

Практическое занятие 6: BPMN События

ДИСЦИПЛИНА	Архитектура индустриального программного обеспечения
ИНСТИТУТ	Институт перспективных технологий и индустриального программирования
КАФЕДРА	Кафедра индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Практические занятия
ПРЕПОДАВАТЕЛЬ	Зарипова Виктория Мадияровна
СЕМЕСТР	1 семестр, 2024-2025 гг.

Тема

События BPMN: эскалация, ошибка, отмена, компенсация, сигнал.

Цель

Ознакомление студентов с понятием и использованием различных типов событий в BPMN, таких как эскалация, ошибка, отмена, компенсация и сигнал.

Входные знания

Студенты должны иметь предварительное понимание основ бизнес-процессов и их моделирования в рамках методологии BPMN.

Итоги занятия:

1. Студент должен быть знаком с понятием и функциональностью каждого типа событий: эскалация, ошибка, отмена, компенсация и сигнал.
2. Студент должен уметь использовать каждый тип события в своих бизнес-процессах в соответствии с их особенностями и требованиями.
3. Студент должен иметь понимание о том, как и когда использовать каждый тип событий для обработки и управления исключительными ситуациями, передачи информации или уведомлений, а также для обеспечения целостности и надежности бизнес-процессов.
4. При успешном выполнении лабораторной работы студент должен иметь готовую модель бизнес-процесса, в которой события эскалации, ошибки, отмены, компенсации и сигнала применены на соответствующих этапах процесса для достижения требуемого функционального поведения и взаимодействия с окружающей системой.

План работы

Вам необходимо научиться моделировать более сложные бизнес-процессы с использованием расширенных элементов BPMN. Мы будем использовать такие события как эскалация, ошибка, отмена, компенсация и сигнал более эффективного и автоматизированного процесса.

- Смоделируйте ошибочные сценарии на BPMN диаграммах ваших бизнес-процессов
- Добавьте там где это необходимо отмену и компенсацию
- В случае необходимости рассылки на все ваши сервисы используйте событие сигнала

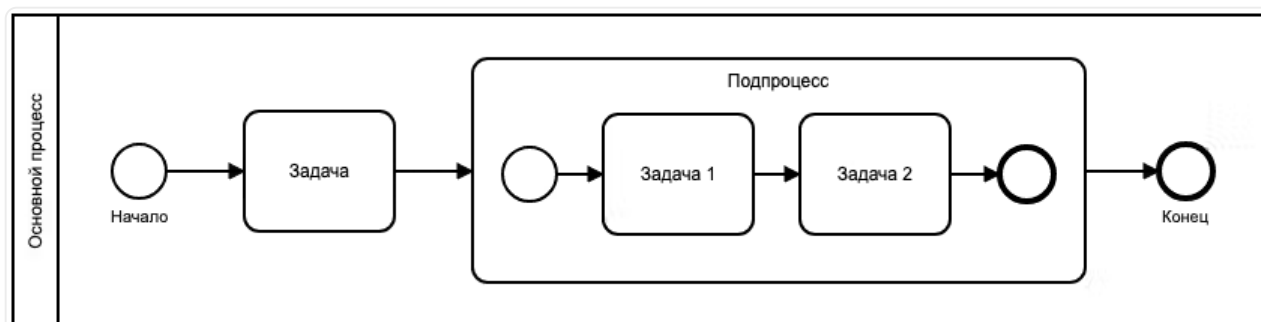
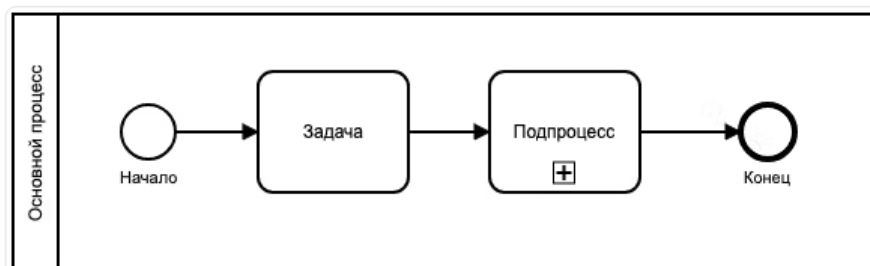
Обзор дополнительных элементов BPMN

