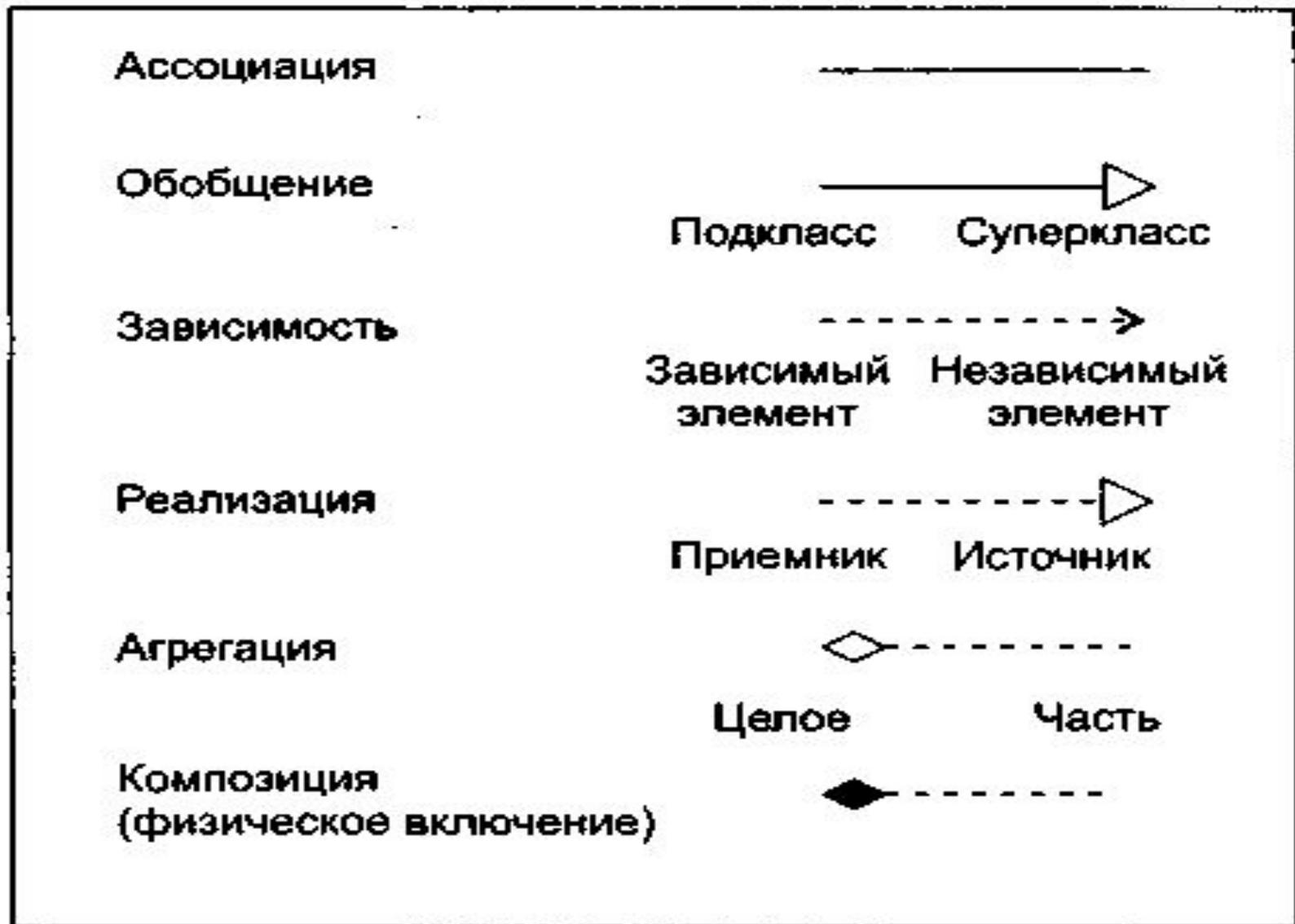


**ДИНАМИЧЕСКИЕ МОДЕЛИ  
ОБЪЕКТНО-  
ОРИЕНТИРОВАННЫХ  
ПРОГРАММНЫХ СИСТЕМ**

# Отношения в диаграммах классов



- **Ассоциации** отображают структурные отношения между экземплярами классов, то есть соединения между объектами.
- **Обобщение** — отношение между общим предметом (суперклассом) и специализированной разновидностью этого предмета (подклассом). Подкласс может иметь одного родителя (один суперкласс) или несколько родителей (несколько суперклассов). Во втором случае говорят о множественном наследовании.
- **Зависимость** является отношением использования между клиентом (зависимым элементом) и поставщиком (независимым элементом).
- **Реализация** — семантическое отношение между классами, в котором класс-приемник выполняет реализацию операций интерфейса класса-источника.
- **Агрегация** - процесс объединения элементов в одну систему
- **Композиция** (агрегирование) — методика создания нового класса в объектно-ориентированном программировании из уже существующих путём включения

# ДИНАМИЧЕСКИЕ МОДЕЛИ

- Динамические модели обеспечивают представление поведения систем. «Динамизм» этих моделей состоит в том, что в них отражается изменение состояний в процессе работы системы (в зависимости от времени).
- Средства языка UML для создания динамических моделей многочисленны и разнообразны. Эти средства ориентированы не только на собственно программные системы, но и на отображение требований заказчика к поведению таких систем.



# Моделирование поведения программной системы

- Для моделирования поведения системы используют:
- автоматы;
- взаимодействия.
- **Автомат** (State machine) описывает поведение в терминах последовательности состояний, через которые проходит объект в течение своей жизни.  
**Взаимодействие** (Interaction) описывает поведение в терминах обмена сообщениями между объектами.  
Таким образом, автомат задает поведение системы как цельной, единой сущности; моделирует жизненный цикл единого объекта.

