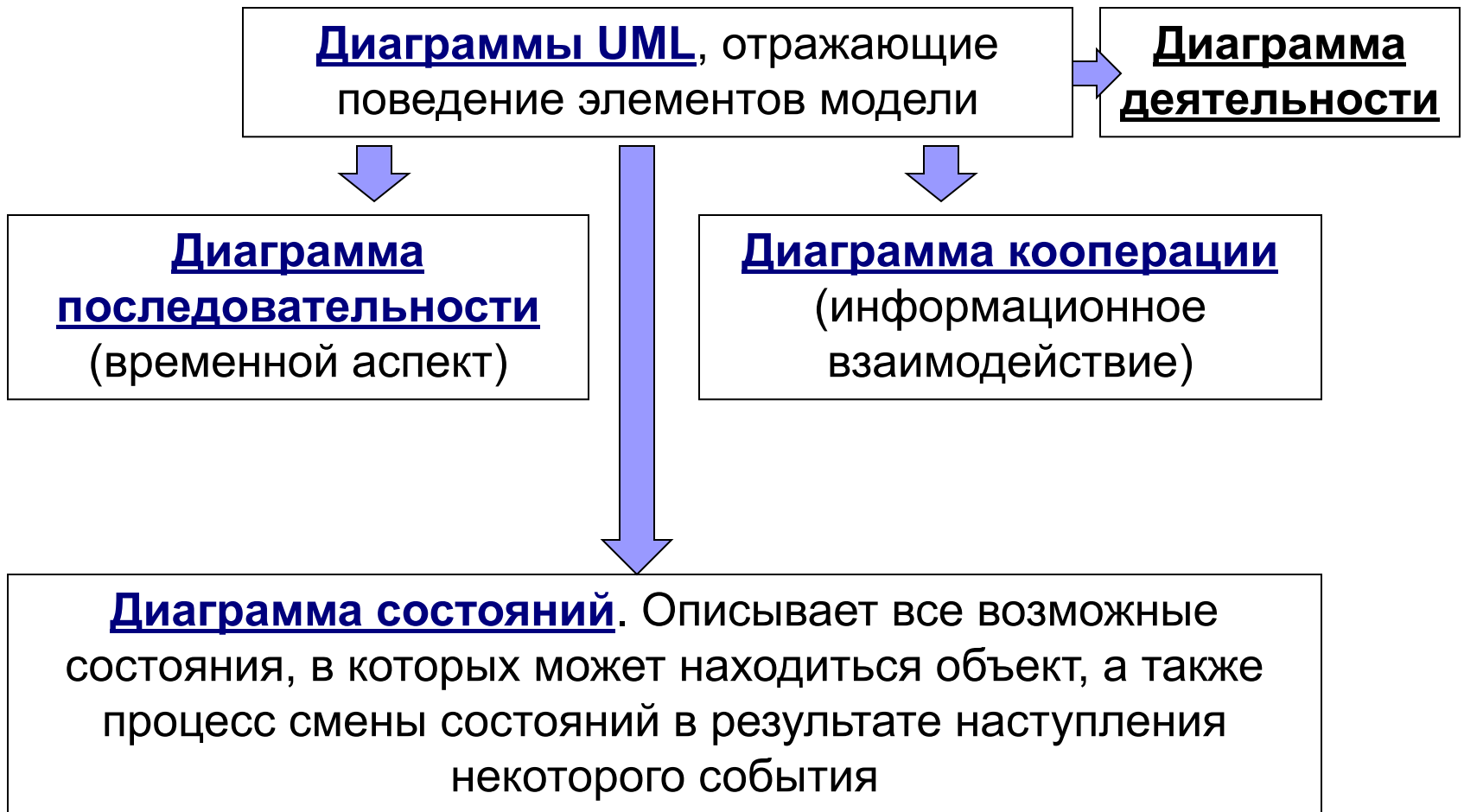


# Диаграмма СОСТОЯНИЙ

# Основные вопросы

- Сущность и назначение диаграммы состояний
- Основные компоненты
- Примеры

# Назначение диаграммы состояний



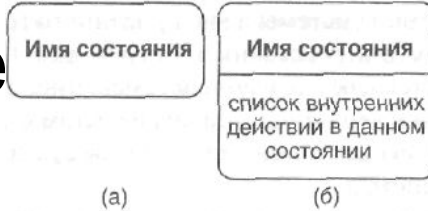
# Диаграмма состояний

- Диаграмма состояний (statechart diagram) используется для описания поведения объектов (отдельных экземпляров класса)
- Диаграмма состояний является графом специального вида, который представляет некоторый автомат. Вершинами этого графа являются состояния. Дуги графа служат для обозначения переходов из состояния в состояние.
- Переход объекта из состояния в состояние происходит в результате наступления некоторого события
- Смена состояний происходит мгновенно
- Диаграммы состояний могут быть вложены друг в друга. Состояние на диаграмме может быть описано с помощью другой диаграммы состояний.



