

Методы и средства проектирования информационных систем и технологий

Лекция 7

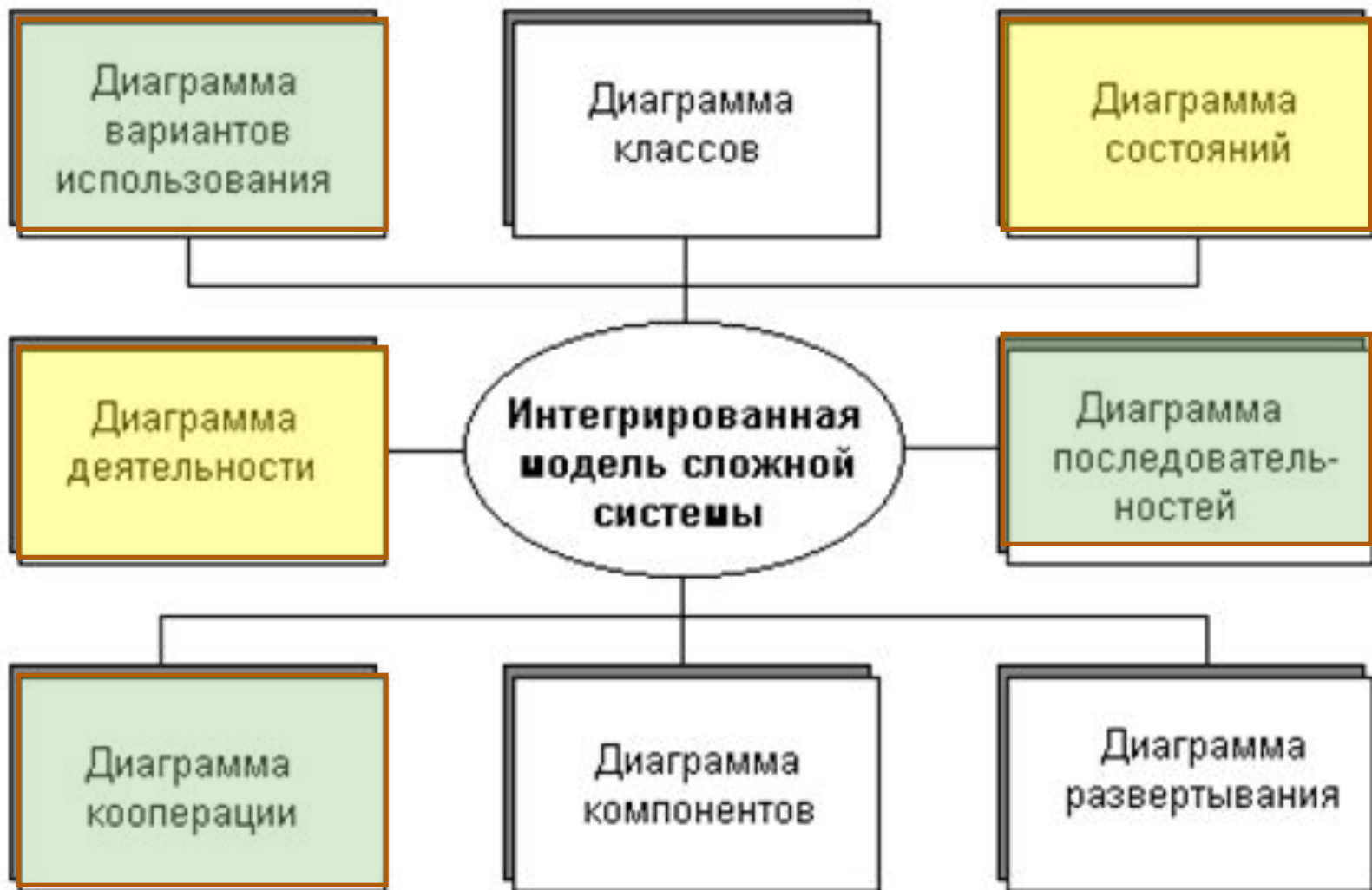


Диаграмма деятельности

Создание Информационной Системы – сложный процесс, который можно представить как поэтапный спуск от общей концепции будущей ИС, через понимание ее логической структуры к наиболее детальным моделям, описывающим физическую реализацию.

Диаграмма вариантов использования (use case) отвечает на вопрос - ЧТО должна делать Система.

Диаграмма деятельности отвечает на вопрос КАК в системе выполняются требуемые функции.

Деятельность может содержать входящие и/или исходящие **дуги деятельности**, показывающие потоки управления и потоки данных. Если поток соединяет две деятельности, он является потоком управления. Если поток заканчивается объектом, он является потоком данных.

Диаграмма деятельности

Деятельность выполняется, только тогда, когда готовы все его «входы», после выполнения, деятельность передает управление и (или) данные на свои «выходы».

Саму диаграмму деятельности принято располагать таким образом, чтобы действия следовали слева направо или сверху вниз.

