

# Лекция 4: Моделирование поведения

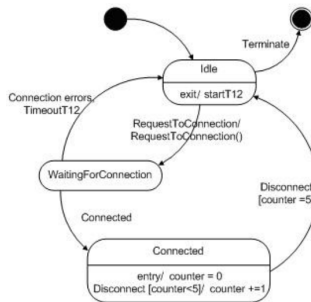
Юрий Литвинов  
yurii.litvinov@gmail.com

19.10.2017г

# Диаграммы конечных автоматов

## Диаграммы состояний

- ▶ Состояния объекта как часть жизненного цикла
- ▶ Моделирование реактивных объектов
  - ▶ Например, сетевое соединение
  - ▶ Или знакомый пример с торговым автоматом
- ▶ Имеют исполнимую семантику
- ▶ Д. Харел, 1987



# Диаграммы конечных автоматов, синтаксис

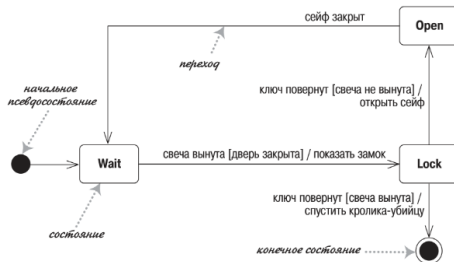
## ► Состояние

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

## ► Событие

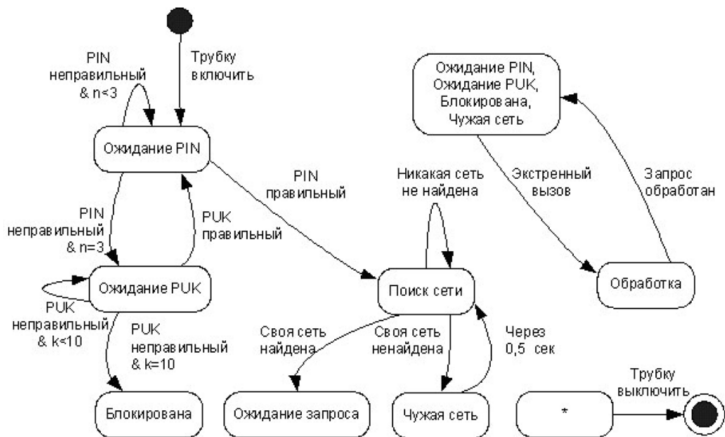
## ► Переход

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



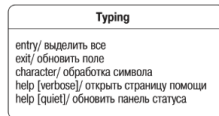
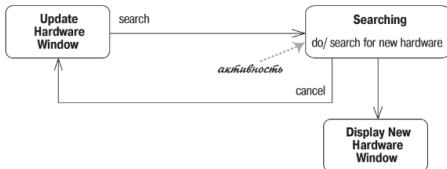
© М. Фаулер, UML. Основы

## Пример, мобильный телефон

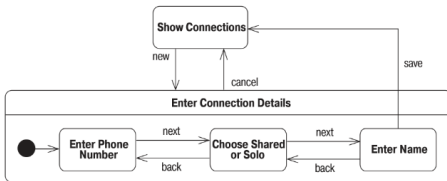


# Диаграммы конечных автоматов, прочие вещи

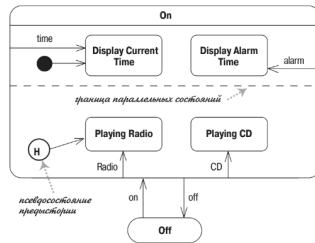
## Активности:



## Вложенные состояния:



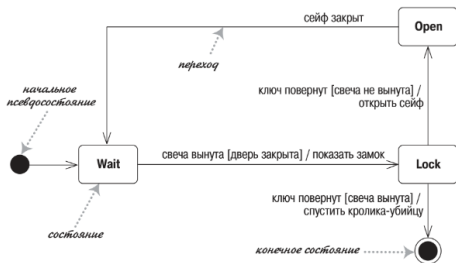
## Параллельные состояния, псевдосостояние истории:



