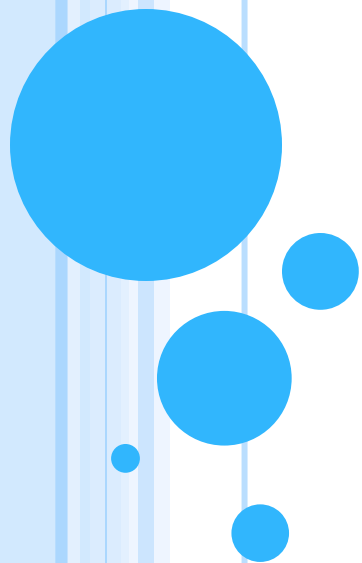


# ДИАГРАММЫ ДЕЯТЕЛЬНОСТИ



## Диаграмма деятельности (activity diagram)

- ◆ – диаграмма, которая изображает поведение объекта или системы с использованием моделей потока данных и потока управления
- ◆ *Деятельность (activity)* является спецификацией параметризованного поведения в форме координируемой последовательности подчиненных единиц, индивидуальными элементами которых являются действия
- ◆ Элементами, из которых состоят деятельности, являются действия
- ◆ *Действие (action)* представляет собой элементарную единицу спецификации поведения, которая не может быть далее декомпозирована в форме деятельности

## Узел деятельности (*activity node*)

- ◆ - является абстрактным классом для отдельных точек в потоке деятельности, соединенных дугами



- ◆ *Дуга деятельности (activity edge)* является абстрактным классом для направленных соединений между двумя узлами деятельности



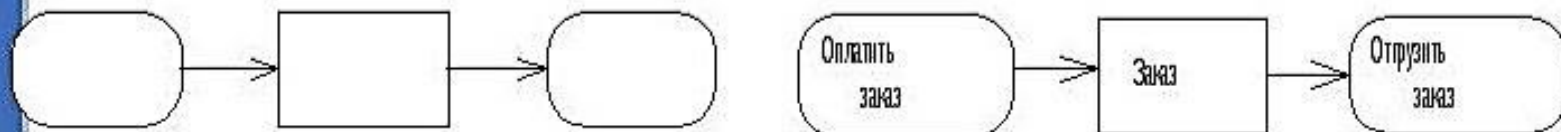
# Поток управления (control flow)

- ◆ - представляется в форме дуги деятельности, которая связывает между собой два узла деятельности и по которой передаются только маркеры управления
- ◆ При этом все маркеры управления, которые предлагаются узлом источником, предлагаются узлу цели, а вдоль дуги потока управления не могут следовать объекты и данные



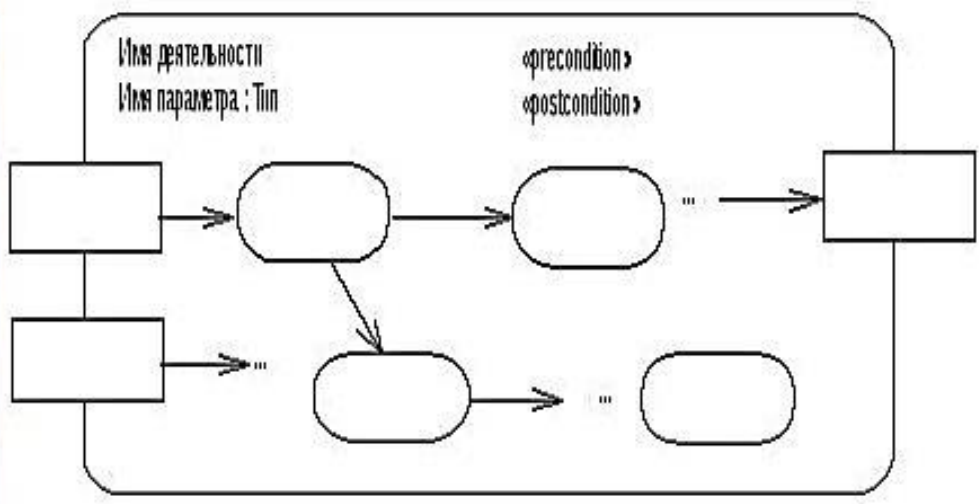
## Поток объектов (object flow)

- ◆ - представляется в форме дуги деятельности, по которой передаются только маркеры объектов или данных
- ◆ При этом все маркеры, предлагаемые узлом источником, предлагаются для узла цели с учетом ограничений, которые могут быть дополнительно специфицированы с помощью веса дуги
- ◆ Узлы объектов, соединенные потоком объектов с необязательными промежуточными узлами действий или управления, должны иметь совместимые типы





# Варианты нотация для деятельности



<code>«activity»</code> UML деятельности
атрибут : тип атрибут : тип
операция (параметры) операция (параметры)

# Семантика деятельности

- ◆ Семантика деятельности в языке UML 2.x основывается на потоке маркеров
- ◆ *Маркер (token)* – элемент модели, предназначенный для представления некоторого объекта, данных или управления и существующий на диаграмме деятельности в отдельном узле
- ◆ Каждый маркер отличается от любого другого, даже если он содержит то же значение, что и другой
- ◆ Любой узел деятельности может начать свое выполнение, только если удовлетворены специфицированные условия для его входных маркеров, причем эти условия зависят от вида узла
- ◆ Когда узел начинает свое выполнение, маркеры принимаются из некоторых или всех его входных дуг, а специальный маркер размещается в этом узле
- ◆ Когда узел завершает выполнение, специальный маркер удаляется из этого узла, а другие маркеры предлагаются в некоторых или всех его выходных дугах