# Элементы диаграммы состояний

■ Состояние



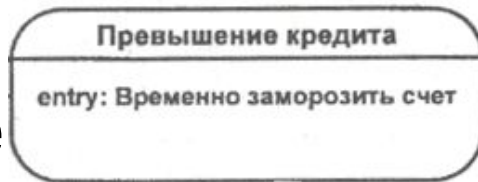
■ Начальное состояние



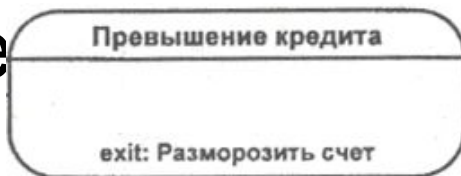
■ Конечное состояние



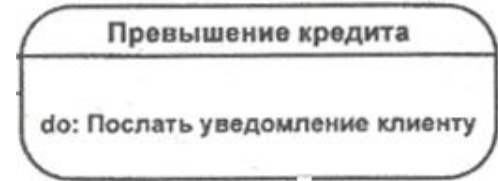
■ Входное действие



■ Выходное действие



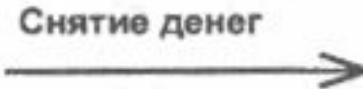
■ Внутренняя деятельность



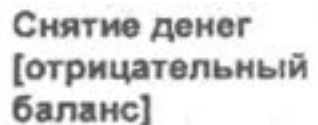
■ Переход



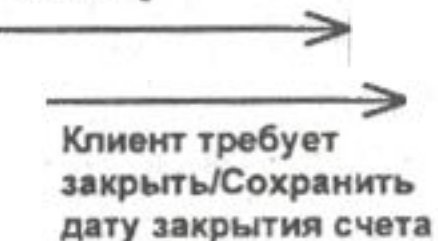
■ Событие



■ Ограничивающие условия



■ Действие



# Основные компоненты диаграммы состояний

Основные компоненты диаграммы состояний:

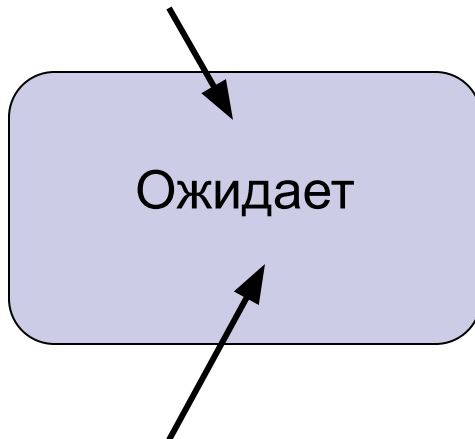
- *состояния;*
- *переходы.*

# Состояние

- Состояние (state) объекта определяется набором значений его атрибутов и связей с другими объектами
- Состояние – это некоторый период времени жизни объекта
- Объекты класса имеют конечное число возможных состояний
- Находясь в определенном состоянии объект воспринимает только определенные события, а другие игнорирует
- Находясь в некотором состоянии объект может совершать некоторую деятельность

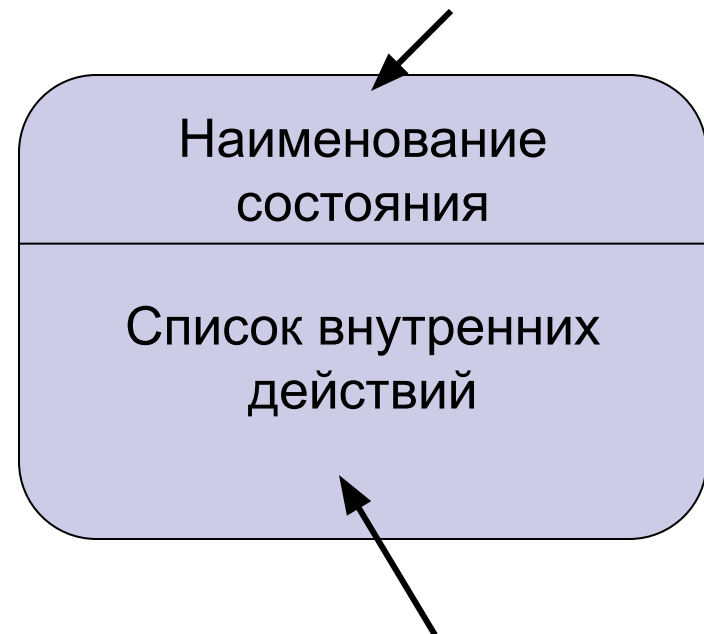
# Состояние

Имя состояния – законченное предложение, начинается с заглавной буквы



В качестве имени состояния используют глагол (звонит) или причастие (занят)

Секция имени



Список внутренних действий





синтаксис действия: имяСобытия/некотороеДействие  
 синтаксис деятельности: do/некотораяДеятельность

# Примеры состояний

Телефон звонит

Студент сдает сессию

Набор номера

Счет открыт

На счете нет денег

# Список внутренних действий

- Формат:

