

* Диаграмма состояний (statechart diagram) в UML

Выполнила Башкина Полина
студентка группы 150301

* Диаграмма состояний является графом специального вида, который представляет некоторый автомат. Вершинами графа являются возможные состояния автомата, изображаемые соответствующими графическими символами, а дуги обозначают его переходы из состояния в состояние.

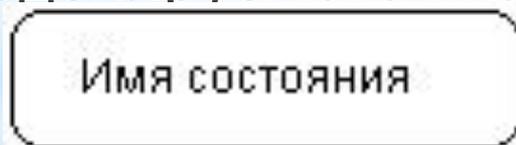
* Автоматы в UML

- * В метамодели UML *автомат* является пакетом, в котором определено множество понятий, необходимых для представления поведения моделируемой сущности в виде дискретного пространства с конечным числом состояний и переходов.
- * Поведение автомата моделируется как последовательное перемещение по графу от вершины к вершине с учетом ориентации связывающих их дуг.

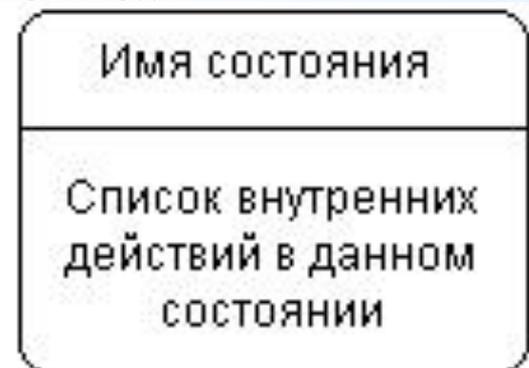
- * Для автомата должны выполняться следующие обязательные условия:
- * состояние, в которое может перейти объект, определяется только его текущим состоянием и не зависит от предыстории;
- * в каждый момент времени автомат может находиться только в одном из своих состояний. При этом, автомат может находиться в отдельном состоянии как угодно долго, если не происходит никаких событий;
- * время нахождения автомата в том или ином состоянии, а также время достижения того или иного состояния никак не специфицируются;
- * количество состояний автомата должно быть конечным и все они должны быть специфицированы явным образом. Отдельные псевдосостояния могут не иметь спецификаций (начальное и конечное состояния). В этом случае их назначение и семантика полностью определяются из контекста модели и рассматриваемой диаграммы состояний;
- * граф автомата не должен содержать изолированных состояний и переходов. Для каждого состояния, кроме начального, должно быть определено предшествующее состояние, а каждый переход должен соединять два состояния автомата;
- * автомат не должен содержать конфликтующих переходов, когда объект одновременно может перейти в два и более последующих состояния (кроме случая параллельных подавтоматов). В языке UML исключение конфликтов возможно на основе введения сторожевых условий

* Понятие состояния объекта

* В языке UML под состоянием понимается абстрактный метакласс, используемый для моделирования отдельной ситуации, в течение которой выполняются некоторые условия. Состояние может быть задано в виде набора конкретных значений атрибутов класса или объекта. Изменение отдельных значений атрибутов будет отражать изменение состояния моделируемого класса или объекта.



(a)



(б)

- * **Имя состояния** представляет собой строку текста, которая раскрывает его содержательный смысл. Имя всегда записывается с заглавной буквы
- * Имя у состояния может отсутствовать и этом случае состояние является анонимным. Если на диаграмме анонимных состояний несколько, то они должны различаться между собой.
- * **Список внутренних действий** содержит перечень действий или деятельностей, которые выполняются во время нахождения моделируемого элемента в данном состоянии. Каждое из действий записывается в виде отдельной строки и имеет следующий формат:
 - * <метка-действия '/' выражение-действия>
 - * Метка действия указывает на обстоятельства или условия, при которых будет выполняться деятельность, определенная выражением действия. При этом, выражение действия может использовать любые атрибуты и связи, которые принадлежат области имен или контексту моделируемого объекта. Если список выражений действия пустой, то разделитель в виде наклонной черты '/' может не указываться.