# Моделирование поведения программной системы

Взаимодействия определяют поведение системы в виде коммуникаций между его частями (объектами), представляя систему как сообщество совместно работающих объектов. Именно поэтому взаимодействия считают основным аппаратом для фиксации полной динамики системы.

Автоматы отображают с помощью:

- диаграмм схем состояний;
- диаграмм деятельности.

Взаимодействия отображают с помощью:

- диаграмм сотрудничества (кооперации);
- диаграмм последовательности.

# Диаграммы схем состояний

**Диаграмма схем состояний** — одна из пяти диаграмм UML, моделирующих динамику систем. Диаграмма схем состояний отображает конечный автомат, выделяя поток управления, следующий от состояния к состоянию.

**Конечный автомат** — поведение, которое определяет последовательность состояний в ходе существования объекта. Эта последовательность рассматривается как ответ на события и включает реакции на эти события.

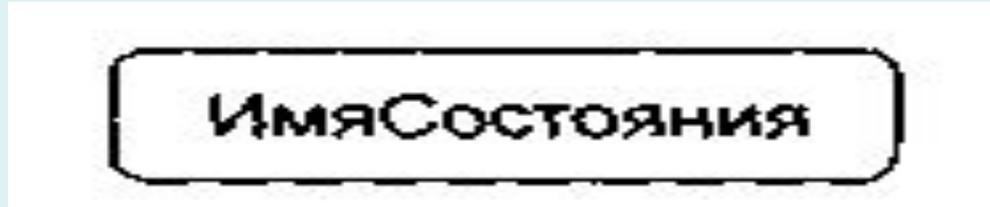
Диаграмма схем состояний показывает:

- 1) набор состояний системы;
- 2) события, которые вызывают переход из одного состояния в другое;
- 3) действия, которые происходят в результате изменения состояния.

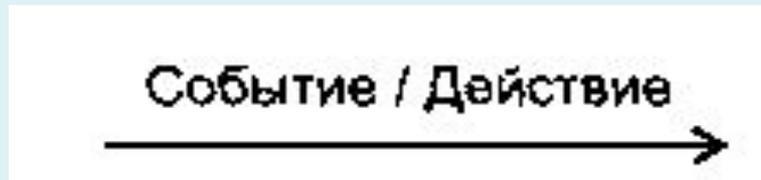
# Диаграммы схем состояний

В языке UML состоянием называют период в жизни объекта, на протяжении которого он удовлетворяет какому-то условию, выполняет определенную деятельность или ожидает некоторого события.

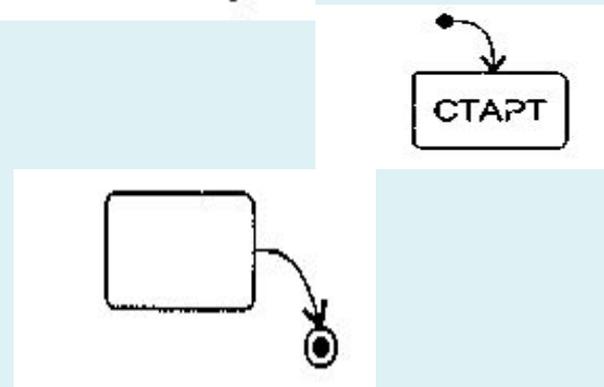
Как показано на рис., состояние изображается как закругленный прямоугольник, обычно включающий его имя и подсостояния (если они есть).



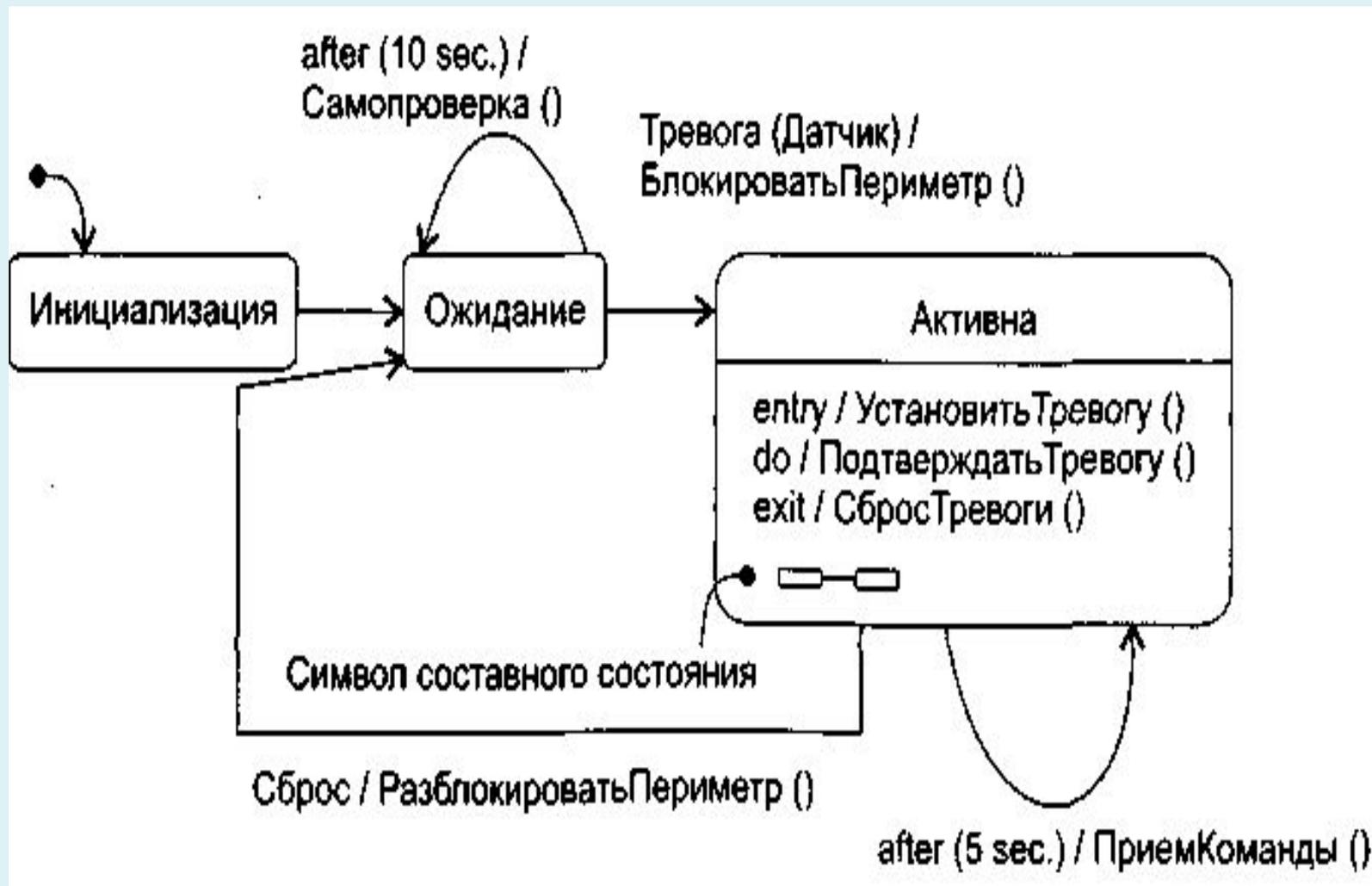
- 
- Переходы между состояниями отображаются помеченными стрелками
- обозначено: Событие — происшествие, вызывающее изменение состояния, Действие — набор операций, запускаемых событием.