Подпроцессы

Обычный подпроцесс (Subprocess)

Обычный подпроцесс BPMN может располагаться только в рамках родительского процесса, которому он принадлежит. Он не может содержать внутри пулы и дорожки, но может быть помещен в пул или дорожку родительского процесса. Кроме того, он может иметь только абстрактное стартовое событие. События-сообщения и события-таймеры не могут запускать подпроцесс. Обычный подпроцесс используется в родительском процессе с целью:

- Упрощения диаграммы процесса путем скрывтия определенной последовательности действий на диаграмме.
- Объединения частей родительского процесса с добавлением определенных событий и маркеров.



Как происходит выполнение подпроцессов BPMN

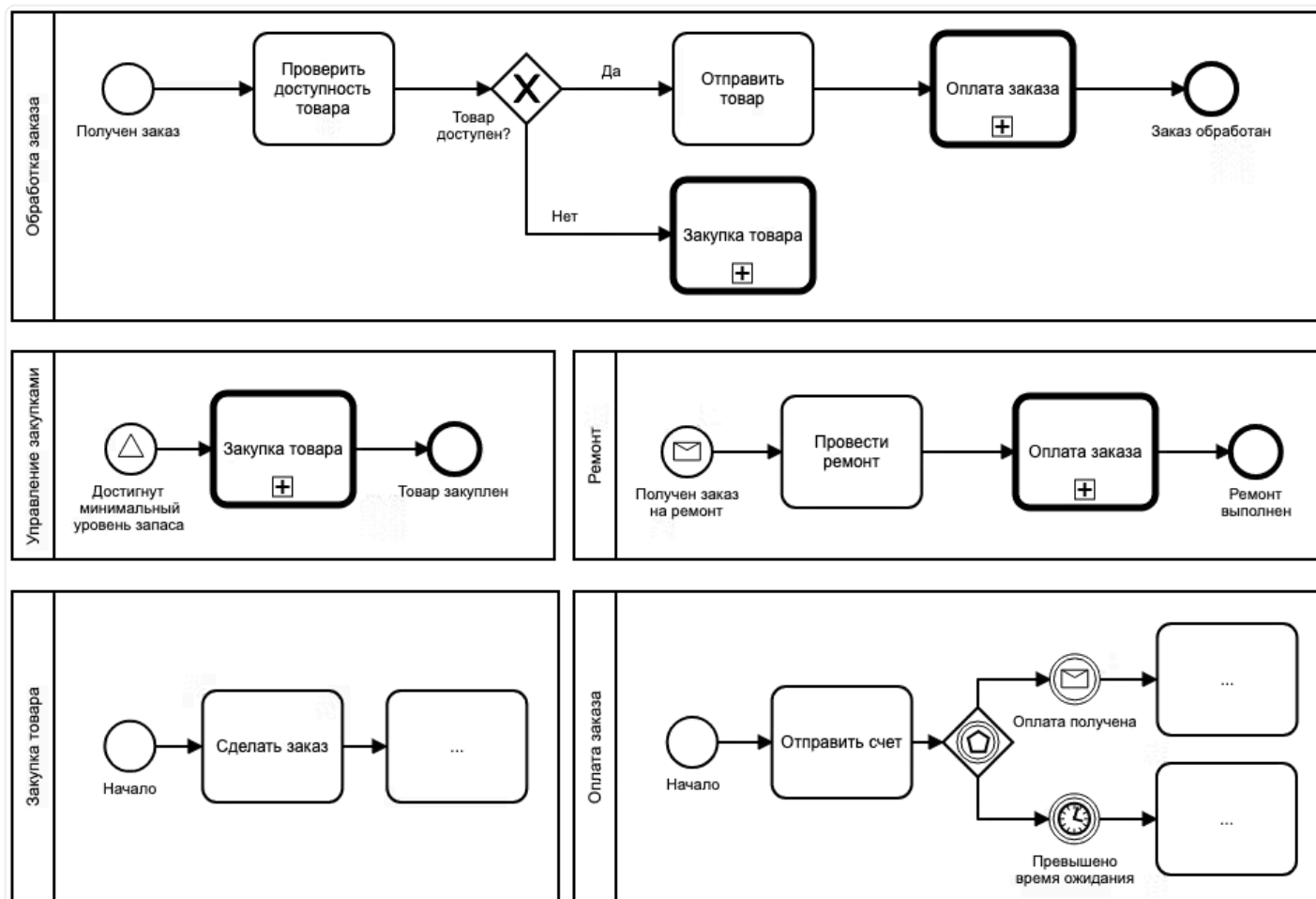
Ключевая особенность заключается в том, что поток управления родительского процесса не может пересекать границы подпроцесса. Следовательно, выполнение **подпроцессов BPMN** происходит по следующим принципам:

- Родительский процесс имеет свой поток управления.
- Подпроцесс запускается потоком управления родительского процесса.
- Внутри подпроцесса создается отдельный поток управления, проходящий от его начала до конца. В это время поток управления родительского процесса ожидает завершения подпроцесса.
- Когда подпроцесс завершится, управление автоматически передается в родительский процесс, и его выполнение продолжается дальше.
- Когда подпроцесс завершится, управление автоматически передается в родительский процесс, и его выполнение продолжается дальше.

Вызов (Call)

Используется для вызова процессов или задач, которые могут быть переиспользованы. Например, глобальный процесс «Вход в систему» может использоваться в разных процессах, причем, вместо дублирования данный процесс моделируется один раз и затем вызывается там, где это необходимо.

Глобальные подпроцессы BPMN могут использоваться в различных процессах многократно. На диаграмме ниже приведен пример подпроцесса «Закупка товара», который вызывается в процессах «Обработка заказа» и «Управление закупками». В первом случае, если клиент купил товар и его нет в наличии. Во втором – если достигнут минимальный уровень запаса товара. Другой пример на этом же рисунке – подпроцесс «Оплата заказа», который вызывается в процессах «Обработка заказа» и «Ремонт».



Поскольку глобальные **подпроцессы вызываются BPMN** из многих других процессов, это отражается на передаче данных между ними. Так, обычные подпроцессы могут напрямую считывать все данные родительского процесса. Однако, для глобальных подпроцессов требуется явное указание, какую информацию нужно получить из вызывающего процесса. Это может показаться всего лишь техническим аспектом, о котором нужно знать, но можно не беспокоиться. Тем не менее, после внимательного рассмотрения, Вы можете увидеть насколько существенна эта разница.

Так, например, когда Ваша бухгалтерия хочет выставить счет-фактуру на ремонт, ей всегда необходимы следующие данные:

- Адрес
- Дата поставки
- Описание товаров или услуг
- Сумма счета-фактуры
- Ожидаемая дата оплаты

Владельцы всех процессов, в которых предусмотрено выставление счета-фактуры (процесс обработки заказов, процесс ремонта и т.д.), должны предоставить эти данные. Учет будет нуждаться в данных в стандартном формате.

Это показывает, что BPMN требует сопоставления данных между вызывающими процессами и глобальными подпроцессами BPMN. BPMN заставляет нас формализовать многие вопросы, которые кажутся очевидными, или которые остались без внимания в процессе проектирования. Формализация – это лучший шанс сохранить целостность сложных бизнес-процессов в быстро меняющихся условиях.

Спонтанный (Ad-Hoc)

Спонтанный (Ad-Hoc) подпроцесс BPMN – это группа действий, взаимоотношения между которыми не установлены. Исполнители сами определяют последовательность и количество повторений этих действий, а также могут игнорировать выполнение действий.