Чтобы указать, где именно находится процесс, используется абстрактная точка «**маркер**» (или «токен»). Визуально на диаграмме маркер не показывается, данное понятие вводится только для удобства описания динамического процесса.

Переход маркера осуществляется между узлами.

Маркер:

может не содержать никакой дополнительной информации (пустой маркер) и тогда он называется **маркером управления** (control flow token)

может содержать **ссылку на объект** или **структуру данных**, и тогда маркер называется **маркером данных** (data flow token).

Диаграмма деятельности

Для создания диаграммы деятельности используются следующие узлы:

Узел управления (control node) – это абстрактный узел действия, которое координирует потоки действий

Начальный узел деятельности (или начальное состояние деятельности) (activity initial node) является узлом управления, в котором начинается поток (или потоки) при вызове данной деятельности извне

Конечный узел деятельности (или конечное состояние деятельности) (activity final node) является узлом управления, который останавливает все потоки данной диаграммы деятельности. На диаграмме может быть более одного конечного узла

Конечный узел потока (или конечное состояние потока) (flow final node) является узлом управления, который завершает данный поток. На другие потоки и деятельность данной диаграммы это не влияет

Объект, над которым выполняются действия. Это не обязательный элемент диаграммы, но в некоторых случаях необходимо показать объект иницирующий выполнение действий, или являющийся результатом его



Для отображения расширений сценария на диаграмме деятельности используются, так называемые узлы решения .

Узел решения предназначен для определения правила ветвления и различных вариантов дальнейшего развития сценария.

В точку ветвления входит ровно один переход, а выходит - два или более. Для каждого исходящего перехода задается булевское выражение, которое вычисляется только один раз при входе в точку ветвления.

Элемент **узел объединения**. Узел объединения имеет два и более входящих узла и один исходящий.



Для отображения условий соответствующих логическому оператору «и» на диаграмме используются синхронизационная черта.

Точка разделения обеспечивает разделение одного потока на несколько параллельных потоков:

- входит ровно один поток;
- выходит два и более потока, каждый из которых далее выполняется параллельно с другими.

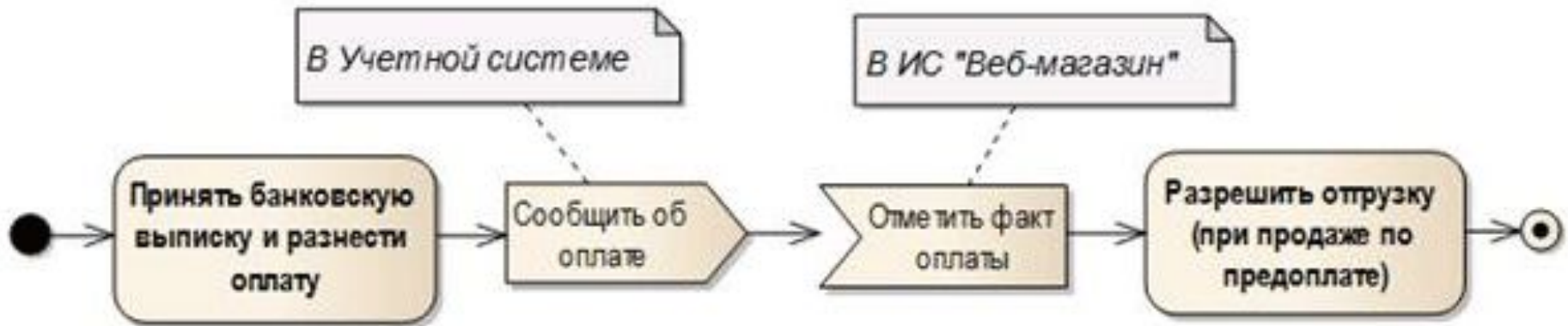
Точка слияния обеспечивает синхронизацию нескольких параллельных потоков.

- входят два или более потока, причем эти потоки выполняются параллельно;
- выходит ровно один поток, причем в точке слияния входящие параллельные потоки синхронизируются, то есть каждый из них ждет, пока все остальные достигнут этой точки, после чего выполнение продолжается в рамках одного потока.



Диаграмма деятельности может описывать поведение, на которое оказывают влияние **внешние события**, происходящие за пределами данной Системы.

На диаграмме это может быть показано при помощи изображения передачи сигнала. Передача сигнала может изображаться путем помещения между двумя действиями соответствующего элемента.