*<метка действия '/' выражение действия>*

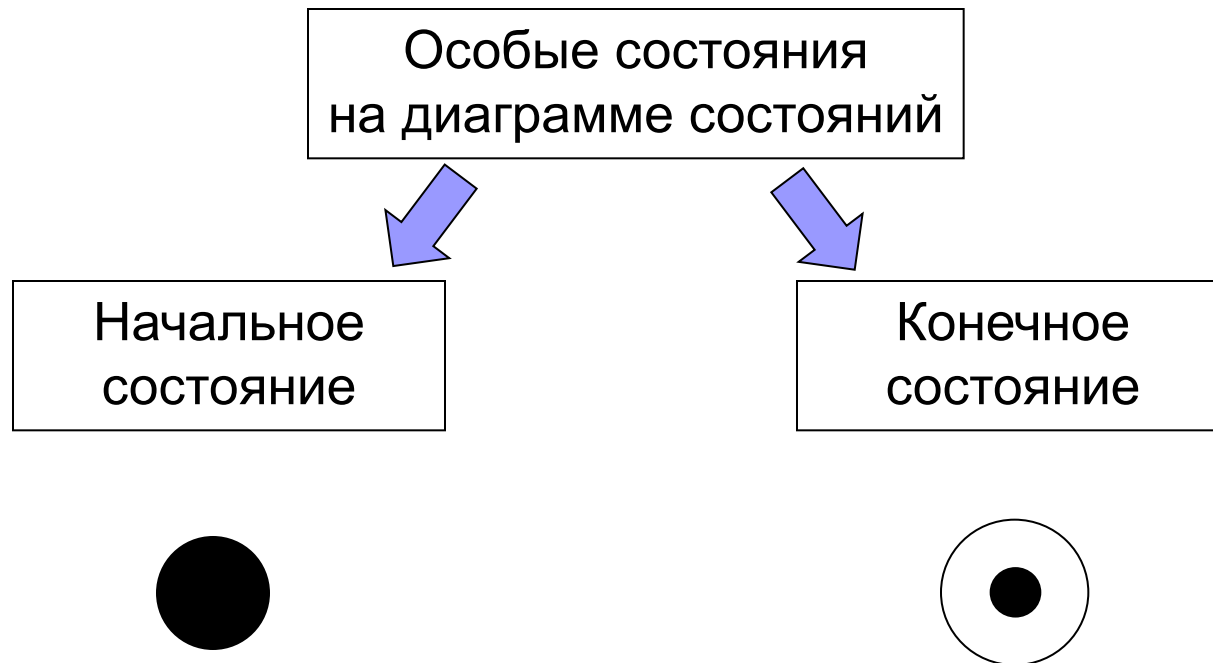
- Перечень меток в языке UML фиксирован:

- **entry;**
- **exit;**
- **do;**
- **include.**

## Ввод пароля

entry / сделать символы  
невидимыми  
символ / получить символ  
exit / сделать символы  
видимыми

# Начальное и конечное состояние



Начальное состояние указывается обязательно и оно должно быть одно. Конечных состояний может или не быть, или может быть несколько.

# Особые состояния

- Начальное состояние (initial state) – псевдосостояние, с которого начинается диаграмма состояний. Оно соответствует моменту создания объекта.
- Реально объект никогда не находится в начальном состоянии, а сразу переходит в следующее состояние
- Конечное состояние (final state) означает уничтожение объекта
- На диаграмме может быть несколько конечных состояний



Начальное состояние



Конечное состояние

# Переход

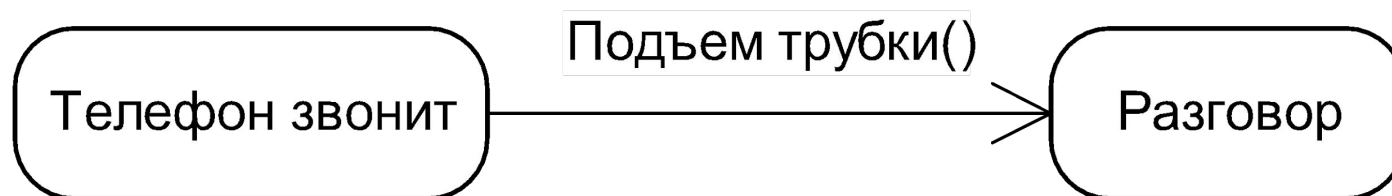
- Переход – отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния другим.
- Синтаксическая метка перехода состоит из трех частей, каждая из которых является необязательной:  
***<событие> [<условие>] / <действие>***

# Переход

- Переход – отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния другим.
- Синтаксическая метка перехода состоит из трех частей, каждая из которых является необязательной:  
***<событие> [<условие>] / <действие>***