- * Перечень меток действия имеет фиксированные значения, которые не могут быть использованы в качестве имен событий. Эти значения следующие:
- * `entry` - эта метка указывает на действие, специфицированное следующим за ней выражением действия, которое выполняется в момент входа в данное состояние (входное действие);
- * `exit` - эта метка указывает на действие, специфицированное следующим за ней выражением действия, которое выполняется в момент выхода из данного состояния (выходное действие);
- * `do` - эта метка специфицирует выполняющуюся деятельность («do activity»), которая выполняется в течение всего времени, пока объект находится в данном состоянии, или до тех пор, пока не закончится вычисление, специфицированное следующим за ней выражением действия. В этом случае при завершении события формируется соответствующий результат;
- * `include` - эта метка используется для обращения к подавтомату, при этом следующее за ней выражение действия содержит имя этого подавтомата.

* Начальное состояние

* *Начальное состояние* представляет собой частный случай состояния, которое не содержит никаких внутренних действий (псевдосостояние). В этом состоянии находится объект по умолчанию в начальный момент времени. Оно служит для указания на диаграмме графической области, от которой начинается процесс изменения состояний

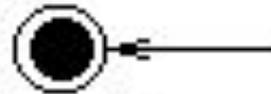
* Конечное состояние

* *Конечное состояние* представляет собой частный случай состояния, которое также не содержит никаких внутренних действий (псевдосостояние). В этом состоянии будет находиться объект по умолчанию после завершения работы автомата в конечный момент времени. Оно служит для указания на диаграмме графической области, в которой завершается процесс изменения состояний или жизненный цикл данного объекта.



(а)

начальное состояние



(б)

конечное состояние

* Переход

- * *Простой переход* (simple transition) представляет собой отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния объекта другим.
- * На диаграмме состояний переход изображается сплошной линией со стрелкой, которая направлена в целевое состояние.

- * Каждый переход может быть помечен строкой текста, которая имеет следующий общий формат:
- * <сигнатура события> '['<сторожевое условие>']'
<выражение действия>.
- * При этом *сигнатура события* описывает некоторое событие с необходимыми аргументами:
- * <имя события> '('<список параметров, разделенных запятыми>')'.
- * *Событие* (event) является самостоятельным элементом языка UML. Формально, событие представляет собой спецификацию некоторого факта, имеющего место в пространстве и во времени.
- * В языке UML события играют роль стимулов, которые инициируют переходы из одних состояний в другие.

* Сторожевое условие

* Сторожевое условие (guard condition), если оно есть, всегда записывается в прямых скобках после события-триггера и представляет собой некоторое булевское выражение