- Переход в начальное состояние
- 
- Переход в конечное состояние
- 



# диаграмма схем состояний для системы охранной сигнализации

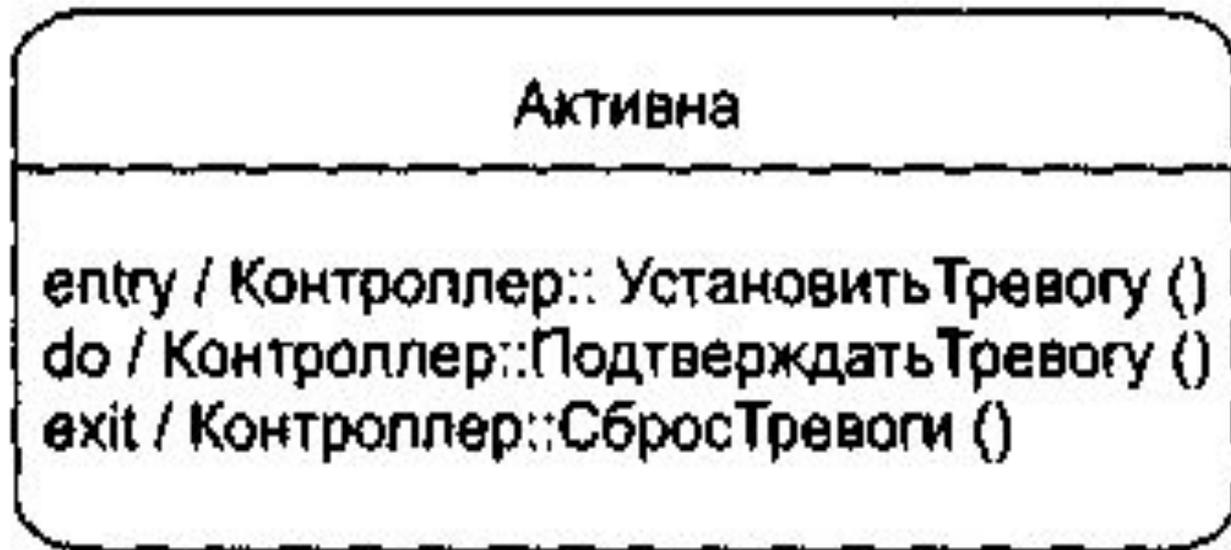


# диаграмма схем состояний для системы охранной сигнализации

- Из рисунка видно, что система начинает свою жизнь в состоянии Инициализация, затем переходит в состояние Ожидание. В этом состоянии через каждые 10 секунд (по событию after (10 sec.)) выполняется самопроверка системы (операция Самопроверка ()). При наступлении события Тревога (Датчик) реализуются действия, связанные с блокировкой периметра охраняемого объекта, — исполняется операция БлокироватьПериметр() и осуществляется переход в состояние Активна. В активном состоянии через каждые 5 секунд по событию after (5 sec.) запускается операция ПриемКоманды(). Если команда получена (наступило событие Сброс), система возвращается в состояние Ожидание. В процессе возврата разблокируется периметр охраняемого объекта (операция РазблокироватьПериметр()).

# ДЕЙСТВИЯ В СОСТОЯНИЯХ

- Для указания действий, выполняемых при входе в состояние и при выходе из состояния, используются метки `entry` и `exit` соответственно.
- Например, как показано на рис., при входе в состояние Активна выполняется операция `УстановитьТревогу()` из класса `Контроллер`, а при выходе из состояния — операция `СбросТревоги()`.



# Действия в состояниях

- Действие, которое должно выполняться, когда система находится в данном состоянии, указывается после метки `do`. Считается, что такое действие начинается при входе в состояние и заканчивается при выходе из него. Например, в состоянии Активна это действие `ПодтвердитьТревогу()`.

# Условные переходы

- Между состояниями возможны различные типы переходов. Обычно переход инициируется событием. Допускаются переходы и без событий. Наконец, разрешены условные или охраняемые переходы.