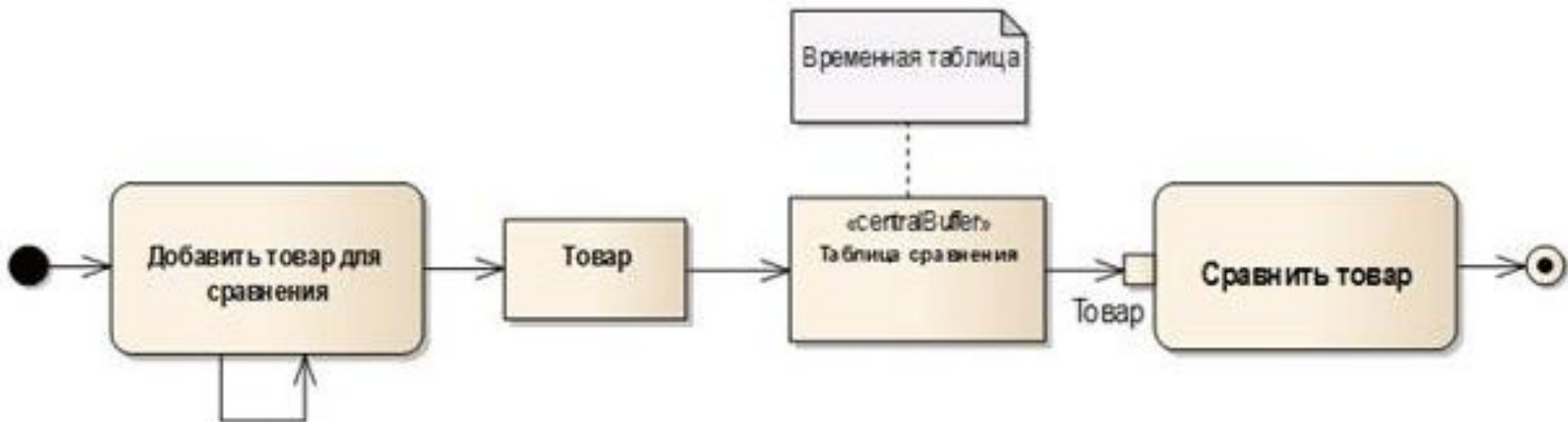
Допустимый графический вид сигнала



- Для отображения объекта, осуществляющего управление потоками из нескольких источников используется два специальных узла: **центральный буфер** и **хранилище данных**.

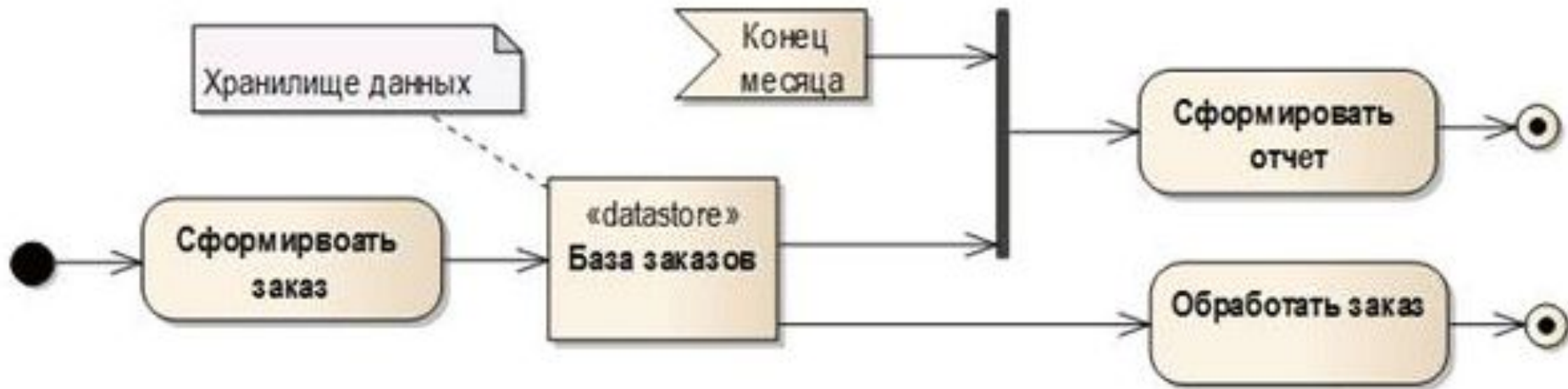
Центральный буфер - объект, который управляет потоками между множественными источниками и приемниками. На диаграмме центральный буфер представляется в виде объекта.

Данный объект может применяться на уровне описания реализации функций Системы для визуализации временных таблиц.



Диаграммы последовательности

Принципиальным отличием **Хранилища данных** является то, что оно содержит все поступившие данные и на выходе отдает лишь копии. Результатом действия «Сформировать заказ» является непосредственно заказ, который помещается в базу заказов.



Разделы группируют действия относительно какой-либо общей характеристики, при этом на течение потоков эта группировка никак не влияет.

Разделение при данной структуре диаграммы деятельности, зачастую символизируют роль пользователя или организационное подразделение, осуществляющее определенные действия в рамках данной деятельности.

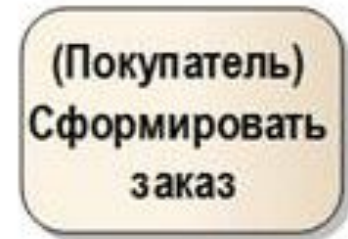
А) Вертикальное расположение дорожек

Покупатель	Менеджер

Б) Вложенные дорожки

Продавец	Контроллер	
	Менеджер	
Покупатель		

В) С указанием имени раздела в скобках перед именем действия



Спецификация UML предлагает несколько способов представления декомпозиции деятельности на диаграмме. Мы можем использовать обозначение под-деятельности (subactivity state), где указываем:

- наименование деятельности;
- предусловие и постусловие
- пиктограмму, информирующую о наличие развернутой диаграммы для данной деятельности .