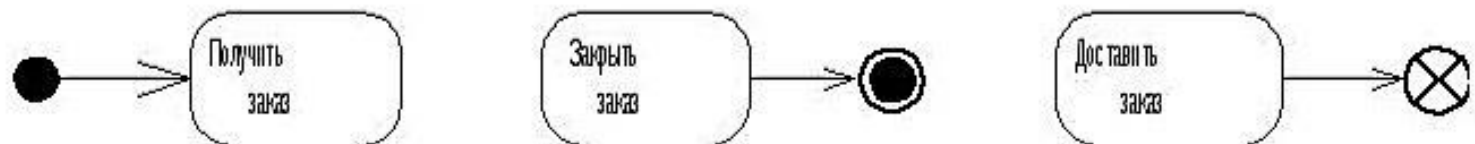
# Семантика действия

- ◆ Выполнение действия становится возможным, когда удовлетворены предварительные условия для его потоков управления и объектов
- ◆ Выполнение действия поглощает входные маркеры управления и маркеры объектов и удаляет их из источников дуг управления и из входных контактов
- ◆ Если на одной дуге являются доступными несколько маркеров управления, то они все поглощаются
- ◆ Действие продолжает выполнение до тех пор, пока оно не будет завершено
- ◆ После завершения действия оно предлагает маркеры объектов во все его выходные контакты, а маркеры управления во все выходящие из него дуги управления, и на этом формально оно заканчивается
- ◆ После окончания выполнения действия с помощью некоторой реализации должны быть восстановлены его ресурсы



## Узлы управления

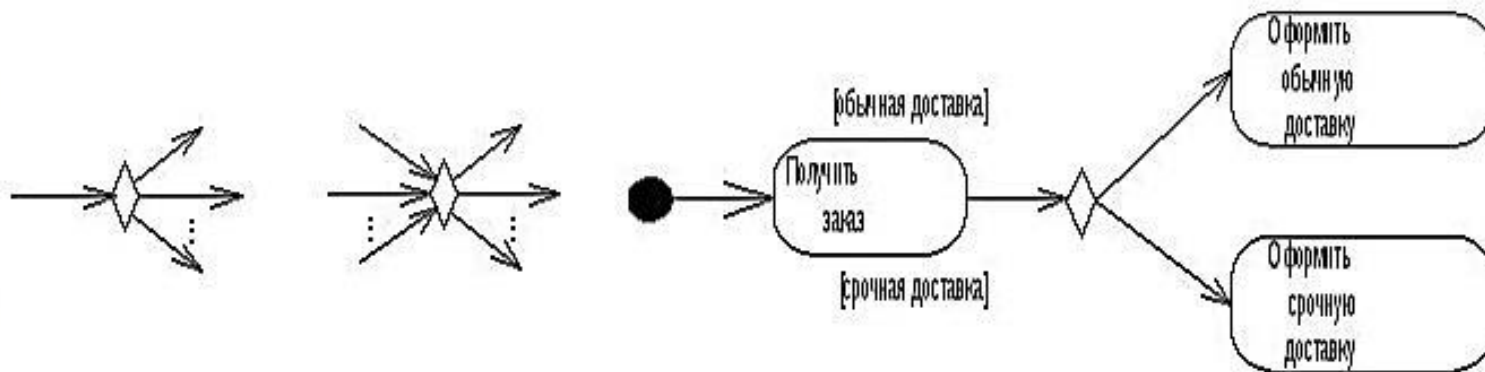
- ◆ *Начальный узел (initial node)* является узлом управления, в котором начинается поток при вызове деятельности
- ◆ *Узел финала деятельности (activity final node)* является узлом управления, который прекращает или останавливает все потоки в деятельности
- ◆ *Узел финала потока (flow final node)* является финальным узлом, который завершает отдельный поток управления или поток объектов, не завершая содержащей его деятельности



## Узел решения (*decision node*)

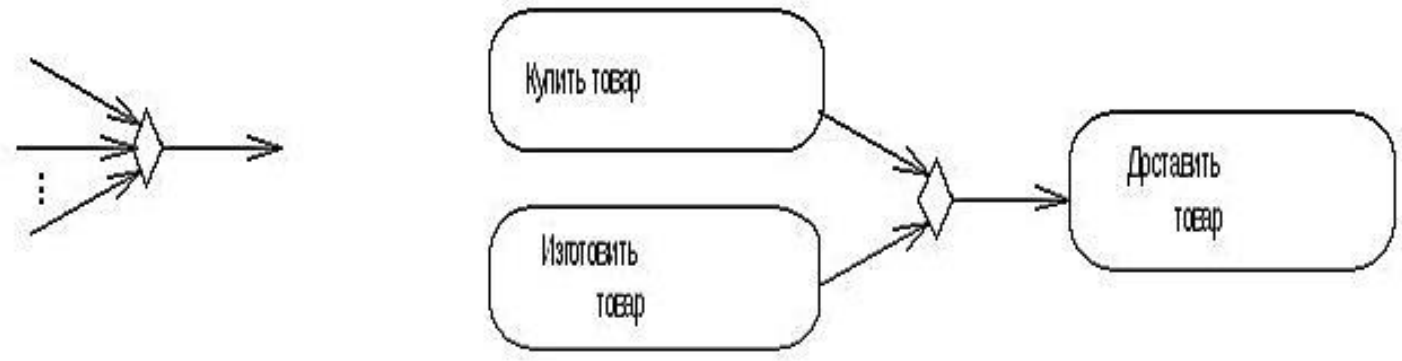
- ◆ - является узлом управления, который выбирает между выходящими потоками
- ◆ Если для узла решения при оценивании оказываются справедливыми более одного сторожевого условия, то семантика такого поведения в языке UML 2.x не определена, поскольку среди выходящих дуг возникает состязание за прием маркера
- ◆ При отсутствии дополнительной спецификации это может привести к несостоятельной (*ill-formed*) модели
- ◆ Чтобы гарантировать выполнение только одного сторожевого условия, иногда удобно использовать процедуру проверки до первого истинного условия