- Обозначен

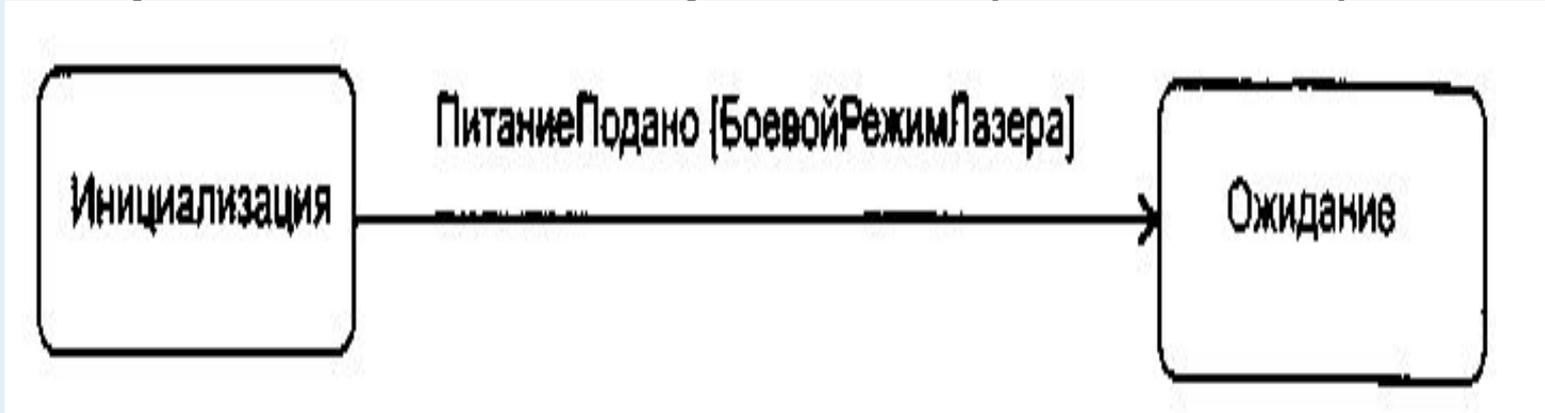
Событие [Условие] / Действие



- *Порядок выполнения условного перехода:*
- происходит событие;
- вычисляется условие УсловиеПерехода;
- при УсловиеПерехода=true запускается переход и активизируется действие, в противном случае переход не выполняется.

## Порядок выполнения условного перехода:

- происходит событие;
- вычисляется условие УсловиеПерехода;
- при УсловиеПерехода=true запускается переход и активизируется действие, в противном случае переход не выполняется.
- Пример условного перехода между состояниями Инициализация и Ожидание приведен на рис. Он происходит по событию ПитаниеПодано, но только в том случае, если достигнут боевой режим лазера.