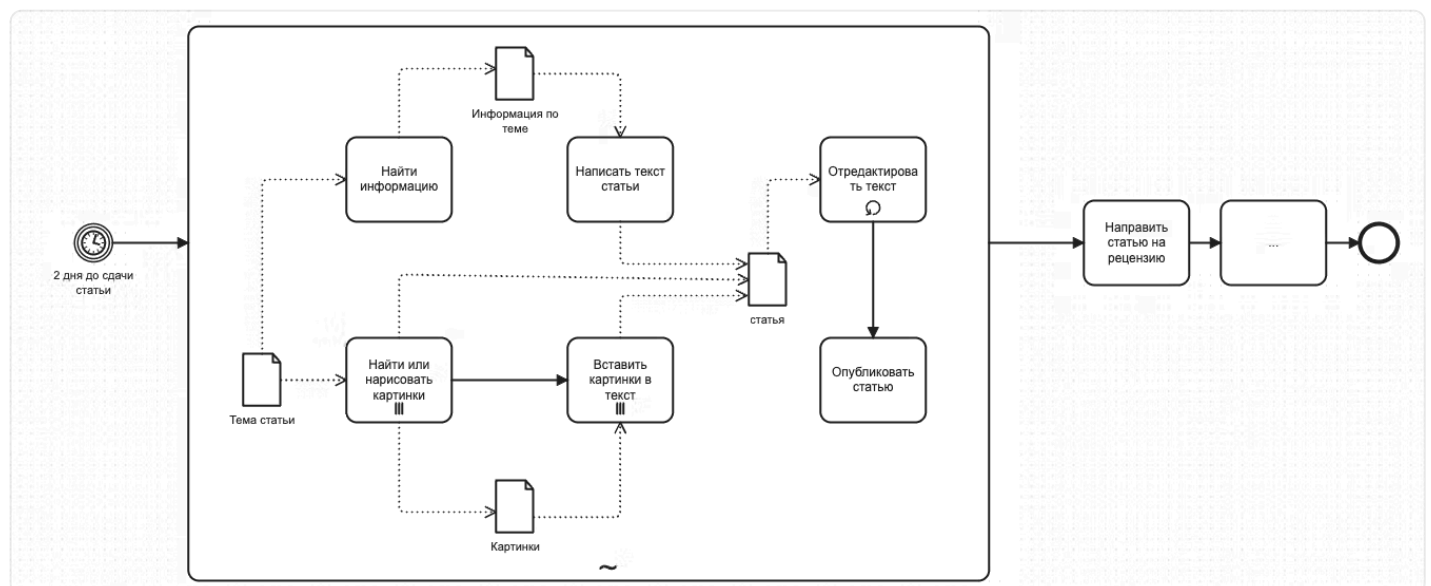
Как правило, действия внутри **спонтанного подпроцесса BPMN** не соединяются. Графически такой процесс обозначается маркером тильды, как показано на рисунке. Этот маркер доступен только для подпроцессов.

Применяется если:

- Нельзя определить строгую последовательность шагов (творческие задачи, задачи по которым нет протокола действий, но требуется результат)
- Иногда возникает необходимость смоделировать еще не «устоявшийся» в компании процесс, по которому, например, еще отсутствует регламент и понятная последовательность действий. Известны только задачи. Для этой цели можно использовать **спонтанный (Ad-Hoc) подпроцесс BPMN**.

У спонтанных процессов ввиду отсутствия последовательности действий нет ни начального ни конечного события, а также управляющих событий. На диаграммах BPMN спонтанные процессы обозначаются знаком тильды “~”.

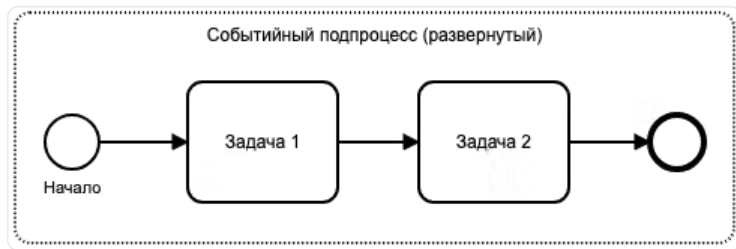
Ниже приведен пример **диаграммы спонтанного (Ad-Hoc) подпроцесса BPMN** с использованием необязательных элементов. Эти элементы накладывают дополнительные ограничения на подпроцесс. Например, задача «Написать текст» может быть выполнена только после получения объекта данных «Найденная информация». Кроме того, задача «Опубликовать статью» будет выполнена только после редактирования ее текста.



Событийный (Event sub-process)

Событийный подпроцесс BPMN – это подпроцесс, который может выполняться один раз или многократно, или не выполняться вовсе.