# Варианты изображения узла решения



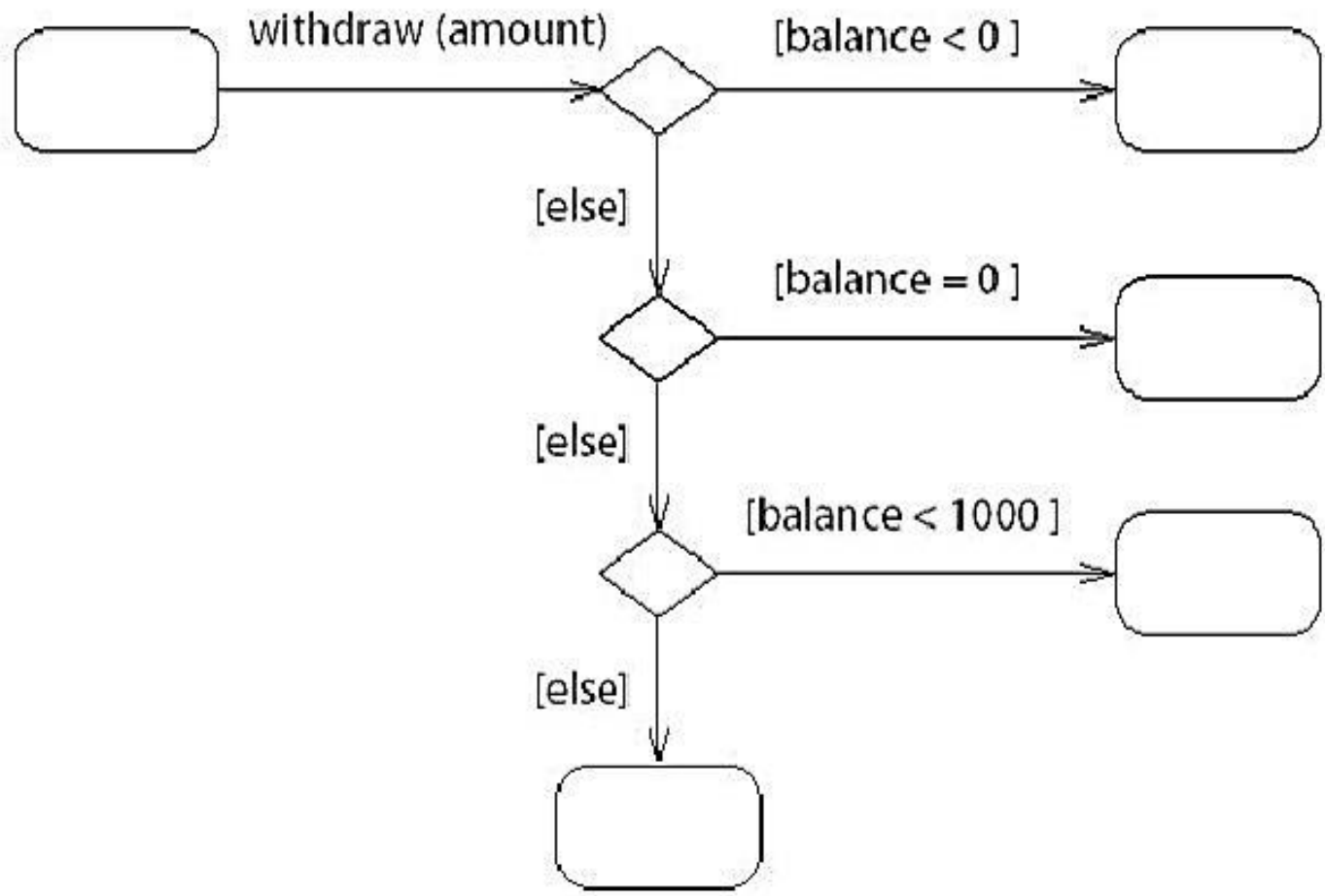
# Узел слияния (*merge node*)

- ◆ - является узлом управления, который соединяет вместе несколько альтернативных потоков





# Пример последовательного ветвления



## Узел разделения (*fork node*)

- ◆ - является узлом управления, который расщепляет поток на несколько параллельных потоков
- ◆ Дуги, выходящие из узла разделения, дополнительно могут иметь сторожевые условия, при невыполнении которых могут возникать паузы с передачей маркеров по этим дугам
- ◆ В данном случае предполагается, что никакие из находящихся далее узлов соединения не зависят от прохода маркеров, передающихся через дугу со сторожевым условием
- ◆ Если этого исключить нельзя, то необходимо ввести узел решения с последующим узлом слияния



## Узел соединения (*join node*)

- ◆ - является узлом управления, который синхронизирует несколько потоков
- ◆ Узлы соединения могут иметь дополнительную логическую спецификацию условий, при выполнении которых они должны генерировать маркер на выходе
- ◆ Если для узла соединения существуют маркеры во всех его входящих дугах, то выходящей дуге предлагаются маркеры согласно следующим правилам:
- ◆ Если все маркеры, предлагаемые на входящих дугах, являются маркерами управления, то выходящей дуге предлагается один маркер управления

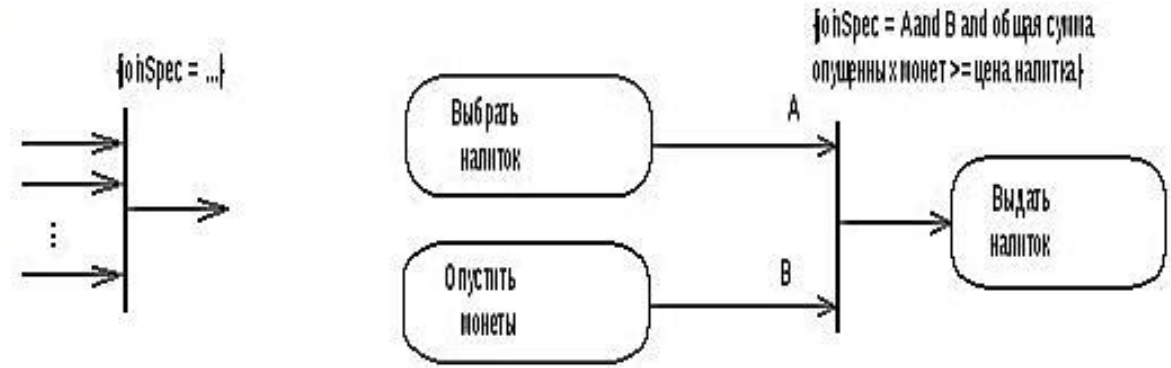
## Примеры изображения узла соединения

- ◆ Если часть маркеров, предлагаемых на входящих дугах, являются маркерами управления, а другие являются маркерами данных, то выходящей дуге предлагаются только маркеры данных
- ◆ Они предлагаются выходящей дуге в том же порядке, в каком предлагаются на входе этого узла соединения