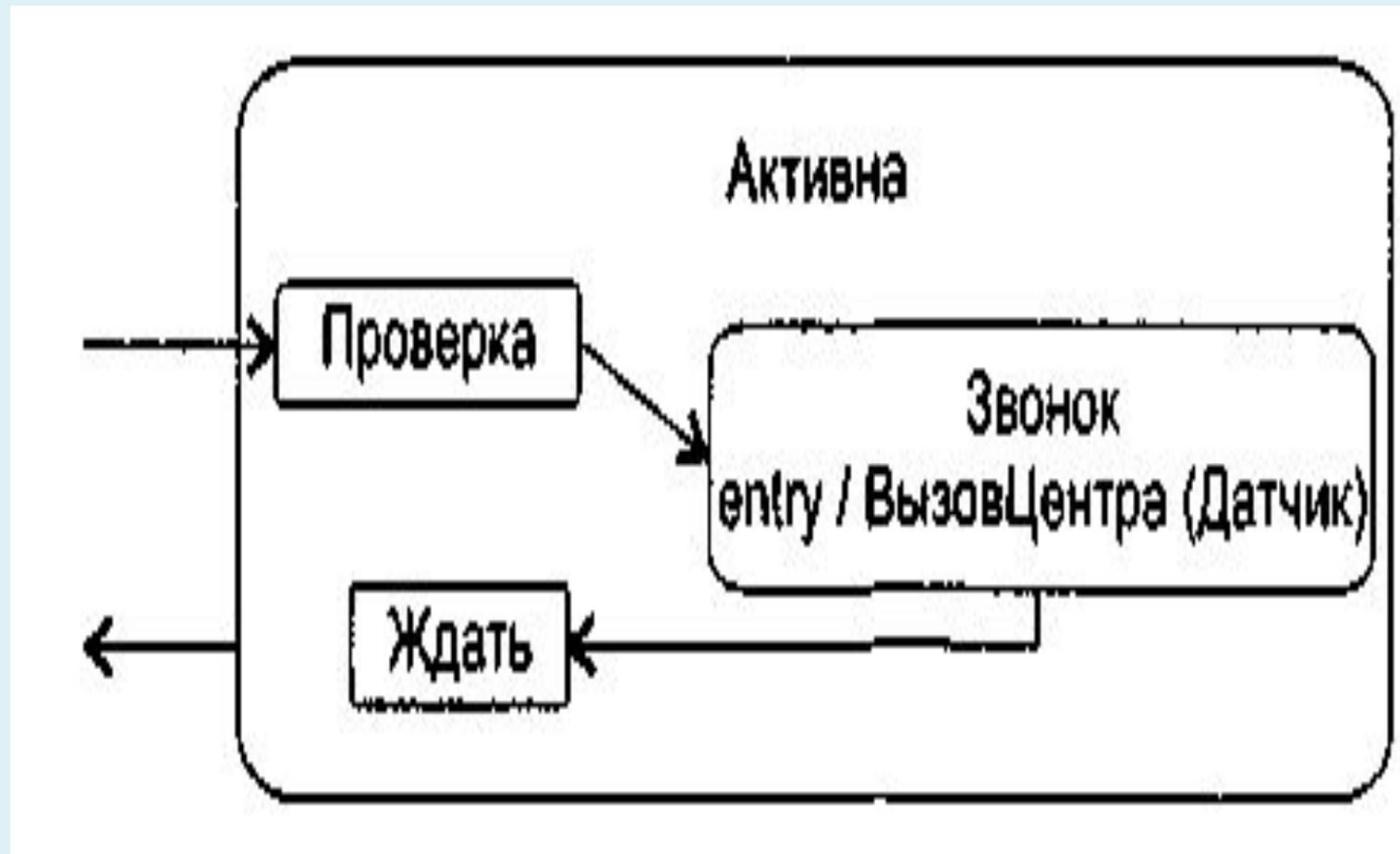


# Вложенные состояния

- Одной из наиболее важных характеристик конечных автоматов в UML является подсостояние. Подсостояние позволяет значительно упростить моделирование сложного поведения. Подсостояние — это состояние, вложенное в другое состояние. На рис. показано составное состояние, содержащее в себе два подсостояния.



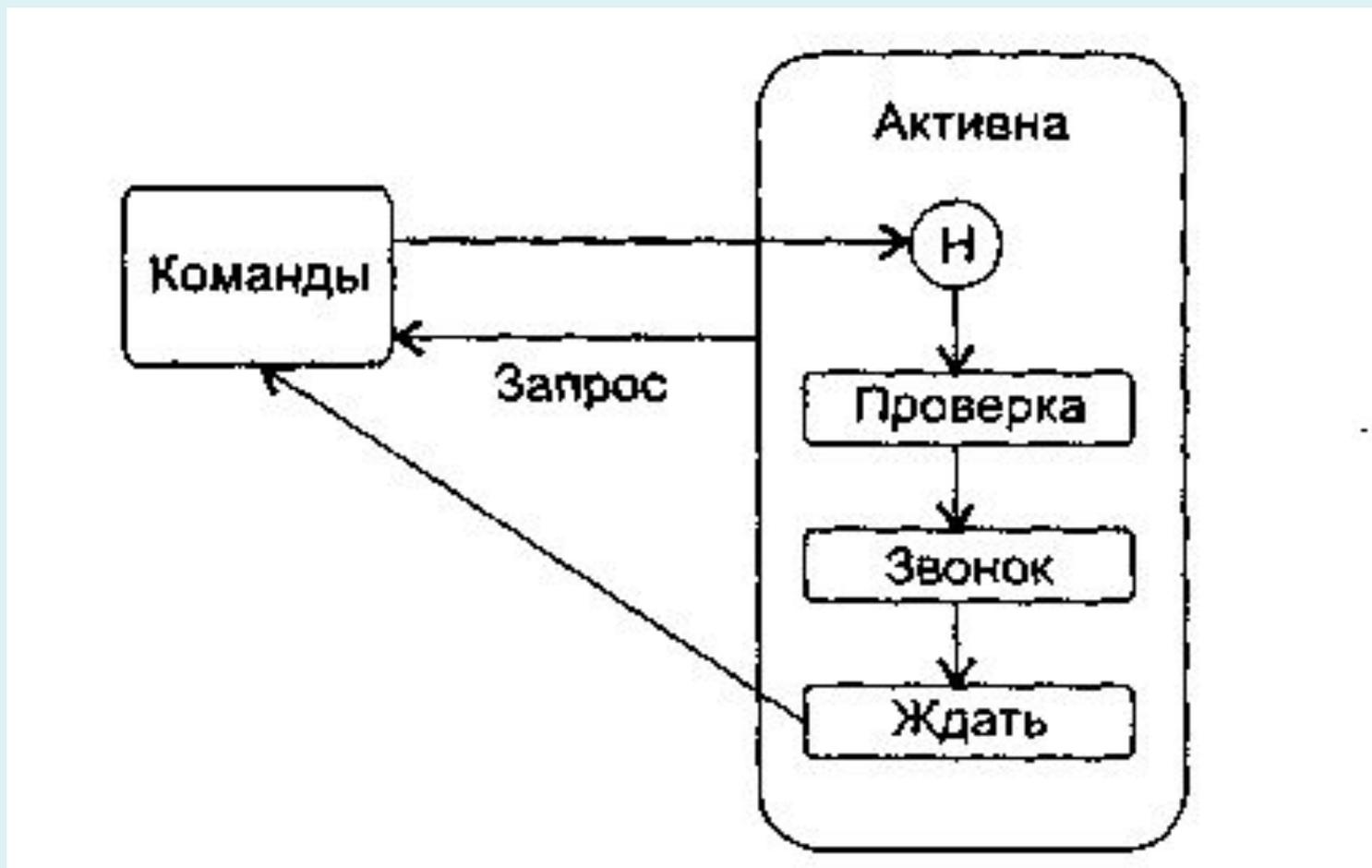
# внутренняя структура составного состояния Активна.



# внутренняя структура составного состояния Активна.

- Семантика вложенности такова: если система находится в состоянии Активна, то она должна быть точно в одном из подсостояний: Проверка, Звонок, Ждать. В свою очередь, в подсостояние могут вкладываться другие подсостояния. Степень вложенности подсостояний не ограничивается. Данная семантика соответствует случаю последовательных подсостояний.
- Возможно наличие параллельных подсостояний — они выполняются параллельно внутри составного состояния. Графически изображения параллельных подсостояний отделяются друг от друга пунктирными линиями.

Иногда при возврате в составное состояние возникает необходимость попасть в то его подсостояние, которое в прошлый раз было последним. Такое подсостояние называют историческим. Информация об историческом состоянии запоминается. Как показано на рис, подобная семантика переходов отображается значком истории — буквой Н внутри кружка.