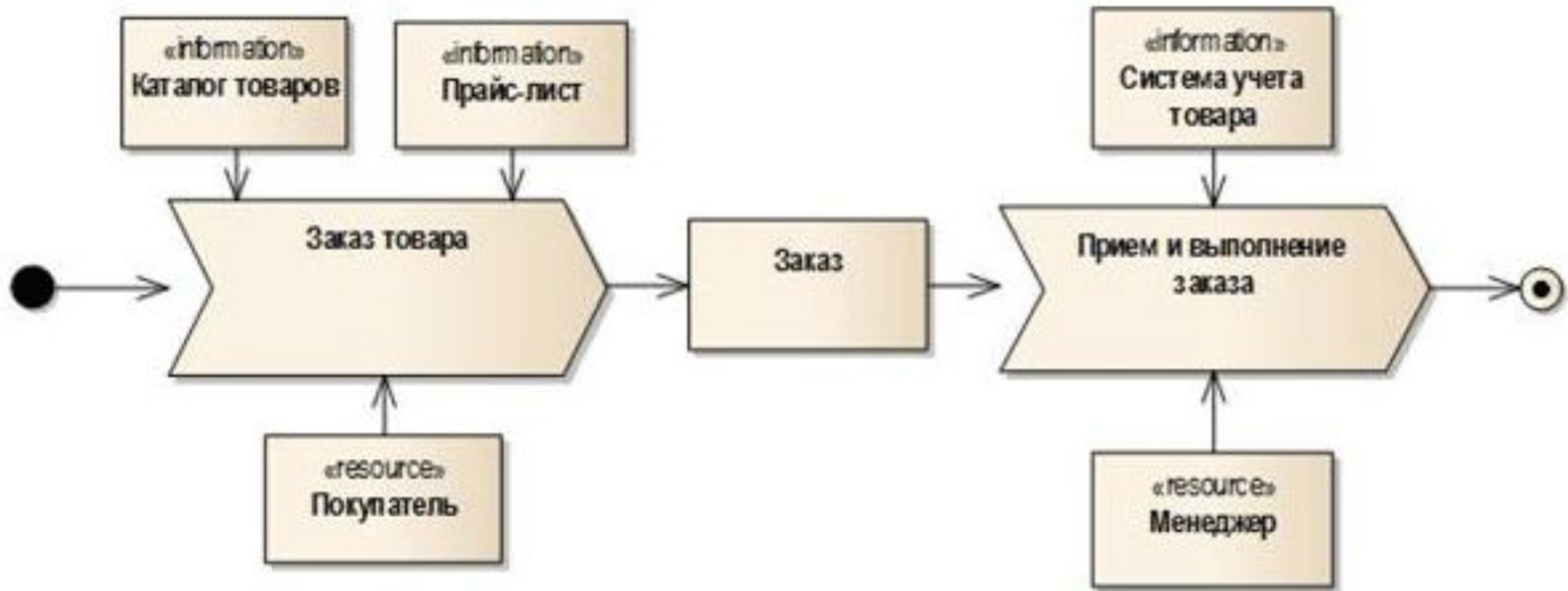
При необходимости описание последовательности действий для под-деятельности может быть размещено непосредственно на основной диаграмме. Для этого описание под-деятельности размещается в отдельный фрейм.

При таком способе декомпозиции мы можем указать:

- предусловия и постусловия;
- входные и выходные параметры (объекты);
- внутреннее устройство деятельности.

Для описания концепции процесса совершения покупки через интернет-магазин можно использовать диаграмму действия в стиле IDEF0, где указываются следующие параметры:

- входные и выходные данные;
- объекты, управляющие процессом (в нашем случае это каталог товаров и прайс-лист);
- используемые ресурсы (в нашем примере это Покупатель и Менеджер).



Для создания концептуальной модели необходимо выделять только основные процессы, так как вспомогательные процессы и сущности могут **перегружать диаграмму**. Также концептуальная модель должна включать только процессы верхнего уровня.