© М. Фаулер, UML. Основы

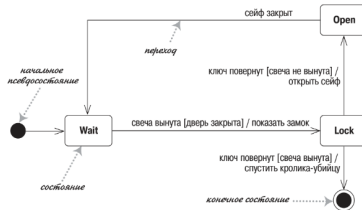
# Генерация кода

```
public void handleEvent(PanelEvent anEvent) {
    switch (currentState) {
        case PanelState.Open:
            switch (anEvent) {
                case PanelEvent.SafeClosed:
                    currentState = PanelState.Wait;
            }
            break;
        case PanelState.Wait:
            switch (anEvent) {
                case PanelEvent.CandleRemoved:
                    if (isDoorOpen) {
                        revealLock();
                        currentState = PanelState.Lock;
                    }
            }
            break;
        case PanelState.Lock:
            switch (anEvent) {
                case PanelEvent.KeyTurned:
                    if (isCandleIn) {
                        openSafe();
                        currentState = PanelState.Open;
                    } else {
                        releaseKillerRabbit();
                        currentState = PanelState.Final;
                    }
            }
            break;
    }
}
```



© М. Фаулер, UML. Основы

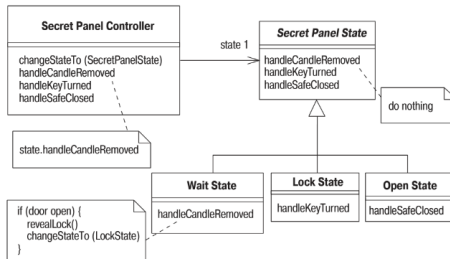
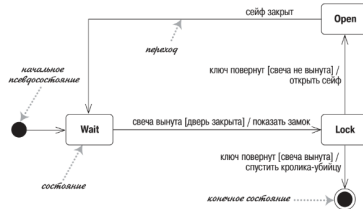
# Таблица состояний



Исходное состояние	Целевое состояние	Событие	Защита	Процедура
Wait	Lock	Candle removed (свеча удалена)	Door open (дверца открыта)	Reveal lock (показать замок)
Lock	Open	Key turned (ключ повернут)	Candle in (свеча на месте)	Open safe (открыть сейф)
Lock	Final	Key turned (ключ повернут)	Candle out (свеча удалена)	Release killer rabbit (освободить убийцу-кролика)
Open	Wait	Safe closed (сейф закрыт)		

© М. Фаулер, UML. Основы

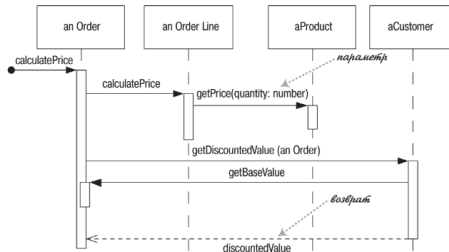
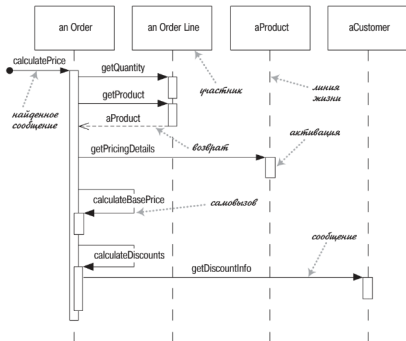
# Паттерн “Состояние”