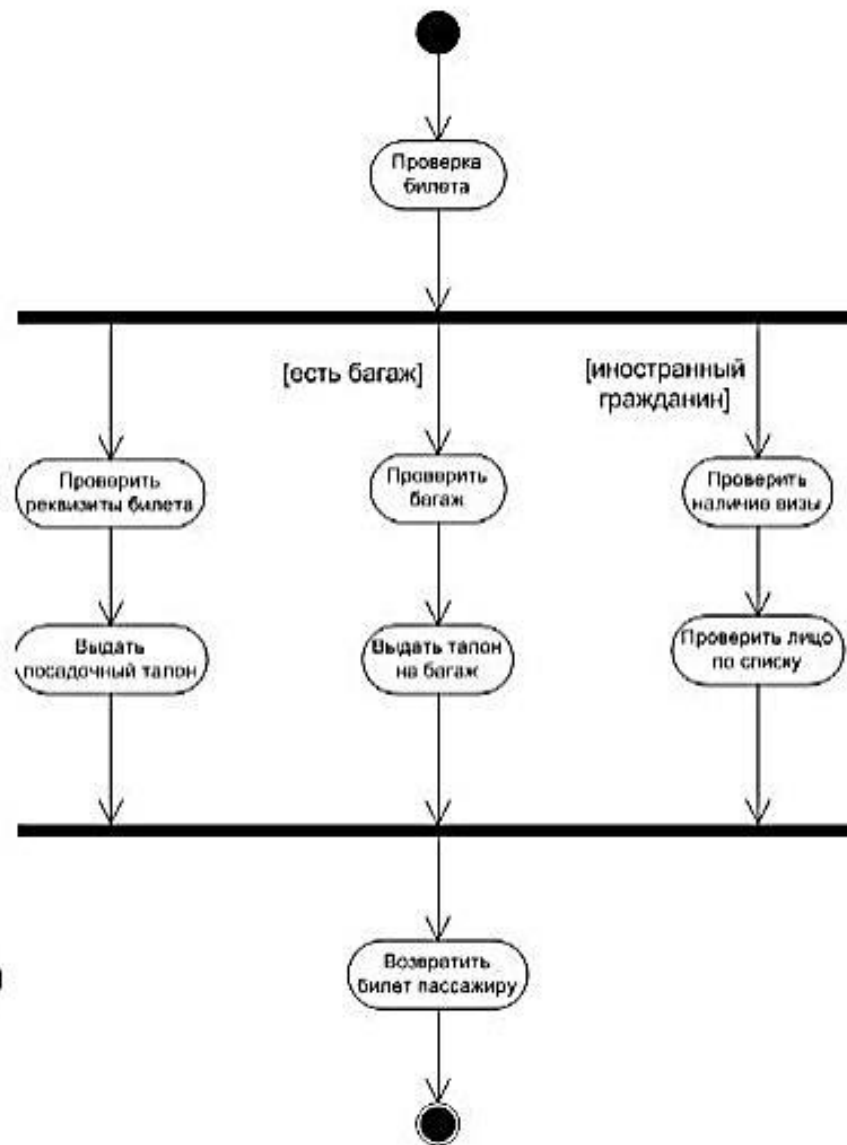


# Примеры изображения узла соединения с дополнительной спецификацией



# Пример условно-параллельных деятельностей

- ◆ Дуги, выходящие из узла разделения, дополнительно могут иметь сторожевые условия, при невыполнении которых могут возникать паузы с передачей управления по этим дугам

