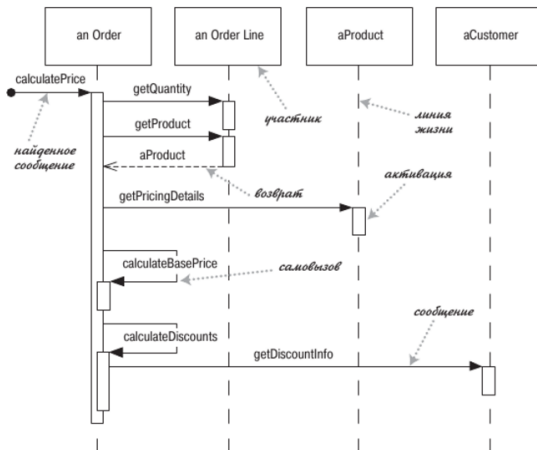


# Диаграммы активностей, разделы Swimlanes

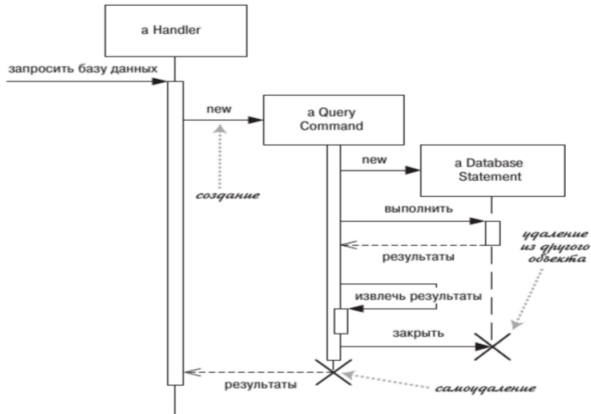


# Диаграммы последовательностей

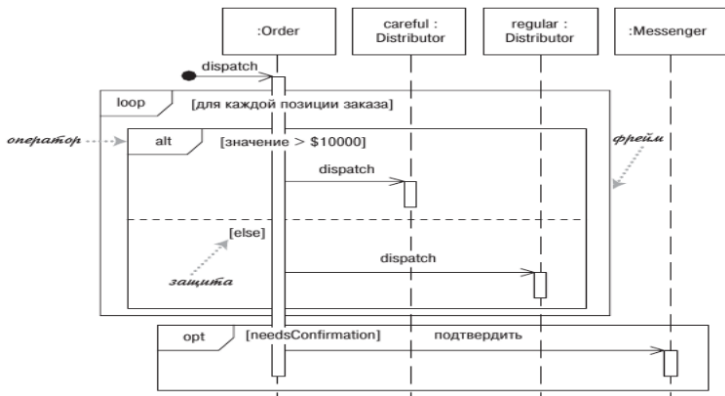
## Sequence diagrams



# Диаграммы последовательностей, создание и удаление объектов



# Диаграммы последовательностей, фреймы



# Диаграммы конечных автоматов

## State diagrams

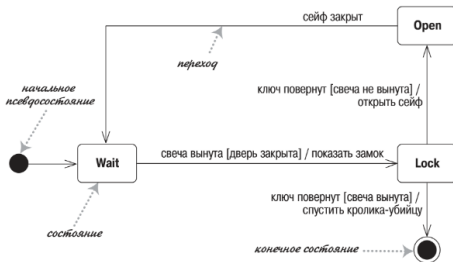
### ► Состояние

- entry activity
- exit activity
- do activity
- внутренний переход

### ► Событие

### ► Переход

- имя события (список параметров) [сторожевое условие] выражение действия



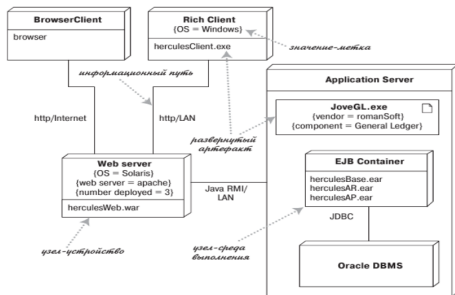
© М. Фаулер, UML. Основы



# Диаграммы развёртывания

## Deployment diagrams

- ▶ Показывает отображение компонентов и физических артефактов на реальные (или виртуальные) устройства
- ▶ Бывает полезна на начальных этапах проектирования, даже до диаграмм компонентов



© М. Фаулер, UML. Основы

# Предметно-ориентированные языки