* **Диаграмма состояний для моделирования почтовой программы-клиента**

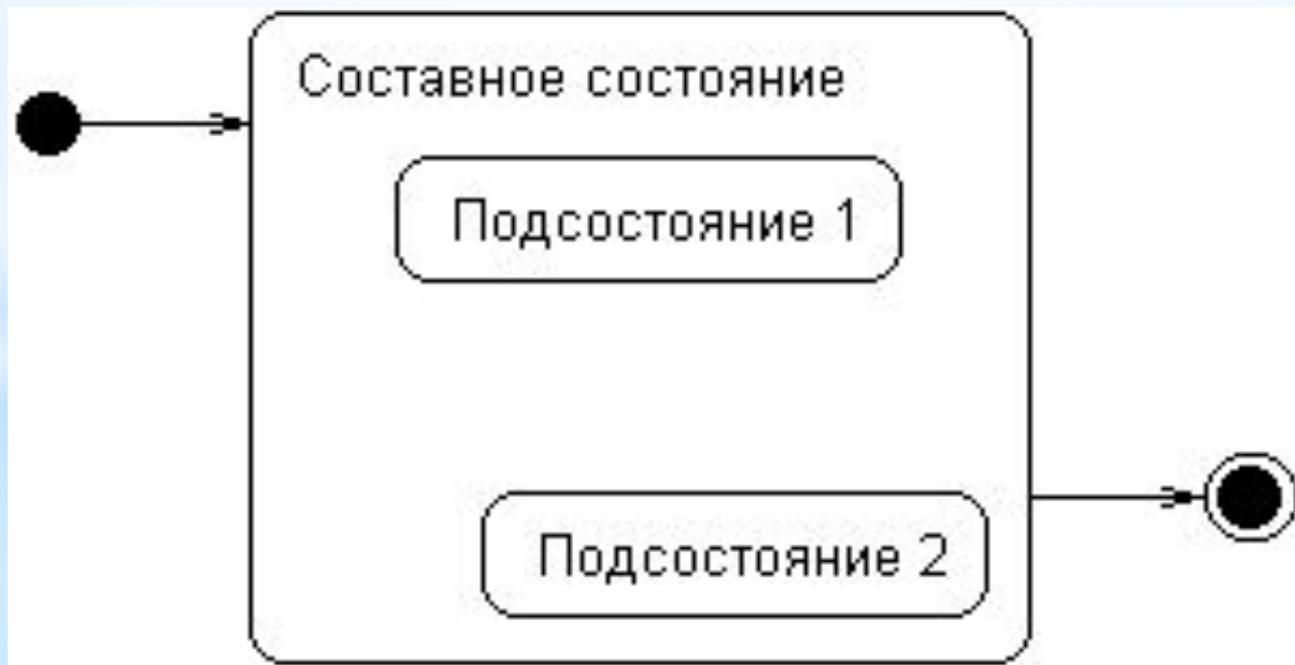


* Выражение действия

- * Выражение действия (action expression) выполняется в том и только в том случае, когда переход срабатывает. Представляет собой атомарную операцию (достаточно простое вычисление), выполняемую сразу после срабатывания соответствующего перехода до начала каких бы то ни было действий в целевом состоянии.