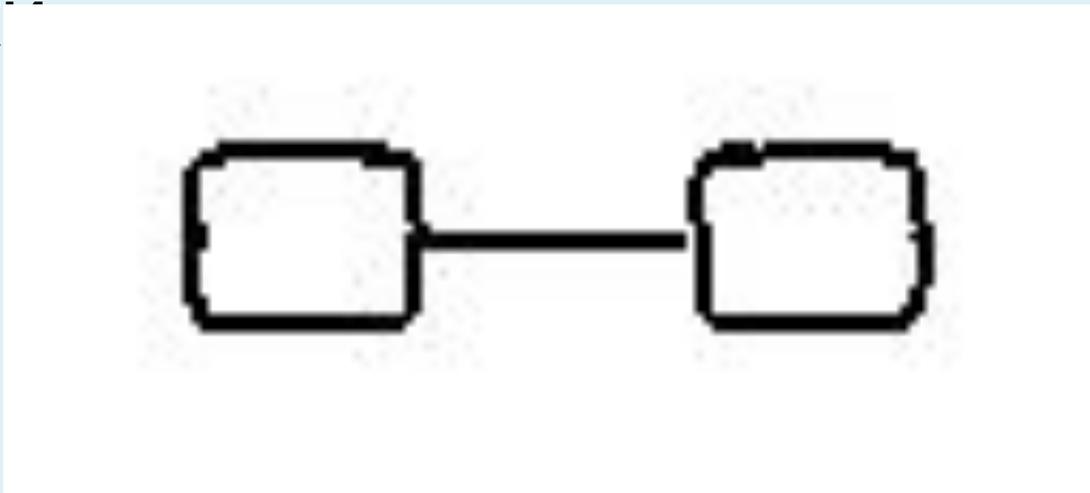


# внутренняя структура составного состояния Активна.

- При первом посещении состояния Активна автомат не имеет истории, поэтому происходит простой переход в подсостояние Проверка. Предположим, что в подсостоянии Звонок произошло событие Запрос. Средства управления заставляют автомат покинуть подсостояние Звонок (и состояние Активна) и вернуться в состояние Команды. Когда работа в состоянии Команды завершается, выполняется возврат в историческое подсостояние состояния Активна. Поскольку теперь автомат запомнил историю, он переходит прямо в подсостояние Звонок (минуя подсостояние Проверка).

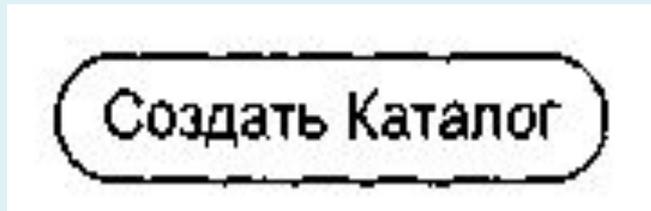
# внутренняя структура составного состояния Активна

- для обозначения составного состояния, имеющего внутри себя скрытые (не показанные на диаграмме) подсостояния, используется символ «Очки»