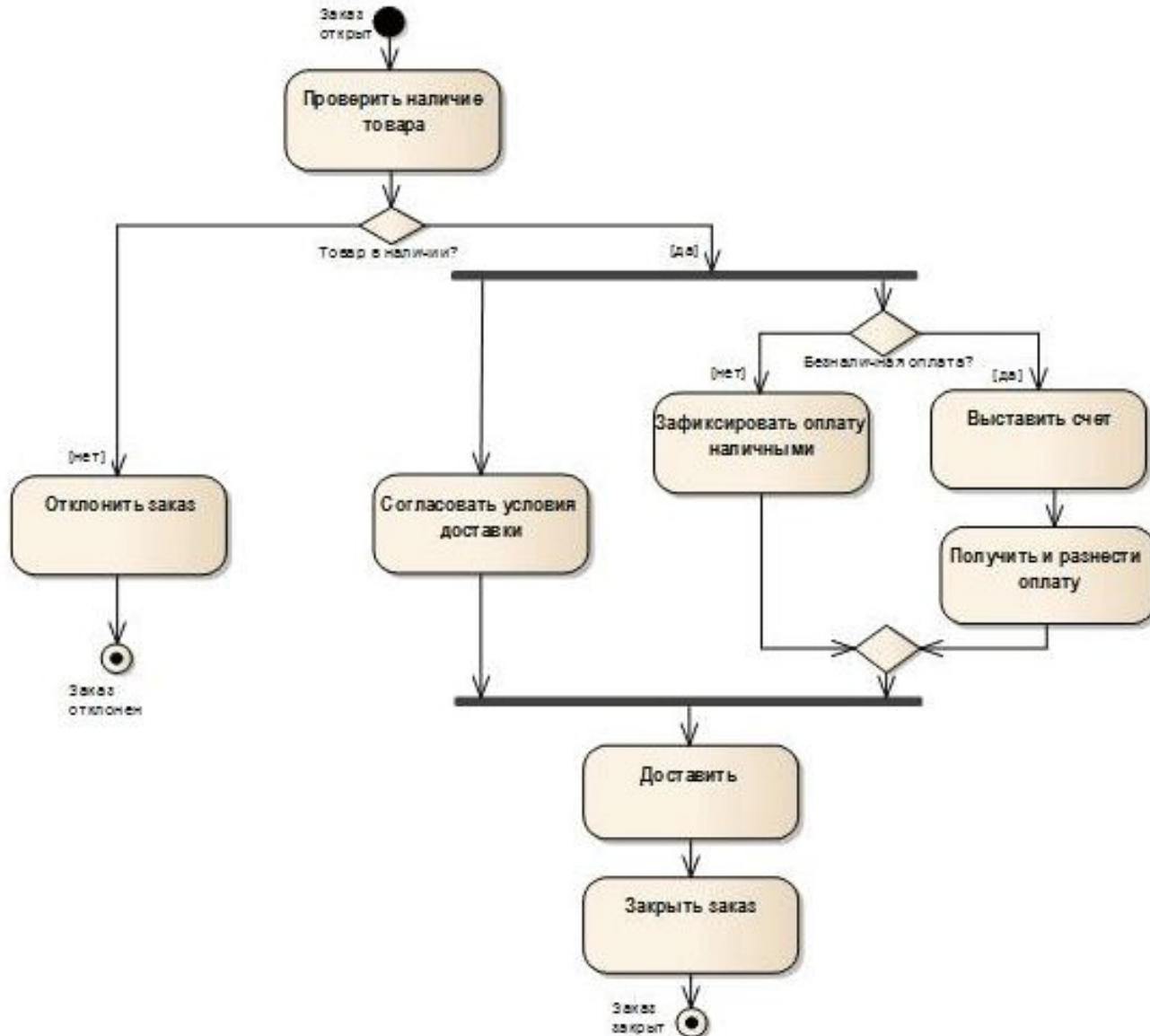


Диаграмма состояния

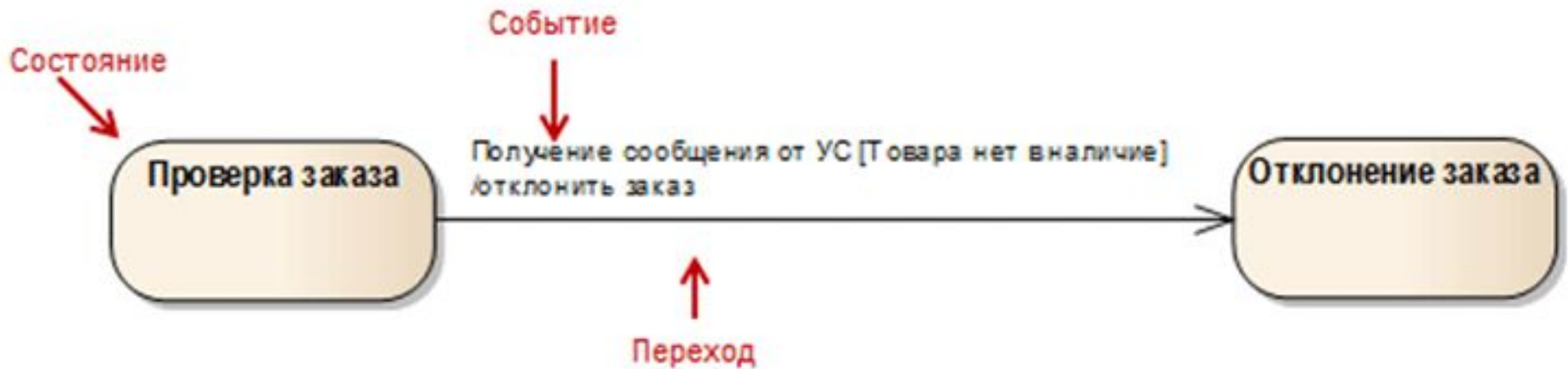
Система может становиться событийно управляемой, поэтому разработчикам важно знать, как должен реагировать тот или иной объект на определенные события.