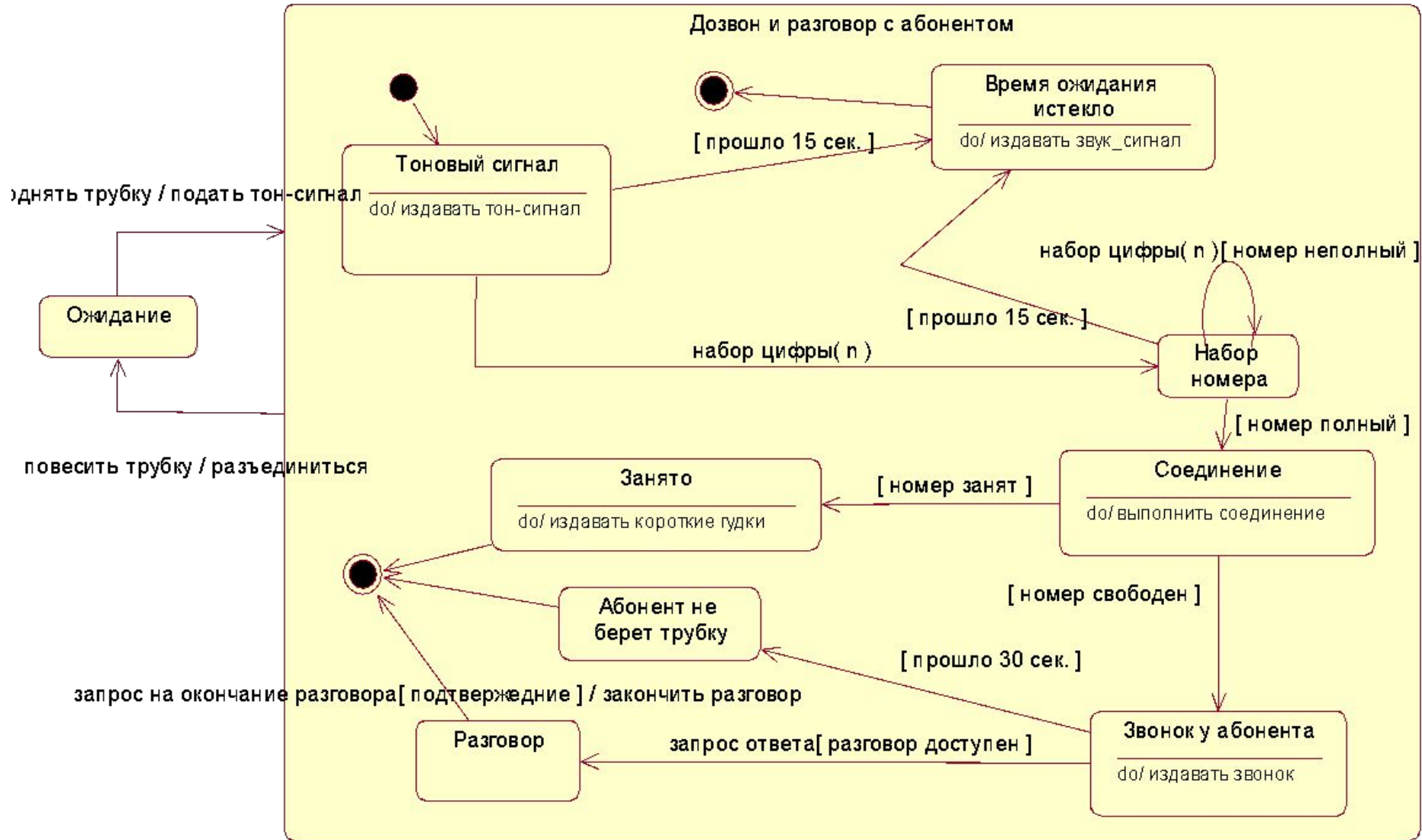
# Переходы

- Переход (transition) – это мгновенная смена одного состояния (исходного) на другое состояние (целевое)
- Переход запускается, когда происходит связанное с ним событие
- Переход может запускаться автоматически, когда заканчиваются действия, выполняемые в состоянии
- Исходное и целевое состояния могут совпадать
- Одно событие может запускать переходы во множестве объектов
- Например, из состояния “звонит” телефон переходит в состояние “разговор” при подъеме трубки





# Пример диаграммы состояний



# Действие, событие, условие

- **Действие**
- Действие (action) – это атомарное и, как правило, быстрое вычисление
- Действие может представлять собой последовательность более простых действий
- Действия могут выполняться:
  - При переходе из одного состояния в другое состояние
  - При входе в состояние (entry)
  - При выходе из состояния (exit)
  - При нахождении объекта в состоянии (do)
- В Microsoft Visio do-действия описываются внутренними переходами (internal transitions). Внутренний переход не меняет состояния и ему соответствует некоторое действие.

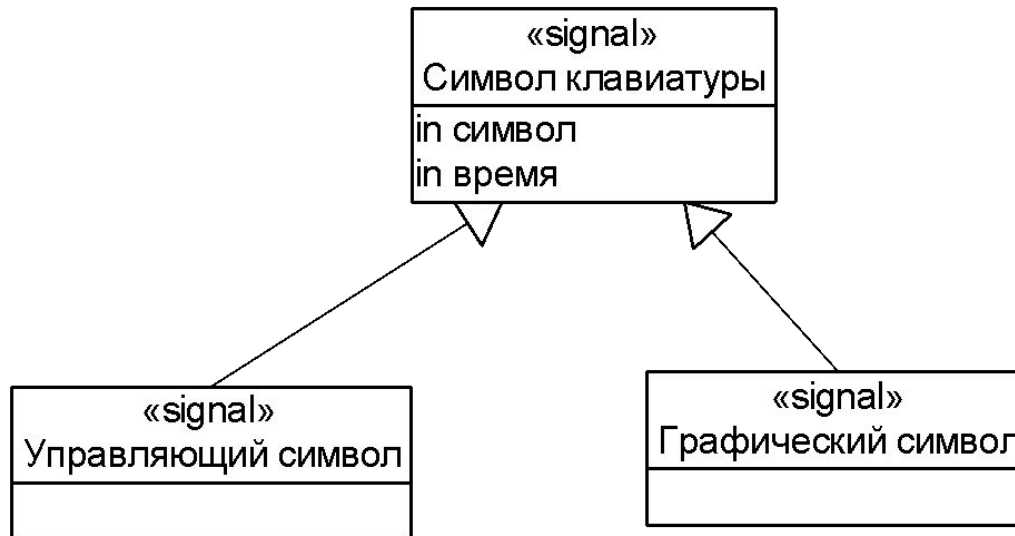
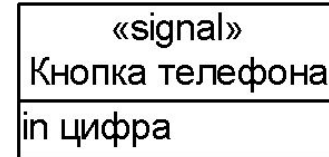
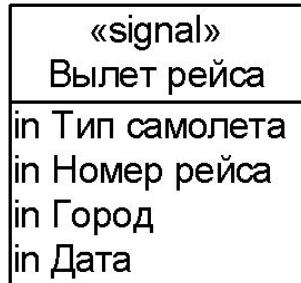
# Событие