© М. Фаулер, UML. Основы

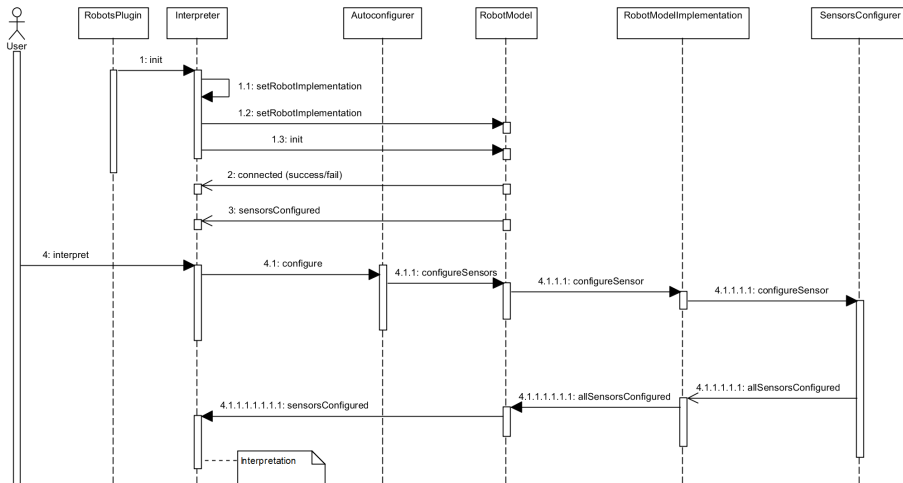


# Диаграммы последовательностей

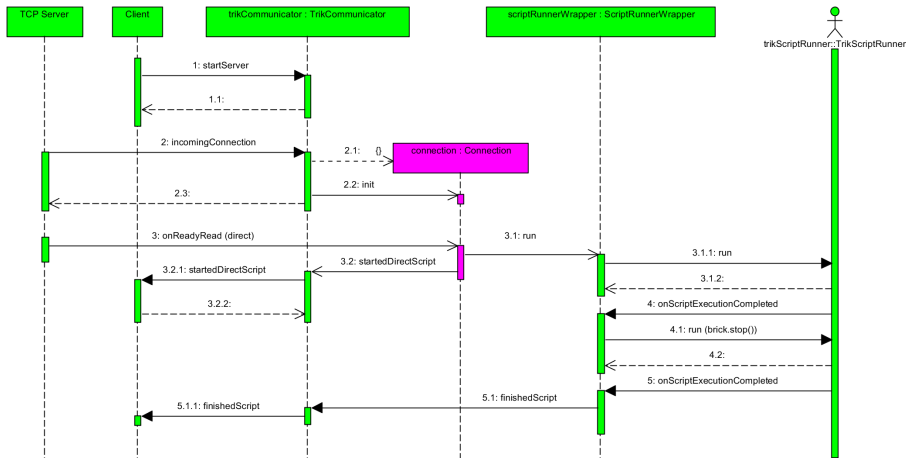


© М. Фаулер, UML. Основы

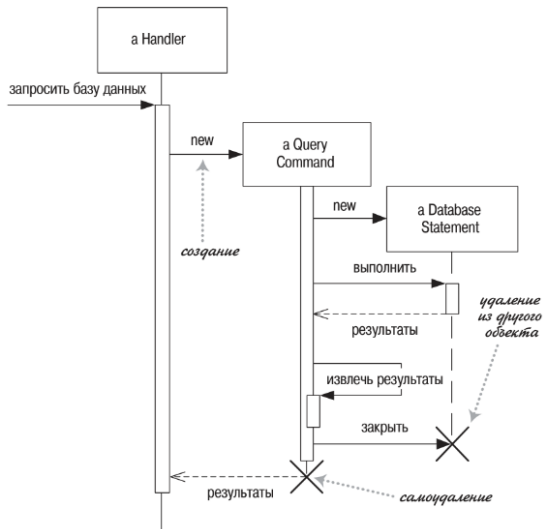
# Пример



# Ещё пример



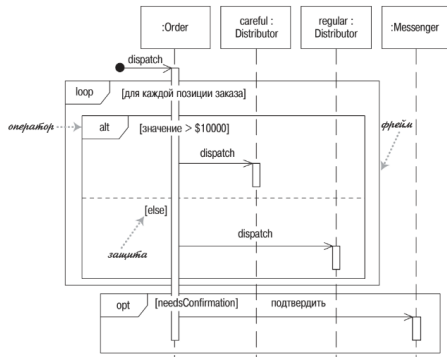
# Создание и удаление объектов



© М. Фаулер, UML. Основы

# Фреймы

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



© М. Фаулер, UML. Основы