Инициаторами событий могут быть как объекты самой Системы, так и её внешнее окружение.

- Также диаграмма состояний используется аналитиками для описания последовательности переходов объекта из одного состояния в другое.
- Диаграмма состояний показывает нам все возможные состояния, в которых может находиться объект, а также процесс смены состояний в результате внешнего влияния.

Основными элементами диаграммы состояний являются «Состояние» и «Переход».

Диаграмма состояний имеет схожую семантику с диаграммой деятельности. Таким образом, если для диаграммы деятельности отличие между понятиями «Деятельность» и «Действие» заключается в возможности дальнейшей декомпозиции, то на диаграмме состояний деятельность символизирует состояние, в котором объект находится продолжительное количество времени, в то время как действие моментально.



Переход может быть инициирован событием, которое также отражается на диаграмме состояний.

Состояние может содержать только **имя** или имя и **дополнительно список внутренних действий**.

Список внутренних действий содержит перечень действий или деятельности, которые выполняются во время нахождения объекта в данном состоянии. Данный список фиксированный. Список основных действий включает следующие значения:

entry - действие, которое выполняется в момент входа в данное состояние (входное действие);

exit - действие, которое выполняется в момент выхода из данного состояния (выходное действие);

do - выполняющаяся деятельность ("do activity") в течение всего времени, пока объект находится в данном состоянии;

defer - событие, обработка которого предписывается в другом состоянии, но после того, как все операции в текущем будут завершены.



Проверка заказа

- + entry / Создание нового заказа
- + exit / Изменение статуса заказа на "Подтвержден"



Различаются следующие виды событий:

Событие вызова (*call event*) - событие, возникающее при вызове метода класса. При срабатывании данного вида события объектом асинхронно создается Сигнал, который принимается другим объектом. Для указания на то, что некоторая операция посылает сигнал, можно воспользоваться зависимостью со стереотипом `send`.

Событие сигнала (*signal event*) - событие, возникающее при посылке сигнала. Если событие сигнала представляет возбуждение сигнала, то событие вызова предназначено для описания выполнения операции. Т.е. переход осуществляется при получении сигнала от другого объекта. В то время как сигнал является событием асинхронным, событие вызова обычно синхронно. Сигнал также может быть представлен на диаграмме в виде объекта со стереотипом «`signal`».

Событие таймера (*time event*) - возникает, когда истек заданный интервал времени с момента попадания автомата в данное состояние. В UML событие времени моделируется с помощью ключевого слова `after`(после), за которым следует выражение, вычисляющее некоторый промежуток времени.

Событие изменения (*change event*) - событие, которое возникает, когда некоторое логическое условие становится истинным, будучи до этого ложным. Данное событие моделируется с помощью ключевого слова `when`

Помимо основных узлов, на диаграмме состояний могут использоваться, так называемые, **псевдосостояния** – вершины которые не обладают поведением, и объект не находится в ней, а «мгновенно» ее проходит.

Под псевдосостояниями на диаграмме состояний понимаются, знакомые уже нам начальное и конечное состояние.

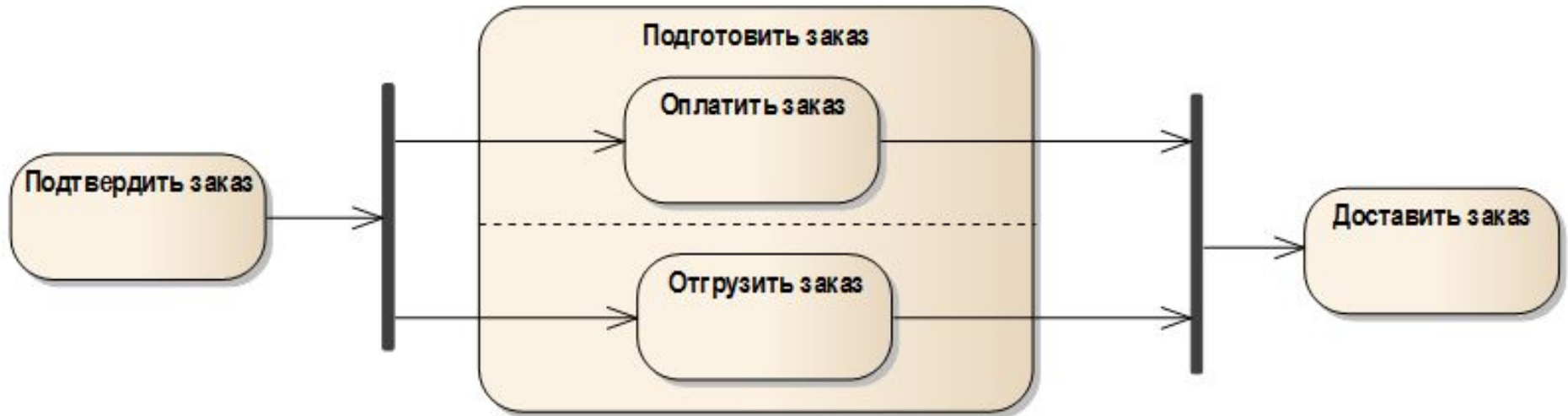
Начальное состояние обычно не содержит никаких внутренних действий и определяет точку, в которой находится объект по умолчанию в начальный момент времени.