- Событие (event) – некоторое явление, которое имеет определенное положение во времени и пространстве  
*Например, поступление денег на счет, нажатие кнопки, изменение температуры*
- Событие происходит в некоторый момент времени и не имеет продолжительности
- Событие влияет на состояния объектов системы
- Конкретное событие называется экземпляром события и может иметь свои собственные параметры
- События бывают четырех видов:
  - Событие сигнала
  - Событие вызова
  - Событие изменения
  - Событие времени

# Событие сигнала (signal event)

- Сигнал (signal) – это некоторая сущность, которая служит для передачи информации между объектами
- Сигнал имеет имя и набор параметров (атрибутов)
- Сигнал можно описать как класс со стереотипом “signal”
- Между сигналами могут быть отношения обобщения.
- Сигналы-потомки наследуют параметры своих предков и реализуют такие же переходы, что и их предки
- В Microsoft Visio сигнал нужно обязательно описать как класс, чтобы использовать его на диаграмме состояний

# Примеры событий сигналов



# Событие вызова (call event)

- Событие вызова – это вызов операции объекта
- С точки зрения вызывающего объекта (объекта-отправителя) такой вызов не отличим от обычного обращения к операции, которая реализуется методом
- Объект-получатель сам определяет как реализовать вызываемую операцию (метод или событие). Реализация в виде события означает переход из одного состояния в другое состояние.
- Параметры операции совпадают с параметрами события вызова
- В отличие от обычного обращения к операции событие вызова допускает параллельную работу объекта-отправителя и объекта-получателя
- Примеры событий:
  - Показать на форме список студентов курса (номер курса)
  - Нажать лампочку на кнопке телефонного аппарата (номер телефона, номер лампочки)
  - Положить на счет некоторую сумму (номер счета, сумма)

# Событие изменения (change event)

- Событие изменения наступает, когда некоторое логическое выражение принимает значение истина.
- Некоторое логическое выражение постоянно проверяется и, когда оно меняет свое значение с FALSE на TRUE, происходит событие изменения
- Примеры событий
  - when (температура в комнате < минимально допустимая)
  - when (давление в шинах < минимально допустимое)
  - when (давление газа > максимально допустимое)

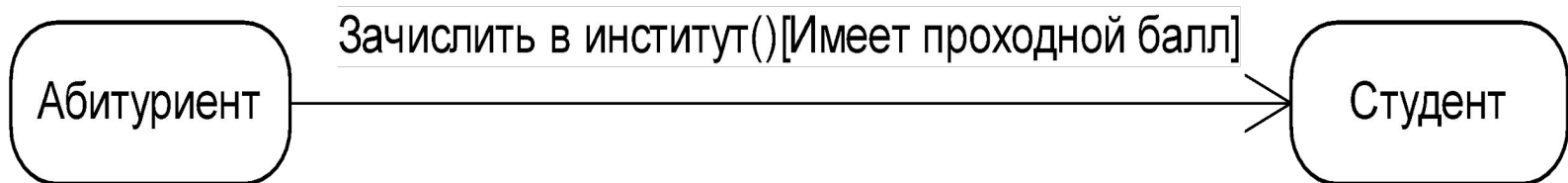
# Событие времени (time event)

- Событие времени – это событие, которое наступает в определенный момент времени или по окончании некоторого интервала времени
- Примеры событий
  - when (дата = 1 января текущего года)
  - after (10 секунд)