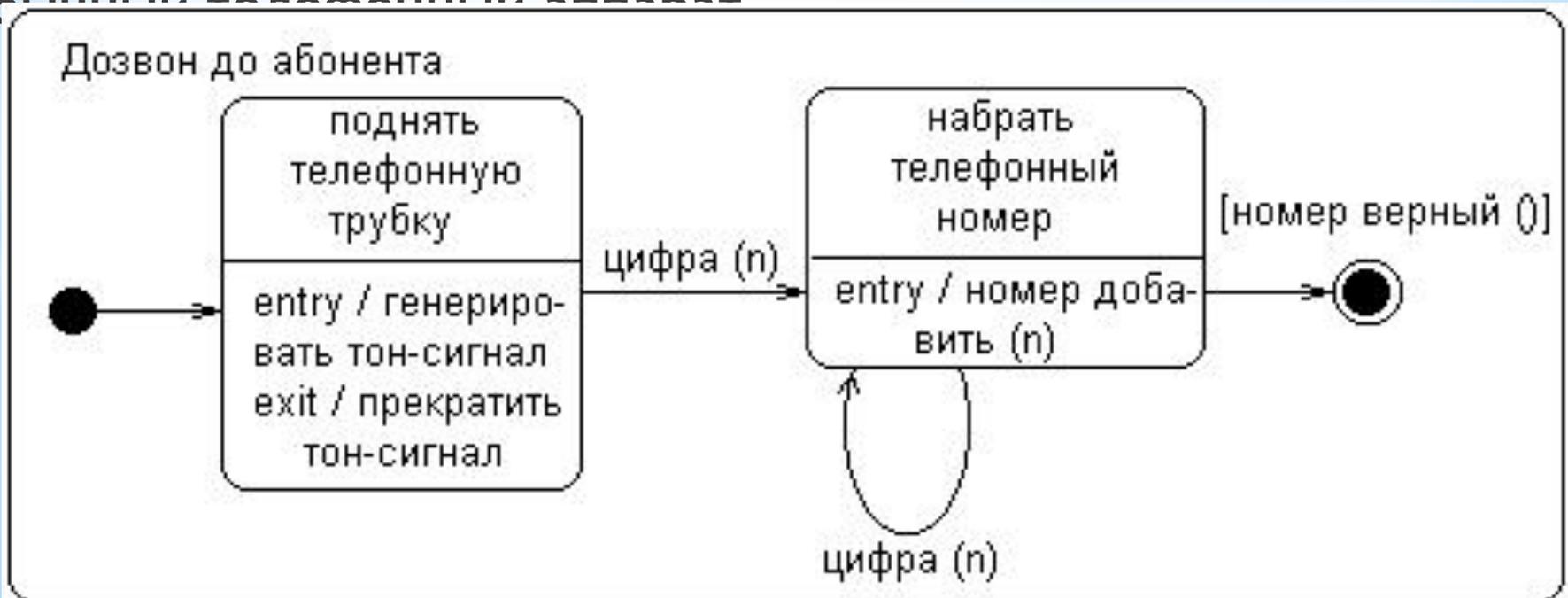
* Составное состояние и ПОДСОСТОЯНИЕ

- * 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).



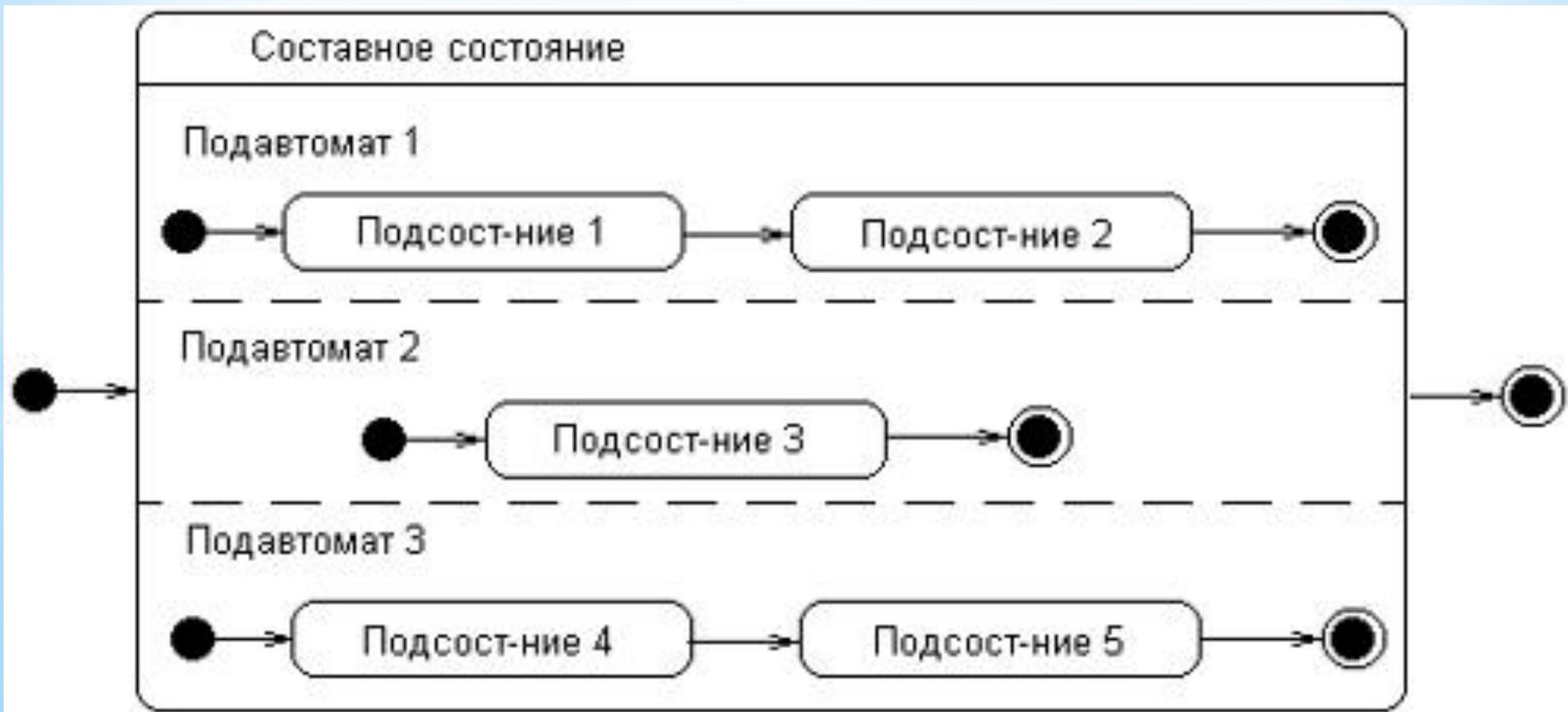
* Последовательные ПОДСОСТОЯНИЯ

- * 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.
- * Для примера рассмотрим в качестве моделируемого объекта обычный телефонный аппарат