Конечное состояние также не содержит никаких внутренних действий и служит для указания на диаграмме области, в которой завершается процесс изменения состояний в контексте конечного автомата.

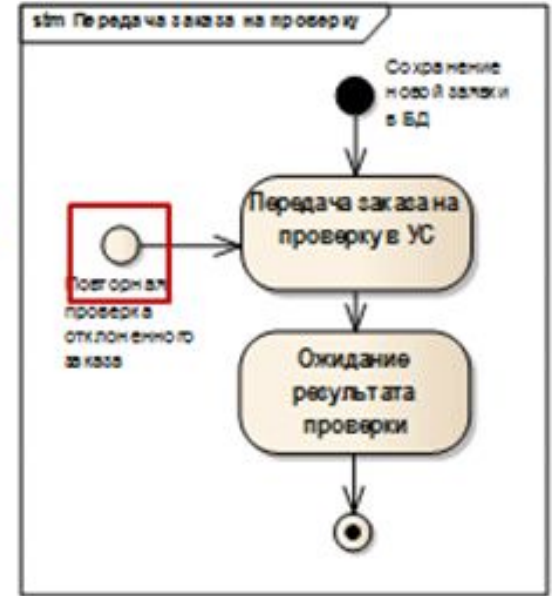
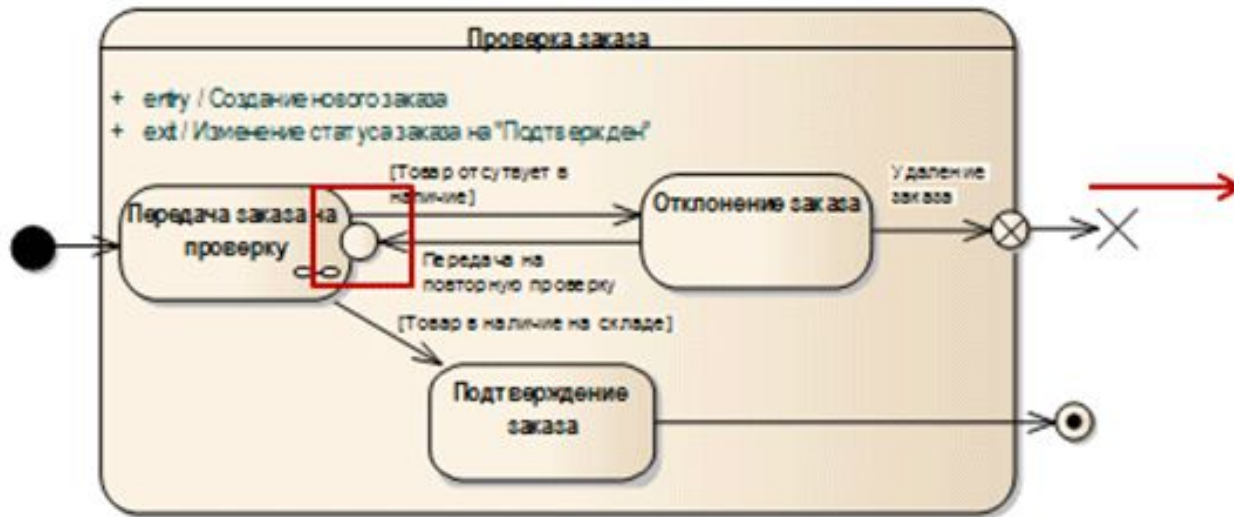
Если необходимо отразить уничтожение объекта используется **узел завершения** (*terminate node*), псевдосостояние, вход в который означает завершение выполнения поведения конечного автомата в контексте его объекта.