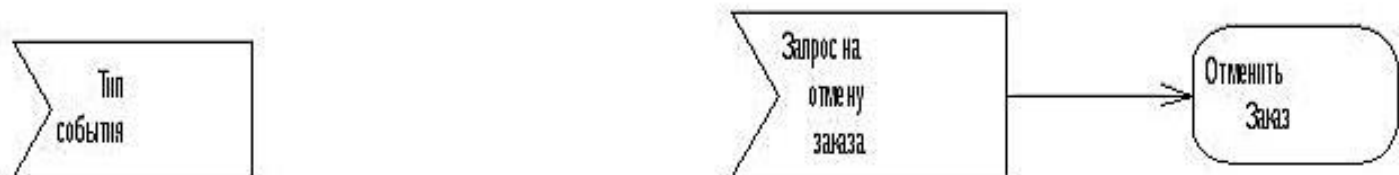


# Специальные действия

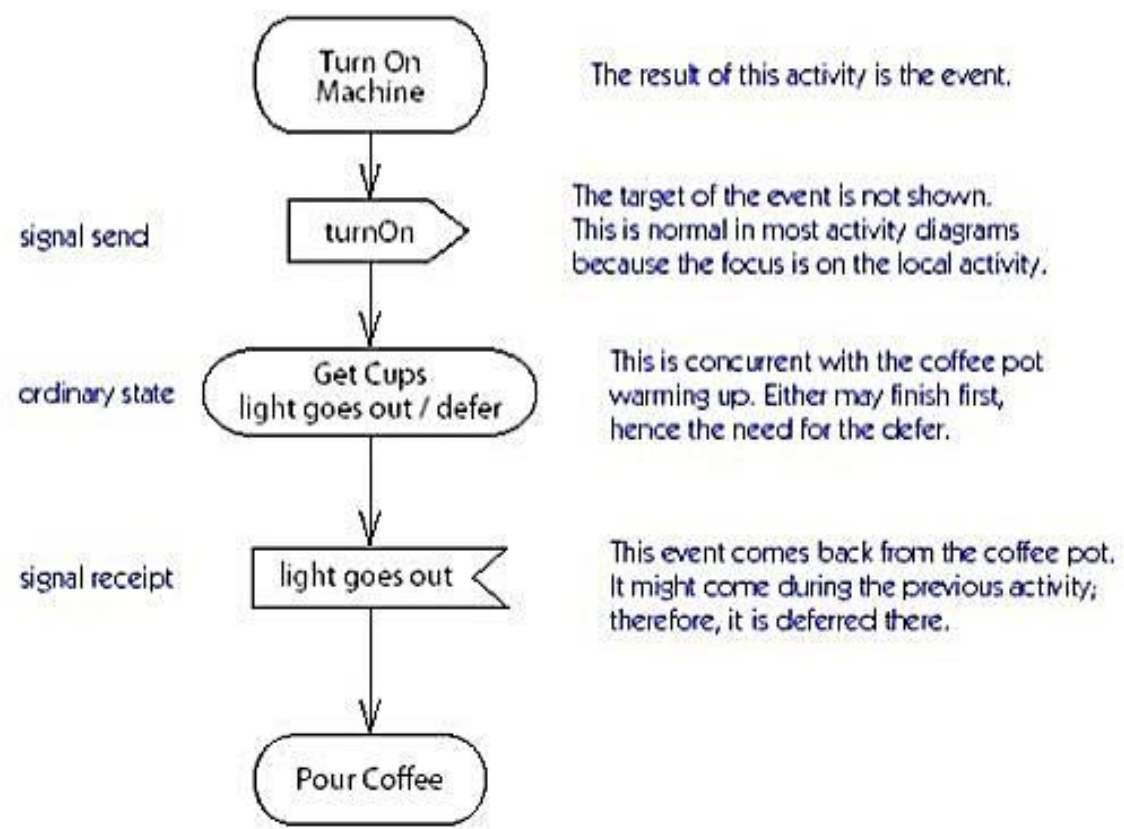
- ◆ *Действие передачи сигнала (send signal action)* является действием, которое на основе своих входов создает экземпляр сигнала и передает его объекту цели



- ◆ *Действие приема события (accept event action)* является действием, которое ожидает наступление некоторого события

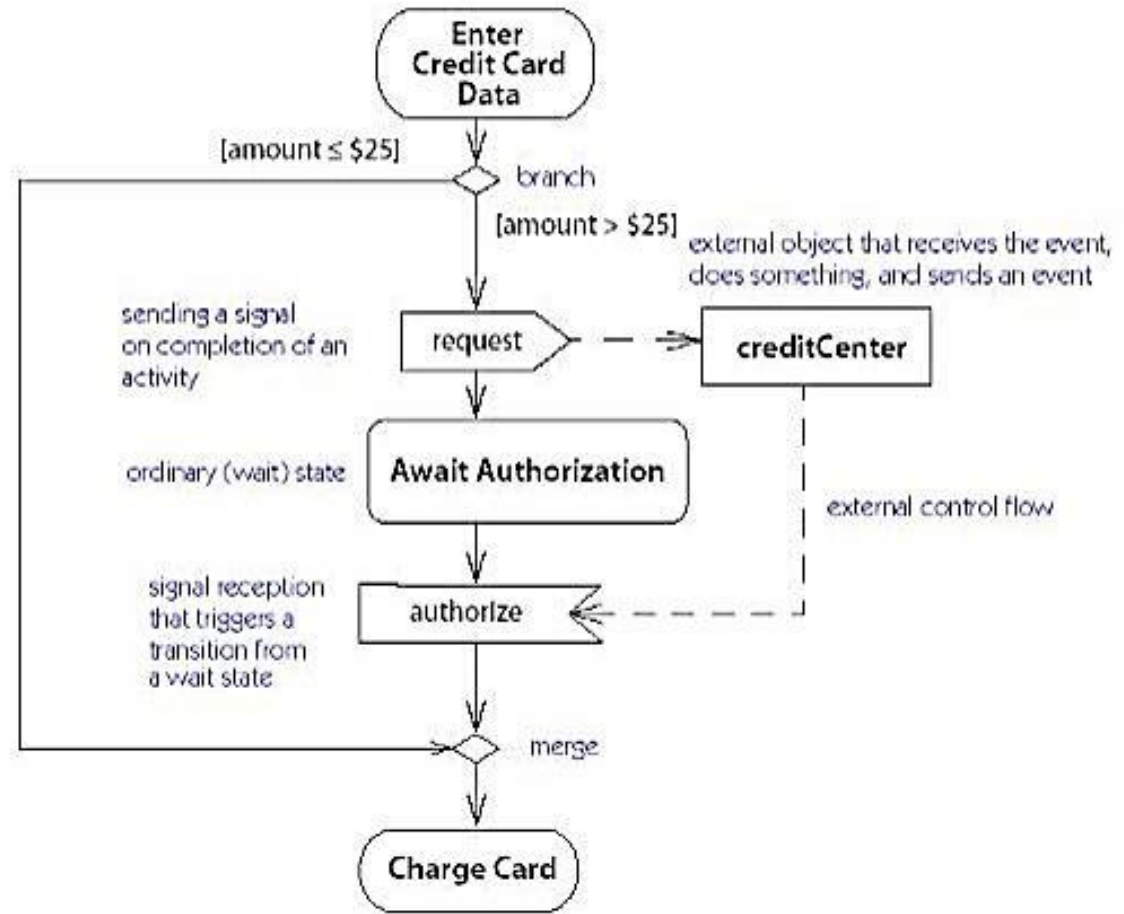


# Пример передачи и приема сигнала





# Пример передачи и приема сигнала



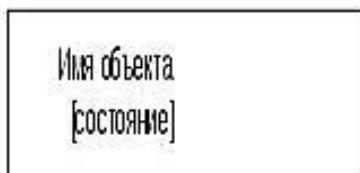
# Ожидание временного события (accept time event action)

- ◆ Специальный случай действия приема события
- ◆ Если наступившее событие является временным событием, то объект должен зафиксировать значение момента времени, когда наступило соответствующее событие



## Узел объекта (*object node*)

- ◆ - является узлом абстрактной деятельности, которая является частью определяющего потока объектов в деятельности
- ◆ Узел объекта для маркеров объектов, находящихся в специальном состоянии, дополнительно содержит спецификацию этого состояния, которая записывается в прямых скобках ниже имени типа
- ◆ Узел объекта для маркеров, содержащих множества объектов различных типов, содержит имена всех этих объектов
- ◆ Узлы объектов с сигналом в качестве типа изображаются с помощью специального символа, внутри которого записывается имя типа сигнала