* Параллельные подсостояния

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

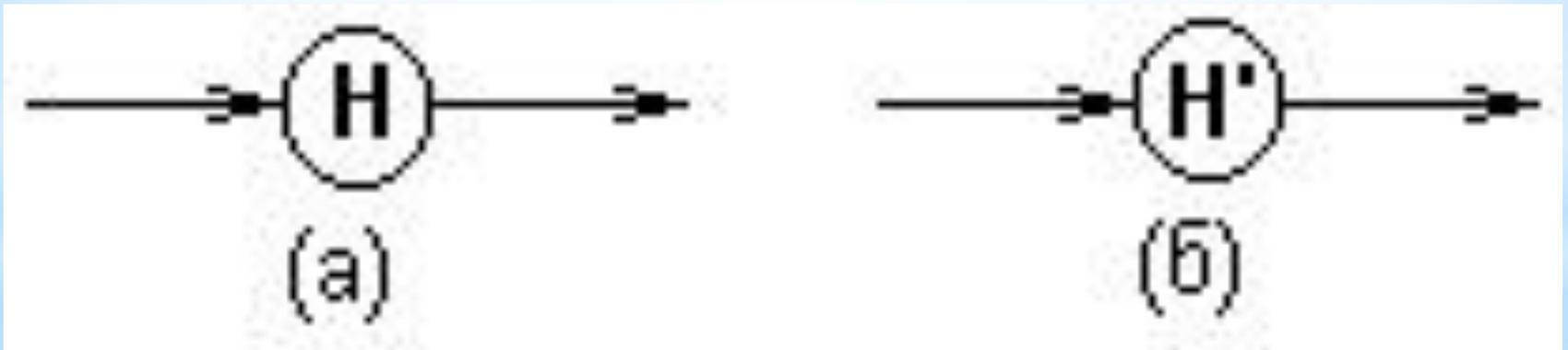


* Составное состояние со скрытой внутренней структурой и специальной пиктограммой



* Историческое состояние

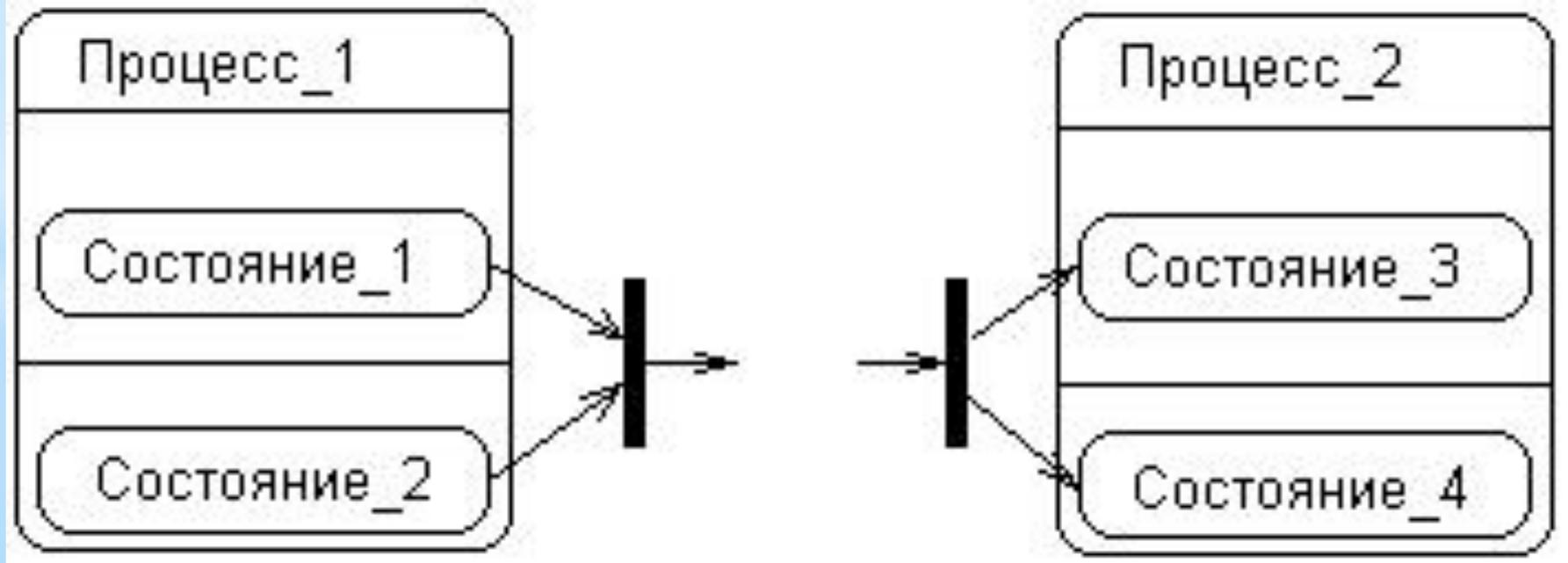
- * Историческое состояние (history state) применяется в контексте составного состояния. Оно используется для запоминания того из последовательных подсостояний, которое было текущим в момент выхода из составного состояния. При этом существует две разновидности исторического состояния: недавнее и давнее