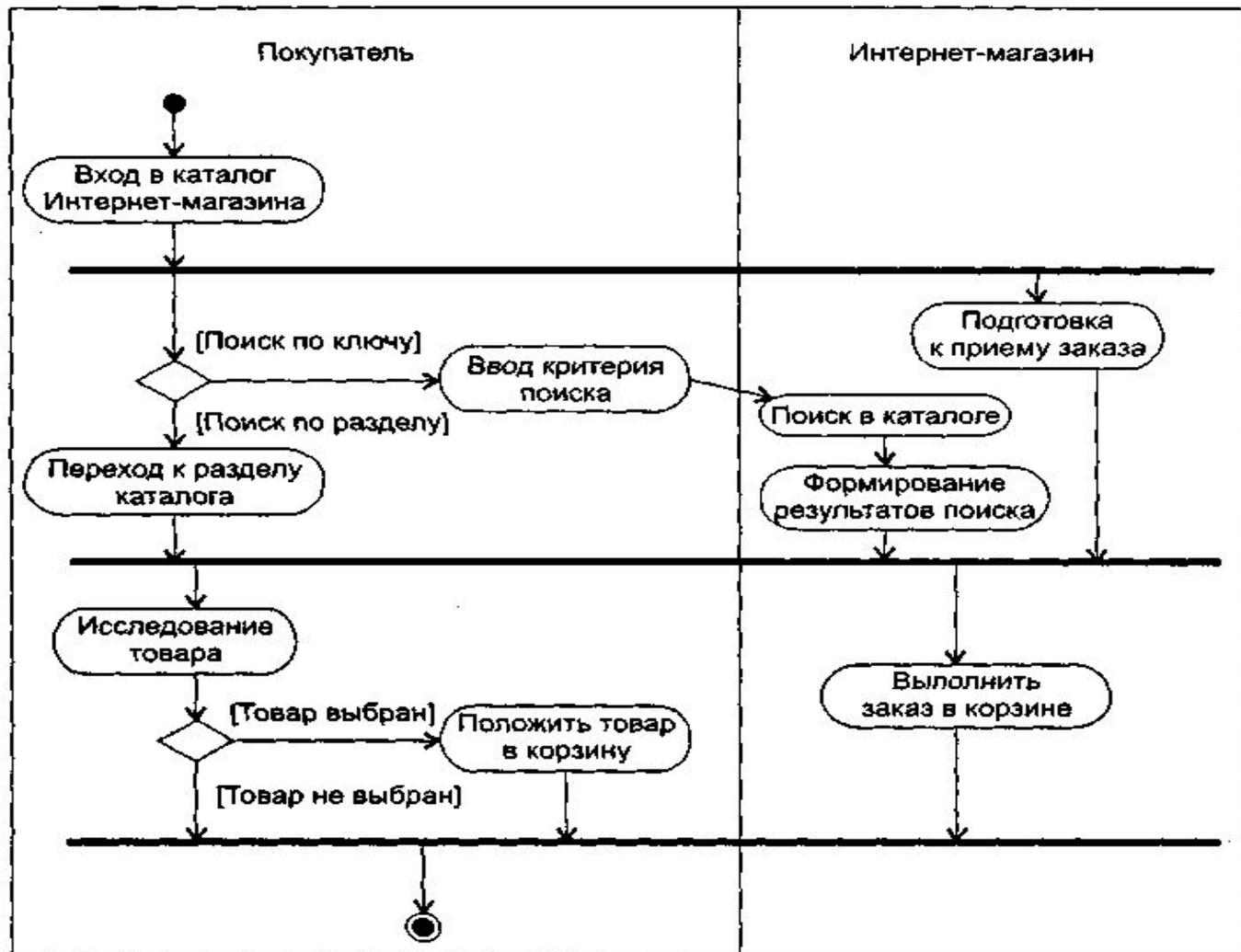


# Диаграммы деятельности

- Диаграмма деятельности представляет особую форму конечного автомата, в которой показываются процесс вычислений и потоки работ.
- Основной вершиной в диаграмме деятельности является состояние действия, которое изображается как прямоугольник с закругленными боковыми сторонами.



# Диаграмма деятельности покупателя в Интернет-магазине

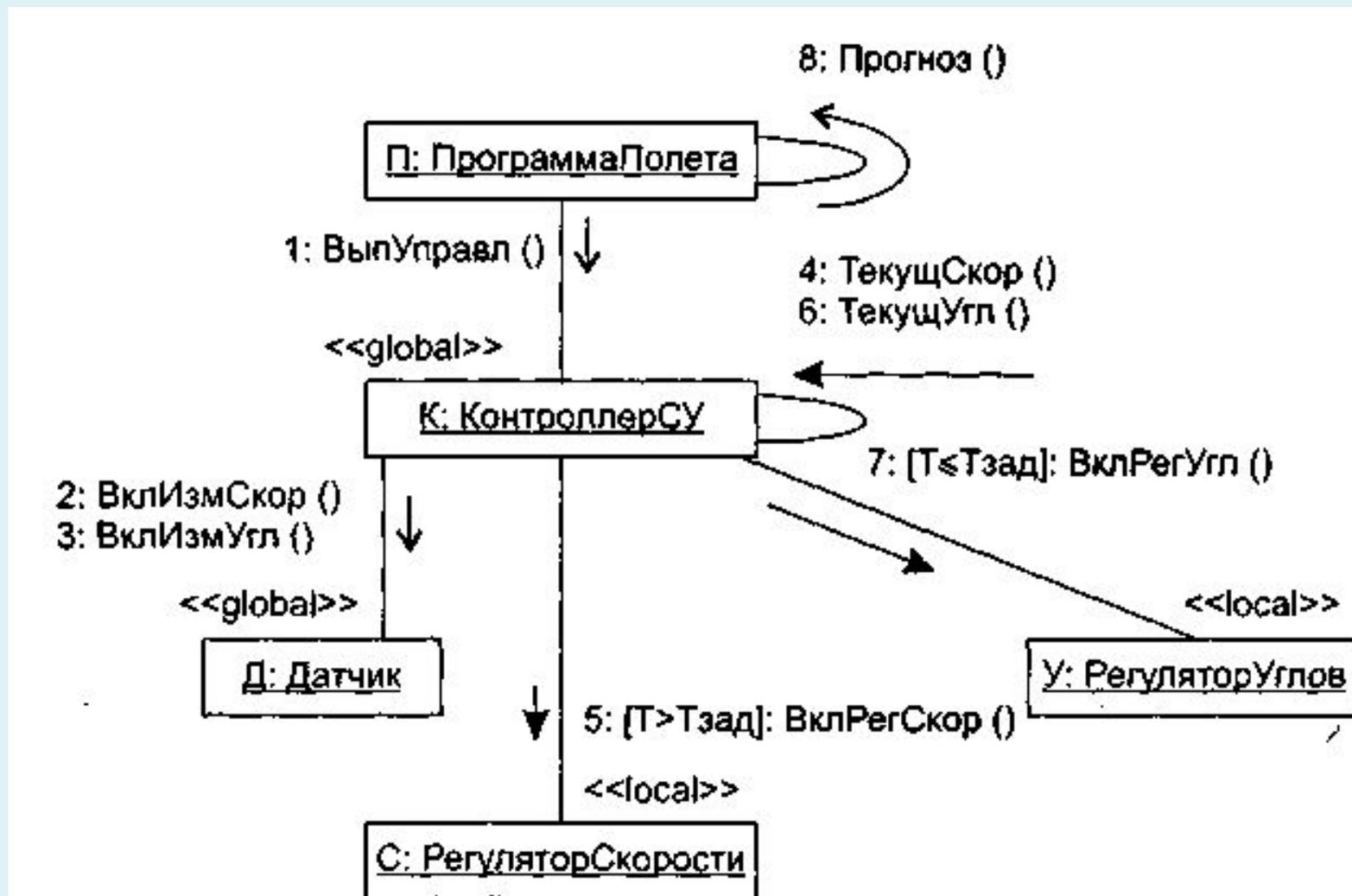


# Диаграммы взаимодействия

- Диаграммы взаимодействия предназначены для моделирования динамических аспектов системы. Диаграмма взаимодействия показывает взаимодействие, включающее набор объектов и их отношений, а также пересылаемые между объектами сообщения. Существуют две разновидности диаграммы взаимодействия — диаграмма последовательности и диаграмма сотрудничества. Диаграмма последовательности — это диаграмма взаимодействия, которая выделяет упорядочение сообщений по времени. Диаграмма сотрудничества — это диаграмма взаимодействия, которая выделяет структурную организацию объектов, посылающих и принимающих сообщения. Элементами диаграмм взаимодействия являются участники взаимодействия — объекты, связи, сообщения.