## Узлы объектов с дополнительными ограничениями

- ◆ Узел объекта с ограниченной верхней границей должен содержать явную спецификацию этой границы
- ◆ Узел объекта может специфицировать порядок управления маркерами объектов, отличающимся от FIFO, в форме дополнительной спецификации
- ◆ Дополнительная спецификация выбора маркеров помечается ключевым словом «selection», размещенном в символе примечания и присоединенном к символу узла



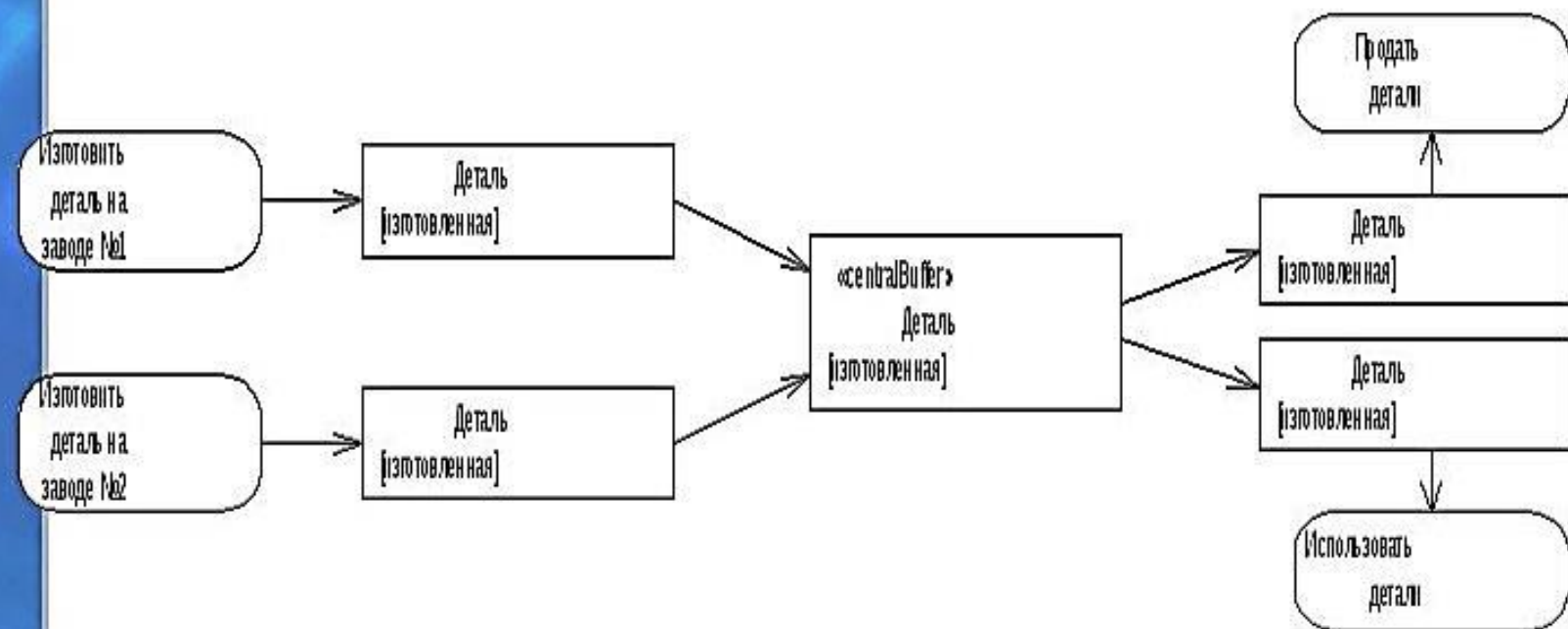


## Центральный буфер (*central buffer*)

- ◆ - является узлом объекта для управления потоками из нескольких источников и мест назначения
- ◆ Центральный буфер принимает маркеры из расположенных до него узлов объектов и отправляет их дальше в расположенные после него узлы объектов
- ◆ Центральный буфер не имеет непосредственных соединений с узлами действия
- ◆ Хотя все узлы объектов имеют функциональность буфера, но центральные буферы отличаются тем, что они не привязаны к действиям, как рассматриваемые далее контакты и параметры



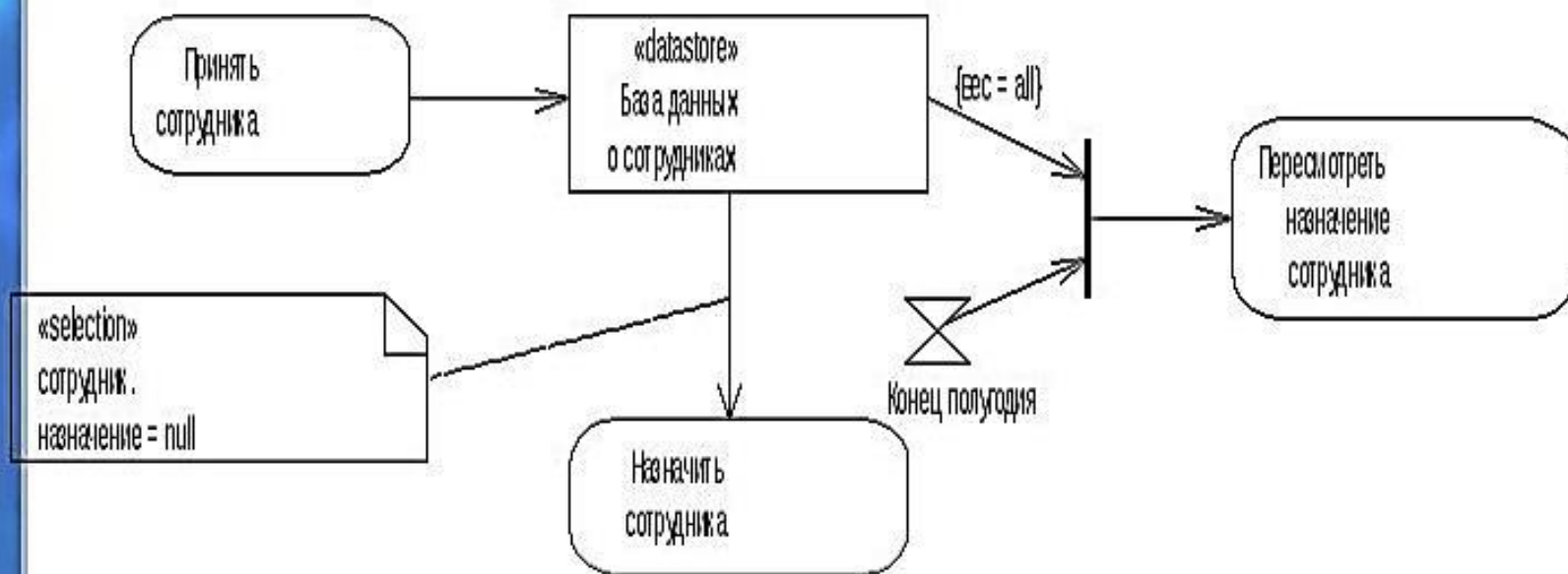
# Пример использования центрального буфера