* Сложные переходы

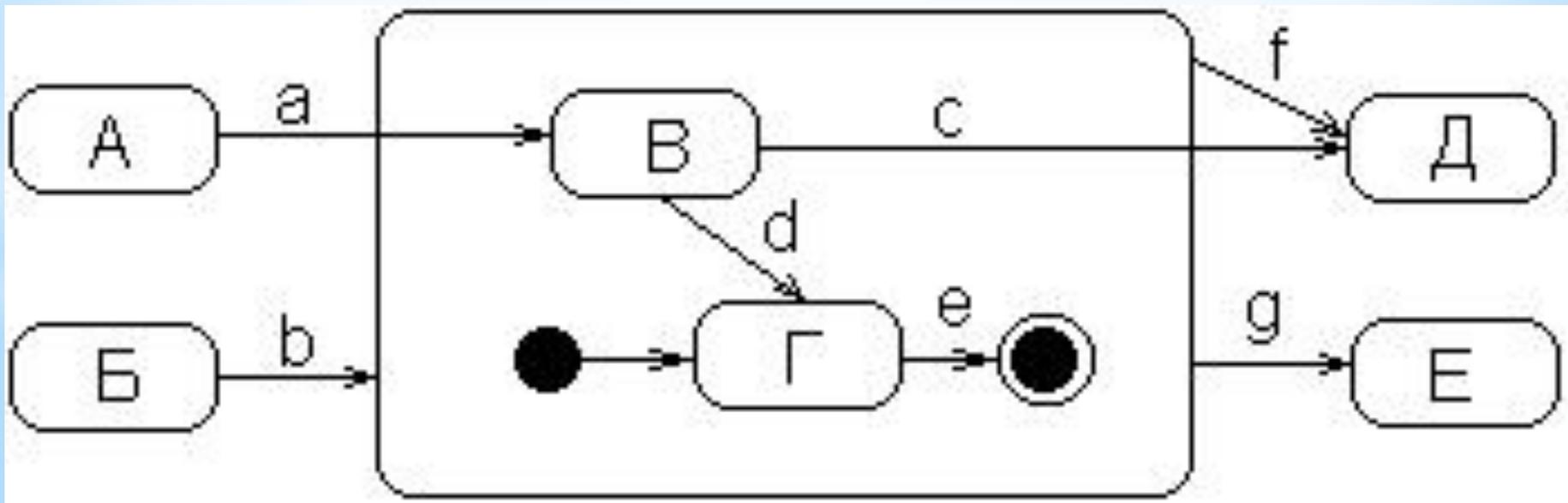
* Переходы между параллельными состояниями

- * В отдельных случаях переход может иметь несколько состояний-источников и несколько целевых состояний. Такой переход получил специальное название - параллельный переход.