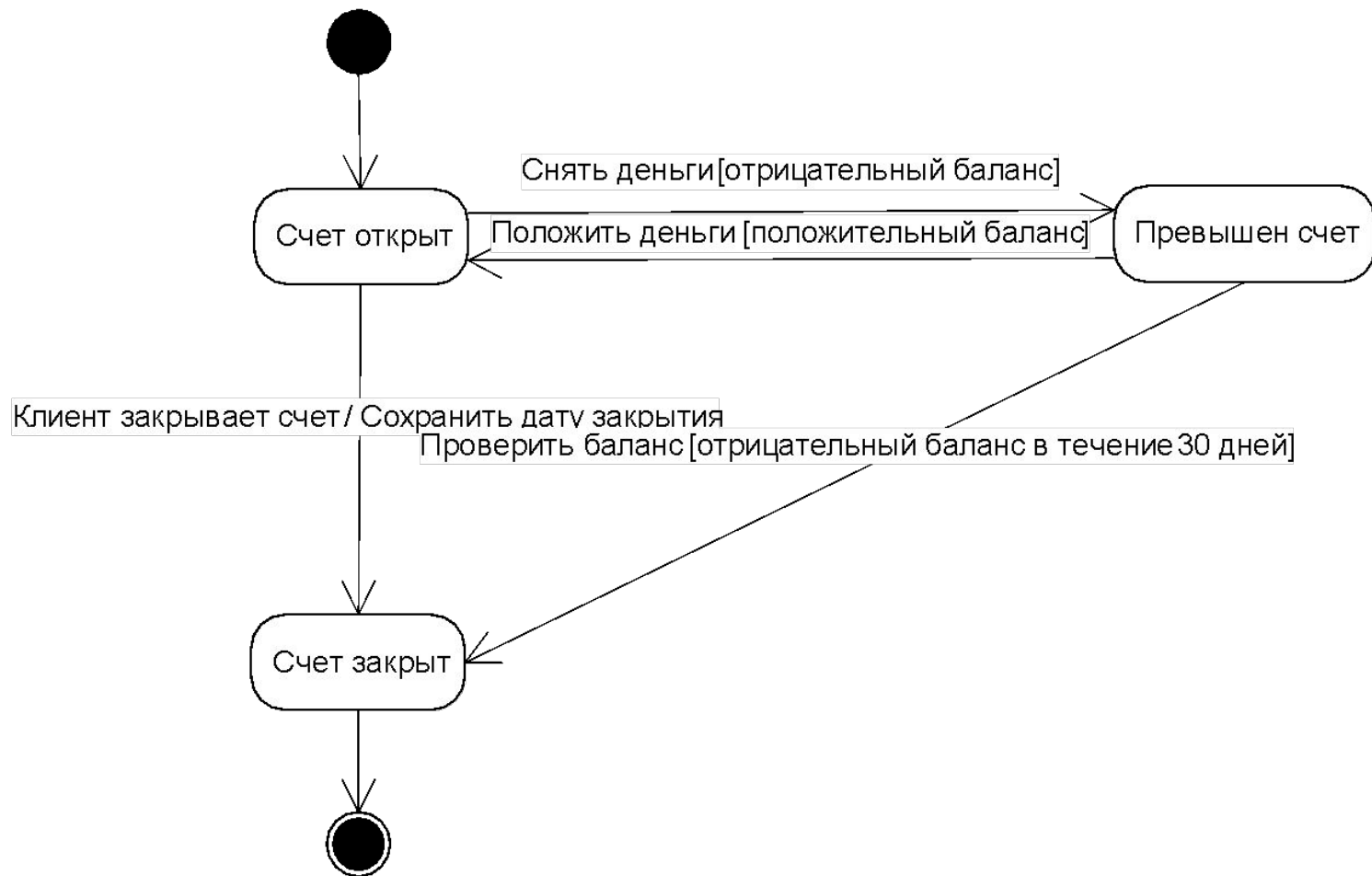


# Условие

- Сторожевое условие (guard condition) – это логическое условие, которое должно быть истинным, чтобы переход был осуществлен
- Сторожевое условие проверяется в момент возникновения события



# Пример диаграммы



# Действия в состояниях

- Состояние “Превышен счет”
  - Entry/Временно заморозить счет
  - Do/Послать уведомление клиенту
  - Exit/Разморозить счет
- Состояние “Счет закрыт”
  - Entry/Выдать кредитную карточку

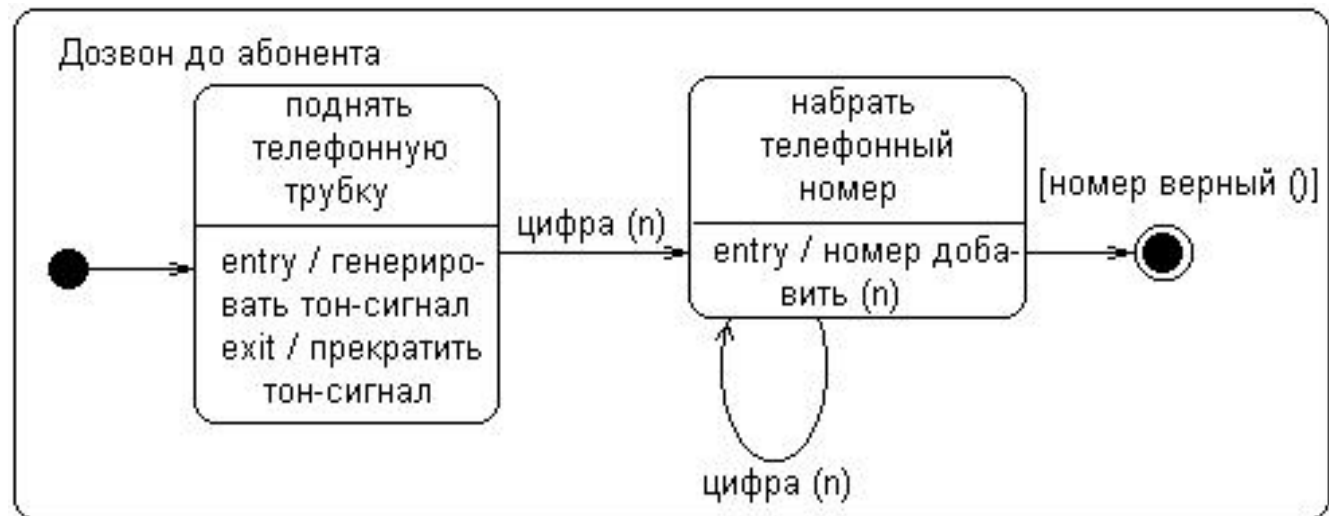
# Составное состояние

- Составное состояние Составное состояние (composite state) - такое сложное состояние, которое состоит из других вложенных в него состояний. Последние будут выступать по отношению к первому как подсостояния (substate). Хотя между ними имеет место отношение композиции