## Хранилище данных (*data store*)

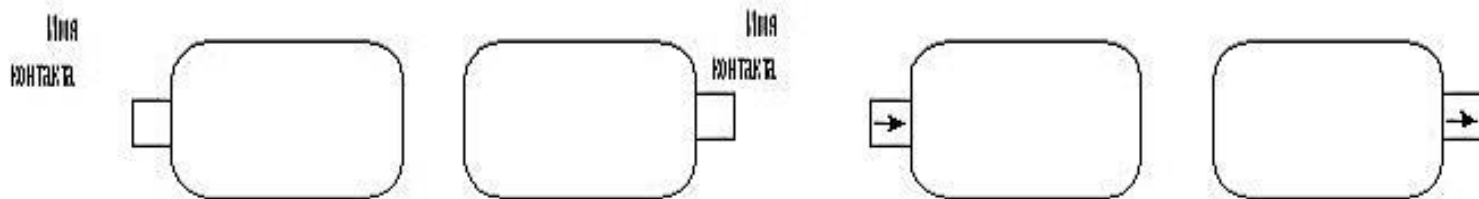
- ◆ - является разновидностью центрального буфера для постоянного хранения объектов или другой информации
- ◆ Хранилище данных принимает все маркеры, которые в него входят, сохраняет их и копирует, когда они выбираются для дальнейшего использования
- ◆ Если входящий маркер представляет объект, который уже имеется в хранилище, то он заменяет любой маркер в узле объекта, содержащий этот объект
- ◆ Копирование маркеров в хранилище данных выполняется всякий раз, когда они выбираются для следования по выходным дугам
- ◆ Ситуация представляется таким образом, что маркеры как бы никогда не покидают хранилище данных

# Пример использования хранилища данных

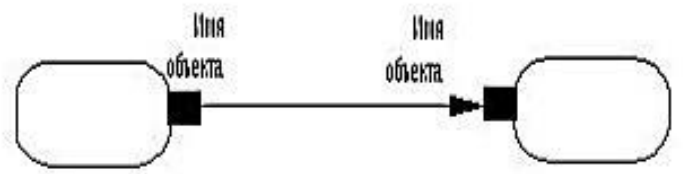
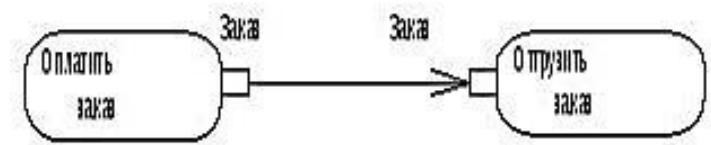


# Входные и выходные контакты объектов

- ◆ *Входной контакт (input pin)* является узлом объекта, который принимает значения от других действий в форме потока объектов
- ◆ *Выходной контакт (output pin)* является узлом объекта, который поставляет значения другим действиям в форме потока объектов.