* Переходы между составными состояниями

* Переход, стрелка которого соединена с границей некоторого составного состояния, обозначает переход в составное состояние (переход b)

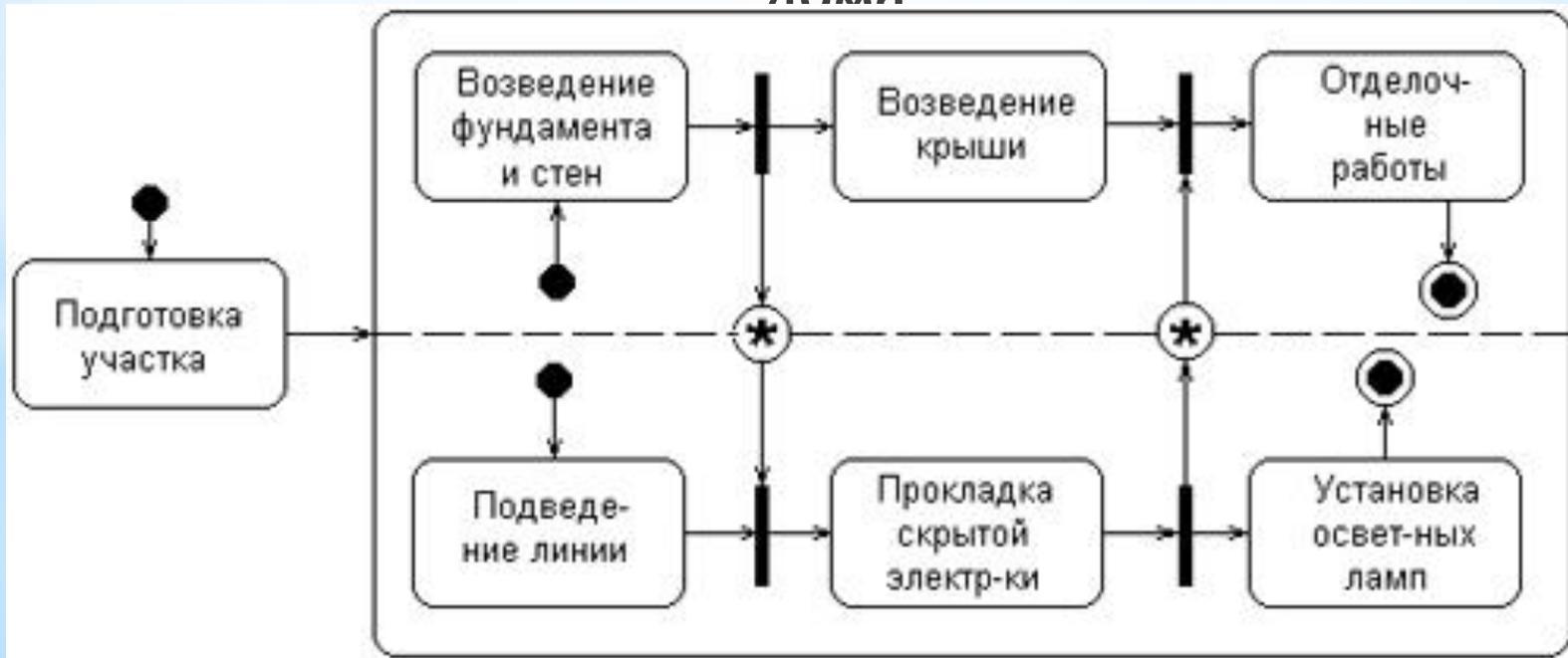




Синхронизирующие состояния

* Синхронизирующее состояние (synch state) обозначается небольшой окружностью, внутри которой помещен символ звездочки "*". Оно используется совместно с переходом-соединением или переходом-ветвлением для того, чтобы явно указать события в других подавтоматах, оказывающие непосредственное влияние на поведение данного подавтомата.

* Диаграмма состояний для примера со строительством дома



* Диаграмма состояний процесса
функционирования
телефонного аппарата