# Последовательные подсостояния

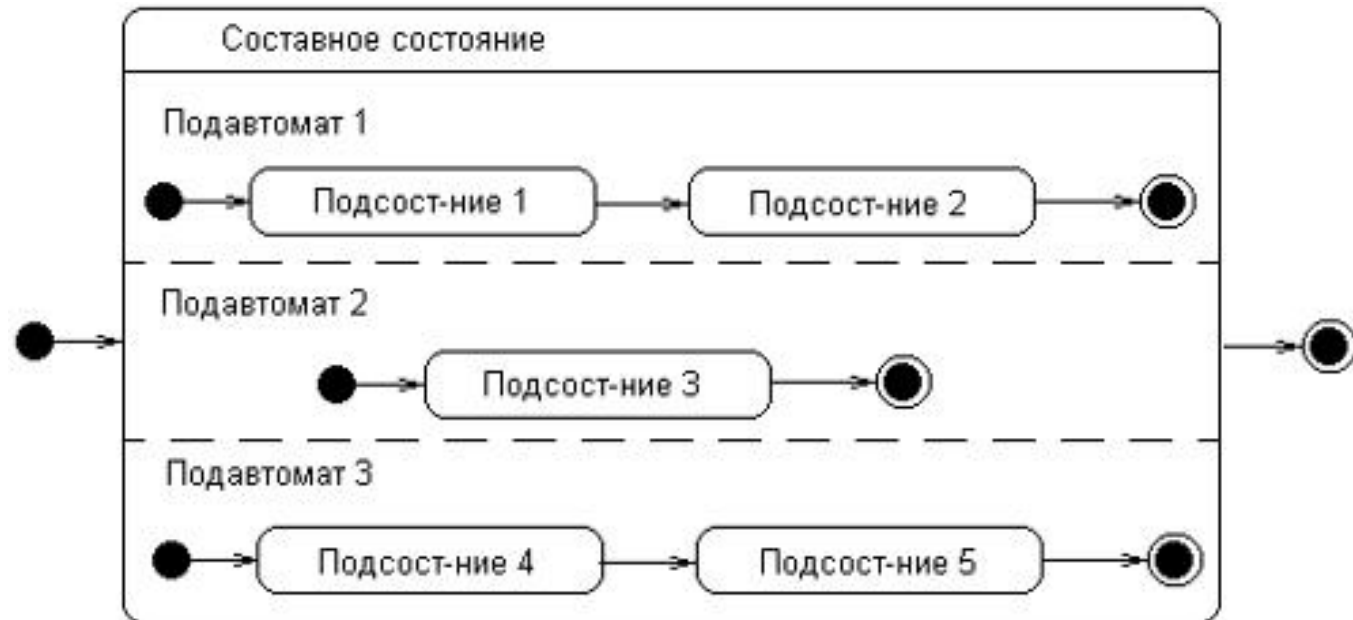
- Последовательные подсостояния (sequential substates) используются для моделирования такого поведения объекта, во время которого в каждый момент времени объект может находиться в одном и только одном подсостоянии.



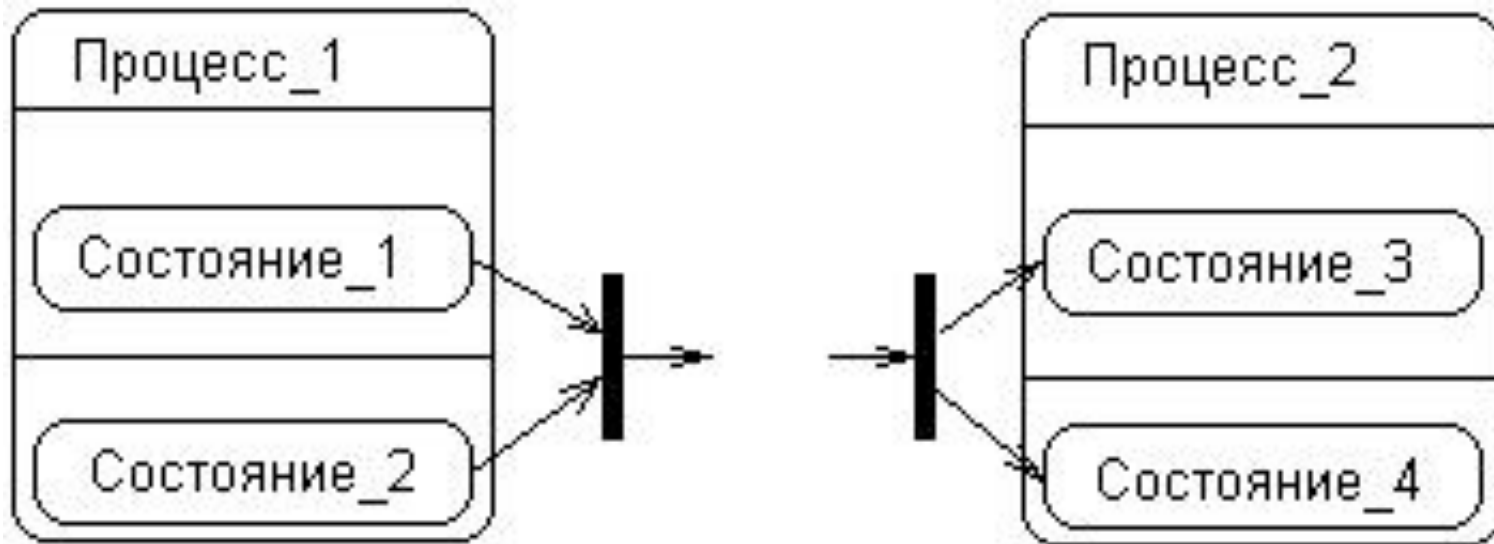
- составное состояние с двумя вложенными последовательными подсостояниями