# Варианты изображения передачи объекта и потока объектов

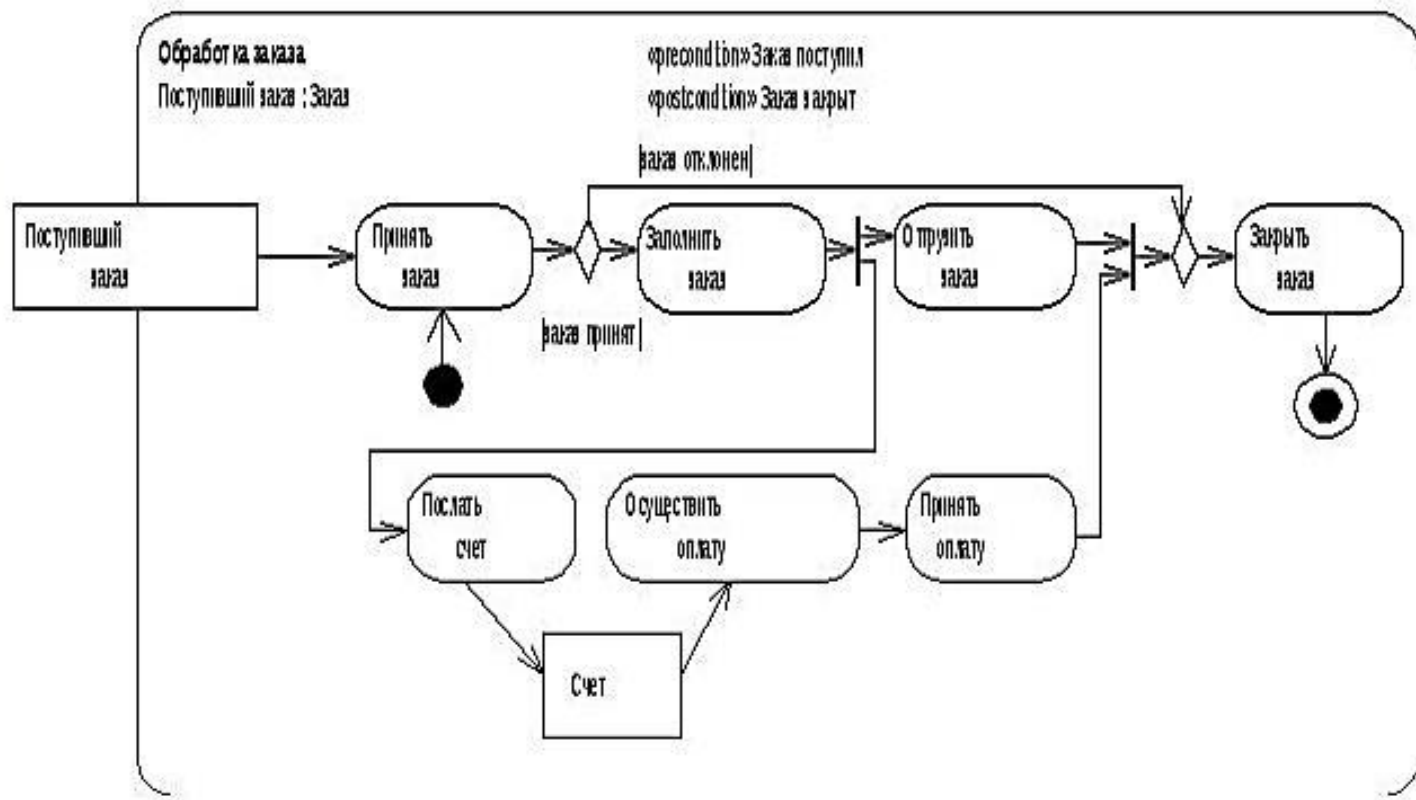




## Узел параметра деятельности (*activity parameter node*)

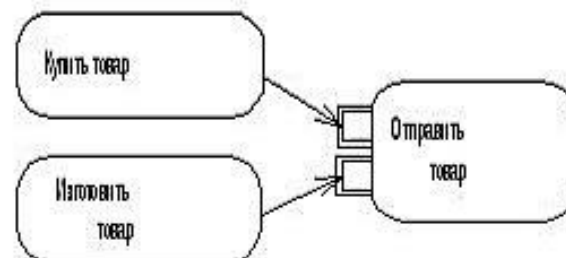
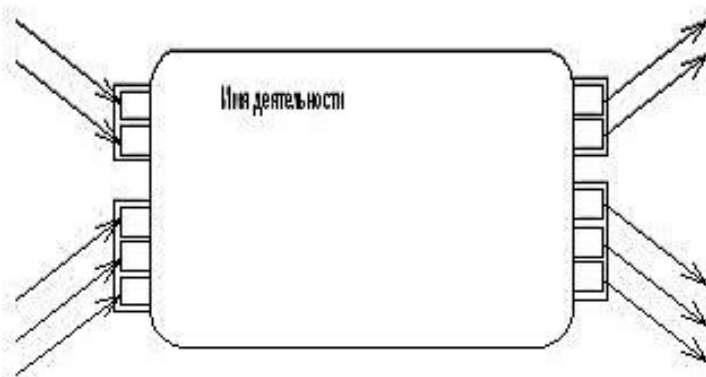
- ◆ - является узлом объекта для моделирования входов и выходов деятельности
- ◆ Вызов и выполнение некоторой деятельности с параметрами выполняется в соответствии со следующими правилами:
- ◆ Для вызываемой деятельности должны быть достигнуты все необходимые не потоковые входы
- ◆ Если все входы в деятельность являются входами потока объектов, то для начала выполнения деятельности должен быть достигнут по крайней мере один из узлов входных параметров этой деятельности
- ◆ Во время выполнения деятельности на ее узлах входных параметров могут поглощаться дополнительные маркеры, а на ее узлах выходных параметров – отправляться
- ◆ Для завершения деятельности должны быть достигнуты все специфицированные узлы выходных параметров этой деятельности

# Пример деятельности с входным параметром



## Множество параметров (*parameter set*)

- ◆ - предназначено для моделирования отдельного альтернативного потока объектов на входе или выходе деятельности
- ◆ Если для деятельности специфицировано несколько множеств входных параметров, то для вызова деятельности необходимо наличие маркеров объектов для входных параметров только для одного такого множества



# Разбиение деятельности (*activity partition*)

- элемент модели, предназначенный для группировки действий, которые относятся к одной деятельности и имеют некоторую общую характеристику

Имя разбиения	
---------------	--

Имя измерения

Имя измерения	Имя разбиения	
	Имя подразбиения	Имя подразбиения

Имя измерения 1

Имя измерения 1	Имя разбиения 2	Имя измерения 2	
		Имя разбиения 3	Имя разбиения 4
Имя разбиения 1			



## Правила использования разбиений на диаграмме деятельности

- ◆ Любые узел или дуга деятельности не могут одновременно принадлежать двум или более разбиениям в одном и том же измерении.
- ◆ Разбиения не влияют на потоки маркеров, за которые несут ответственность экземпляры классификаторов, представленными отдельными разбиениями.
- ◆ Разбиения одного измерения и уровня вложенности должны быть представлены частями внутренней структуры одного и того же классификатора.
- ◆ Разбиение может быть представлено атрибутом, а ее подразбиения – значениями этого атрибута
- ◆ Если разбиение имеет некоторое измерение, то оно не может содержаться ни в каком другом разбиении.



# Обработчик исключения (*exception handler*)

- ◆ - представляет собой спецификацию деятельности, выполнение которой происходит в случае исключения в ходе выполнения некоторого защищенного узла
- ◆ Правила использования обработчиков исключений:
- ◆ Тело обработчика должно иметь один вход, и этот вход является тем же, что и вход исключения.
- ◆ Контакты результата тела обработчика исключения должны соответствовать по числу и типу контактам результата защищенного узла.
- ◆ Защищенный узел и узел тела обработчика должны быть одного и того же уровня вложенности.
- ◆ В противном случае нотация может быть интерпретирована ошибочно как прерывающая дуга, которая пересекает границу.
- ◆ К одному защищенному узлу могут быть присоединены несколько обработчиков исключений, каждый со своими собственными стрелками молнии