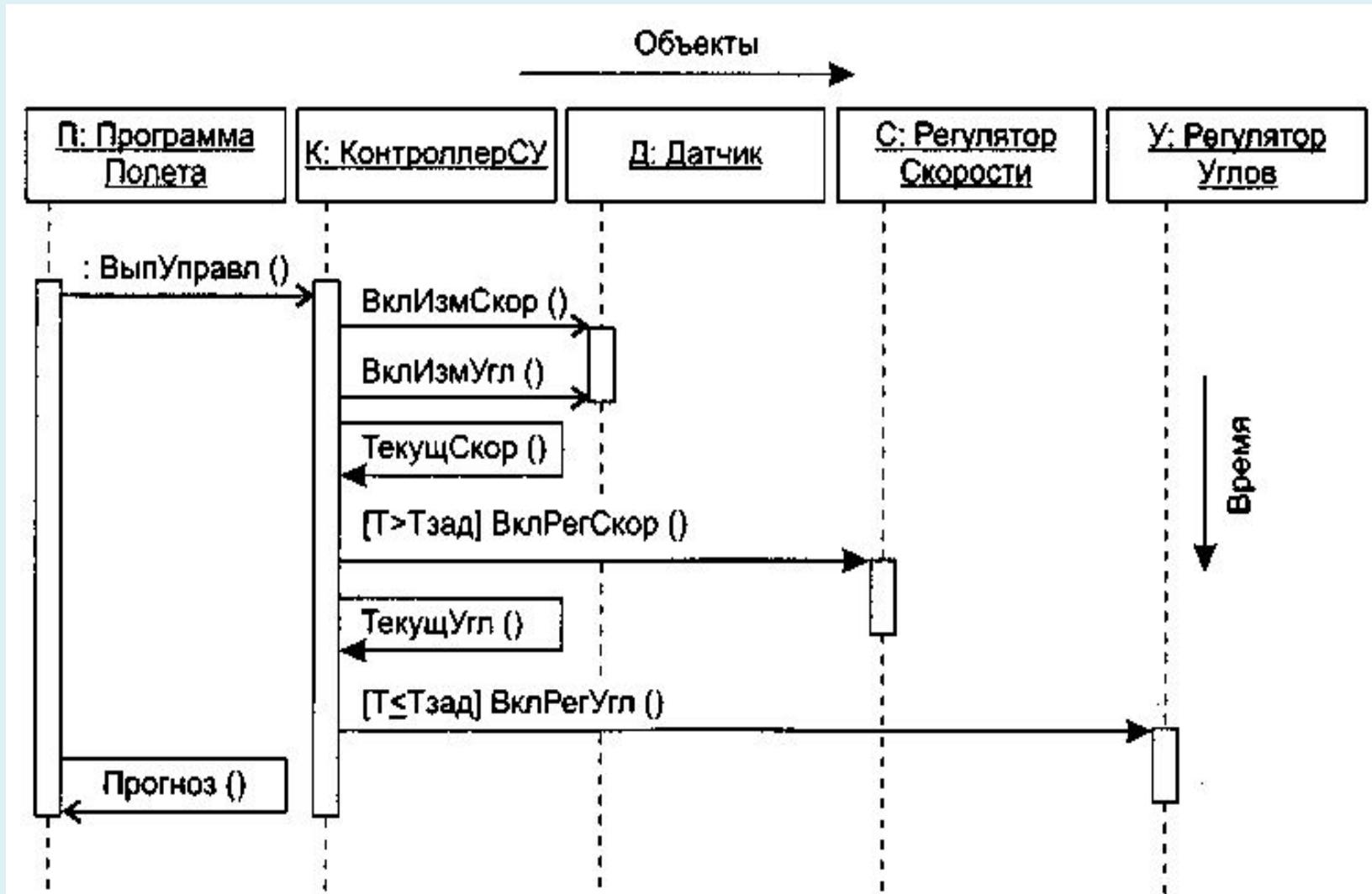
- Диаграммы сотрудничества отображают взаимодействие объектов в процессе функционирования системы. Такие диаграммы моделируют сценарии поведения системы. В русской литературе диаграммы сотрудничества часто называют диаграммами кооперации.

# Диаграмма сотрудничества



- Диаграмма последовательности — вторая разновидность диаграмм взаимодействия. Отражая сценарий поведения в системе, эта диаграмма обеспечивает более наглядное представление порядка передачи сообщений. Правда, она не позволяет показать такие детали, которые видны на диаграмме сотрудничества (структурные характеристики объектов и связей).
- Графически диаграмма последовательности — разновидность таблицы, которая показывает объекты, размещенные вдоль оси  $X$ , и сообщения, упорядоченные по времени вдоль оси  $Y$ .

# диаграмма последовательности



# Вопросы

- Поясните два подхода к моделированию поведения системы. Объясните достоинства и недостатки каждого из этих подходов.
- Охарактеризуйте вершины и дуги диаграммы схем состояний. В чем состоит назначение этой диаграммы?
- Как отображаются действия в состояниях диаграммы схем состояний?
- Как показываются условные переходы между состояниями?
- Как задаются вложенные состояния в диаграммах схем состояний?
- Поясните понятие исторического подсостояния.
- Охарактеризуйте средства и возможности диаграммы деятельности.
- Когда не следует применять диаграмму деятельности?
- Какие средства диаграммы деятельности позволяют отобразить параллельные действия?