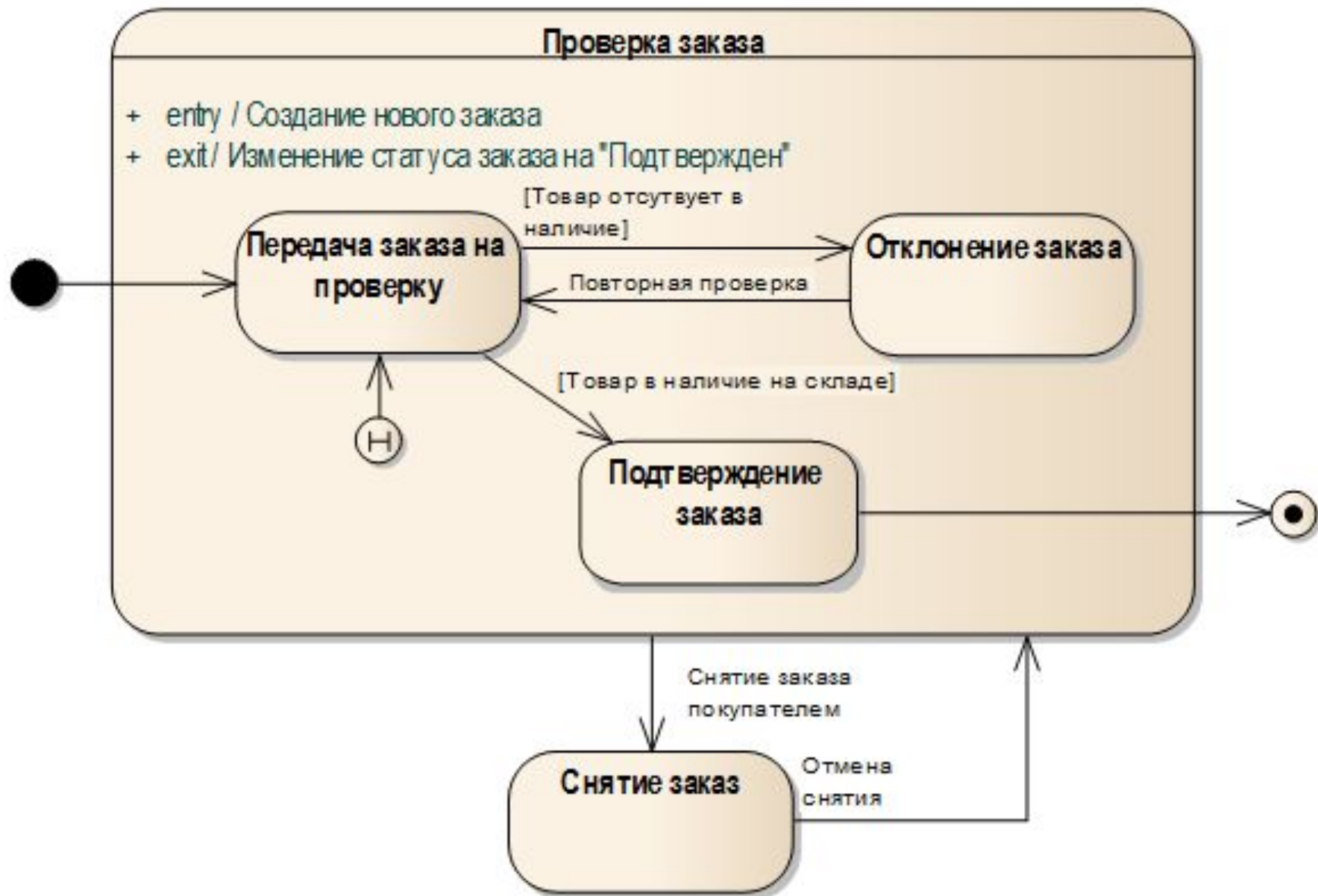


Составное состояние может содержать два или более параллельных подавтомата или несколько последовательных подсостояний.



Иногда возникает ситуация, когда необходимо показать переход из одного состояния в подсостояние композитного состояния, декомпозиция которого производится на отдельной диаграмме.





Диаграммы деятельности

- Подобно большинству других методов моделирования диаграммы деятельности обладают определенными достоинствами и недостатками, поэтому их лучше всего использовать в сочетании с другими методами.
- Самым большим достоинством диаграмм деятельности является возможность представления и описания параллельного поведения.
- Большой недостаток этих диаграмм заключается в том, что они не позволяют представить связи между действиями и объектами достаточно наглядно.

Желательно использовать диаграммы в следующих случаях:

- ***Анализ варианта использования.*** На этом этапе не интересует связь между действиями и объектами; нужно понять, какие действия должны иметь место и каковы зависимости в поведении системы.

Диаграммы деятельности

- **Понимание потока работ.** Прежде чем приступить к рассмотрению содержания вариантов использования, целесообразно привлечь диаграммы деятельности для лучшего понимания соответствующего бизнес-процесса. Эти диаграммы лучше разрабатывать совместно с бизнес-аналитиками, поскольку при этом можно понять особенности бизнес-процесса и возможности его изменения.
- **Описание сложного последовательного алгоритма.** В этом случае диаграмма деятельности не позволяет представить ничего сверх того, что может быть изображено на согласованной с обозначениями языка **UML** схеме алгоритма. При этом можно использовать принятые на схемах алгоритмов специальные обозначения.
- **Работа с многопоточными приложениями.**

Диаграммы деятельности

Диаграммы деятельности не следует использовать в следующих ситуациях:

- **Пытаться представить кооперацию объектов.** Диаграммы взаимодействия являются более простыми и обеспечивают более наглядное представление кооперации.
- **Пытаться представить поведение объектов в течение их жизненного цикла.** Для этой цели лучше использовать диаграмму состояний.
- **Представление сложных логических условий.** Для этой цели лучше использовать таблицу истинности.