# Параллельные подсостояния

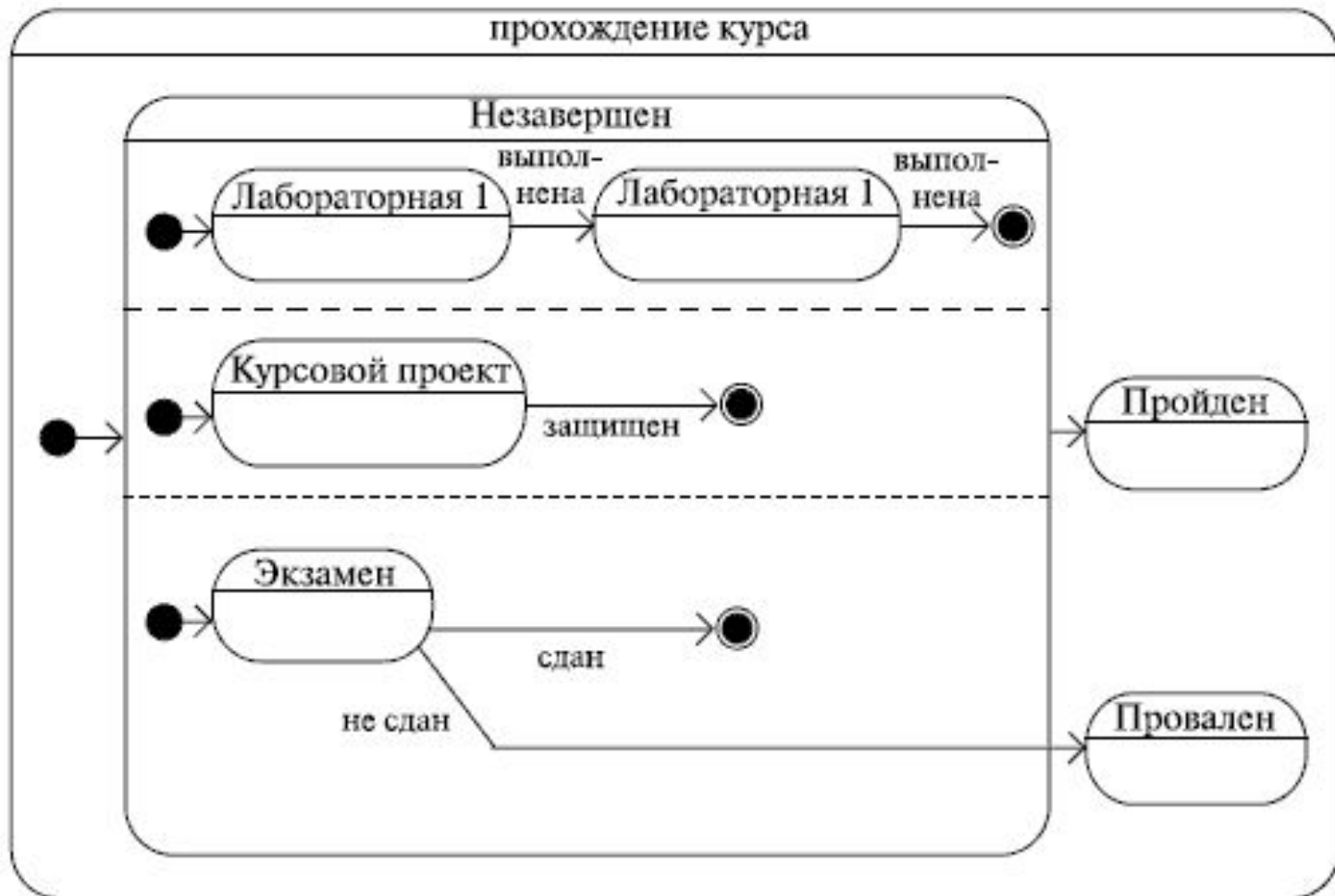
- Параллельные подсостояния (concurrent substates) позволяют специфицировать два и более подавтомата, которые могут выполняться параллельно внутри составного события.



# Переход между параллельными состояниями



# Составное состояние с параллельным выполнением





# Построение Диаграмм Состояния

- Определение содержания
- Выявление исходного, конечного и устойчивых состояний.
- Определение порядка перехода через устойчивые состояния
- Выявление событий, действий и условий, связанных с транзакциями
- Построение диаграммы