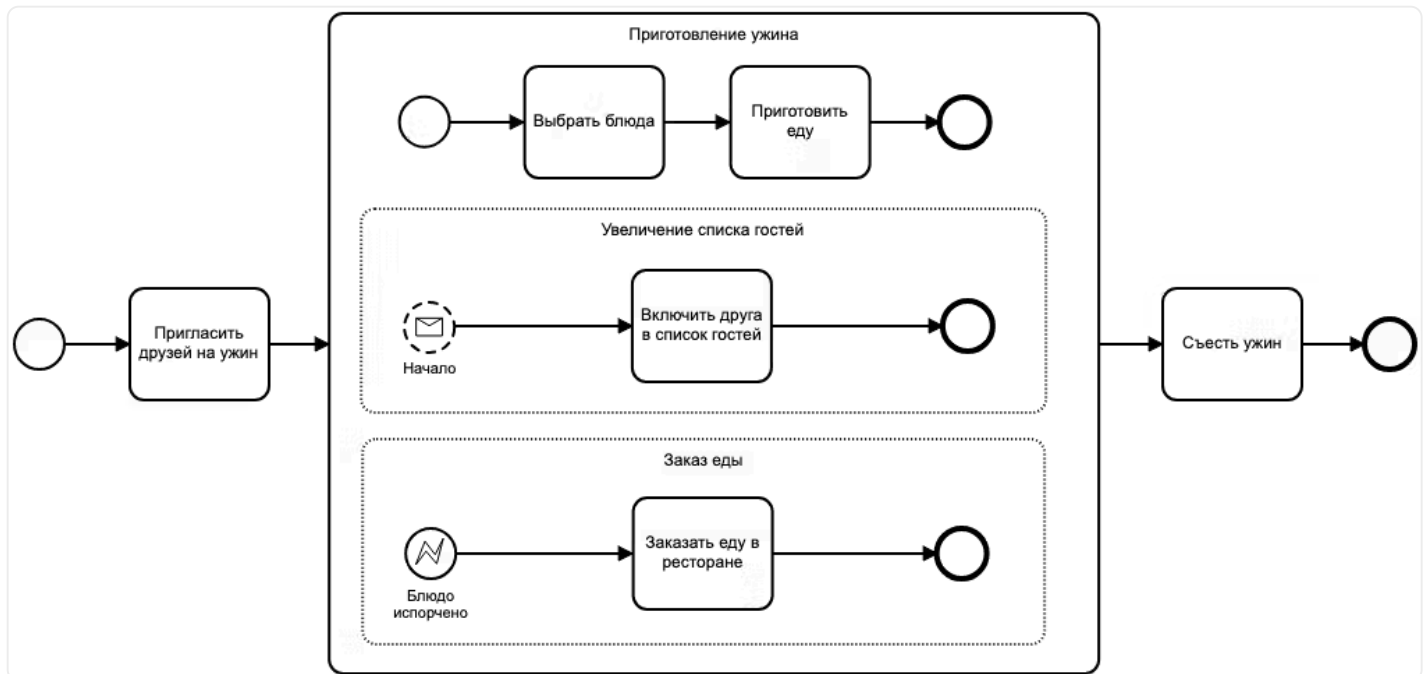
Событийный подпроцесс изображается на диаграмме прямоугольником с закругленными углами и границей, выполненной тонкой пунктирной линией.



Событийный подпроцесс BPMN запускается собственным стартовым событием и не имеет входящих и исходящих потоков операций. Это его главное отличие от обычного подпроцесса, который запускается потоком операций родительского процесса. **Событийный подпроцесс BPMN** должен содержать минимум одно стартовое событие.

Событийный подпроцесс BPMN может влиять на родительский процесс следующим образом:

1. прерывать его, если использовано прерывающее стартовое событие. В этом случае родительский процесс будет остановлен, пока не будет выполнен событийный подпроцесс.
2. не прерывать его, если использовано не прерывающее стартовое событие. В этом случае родительский процесс будет продолжать выполняться вместе с событийным подпроцессом.



Процесс-транзакция (Transaction)

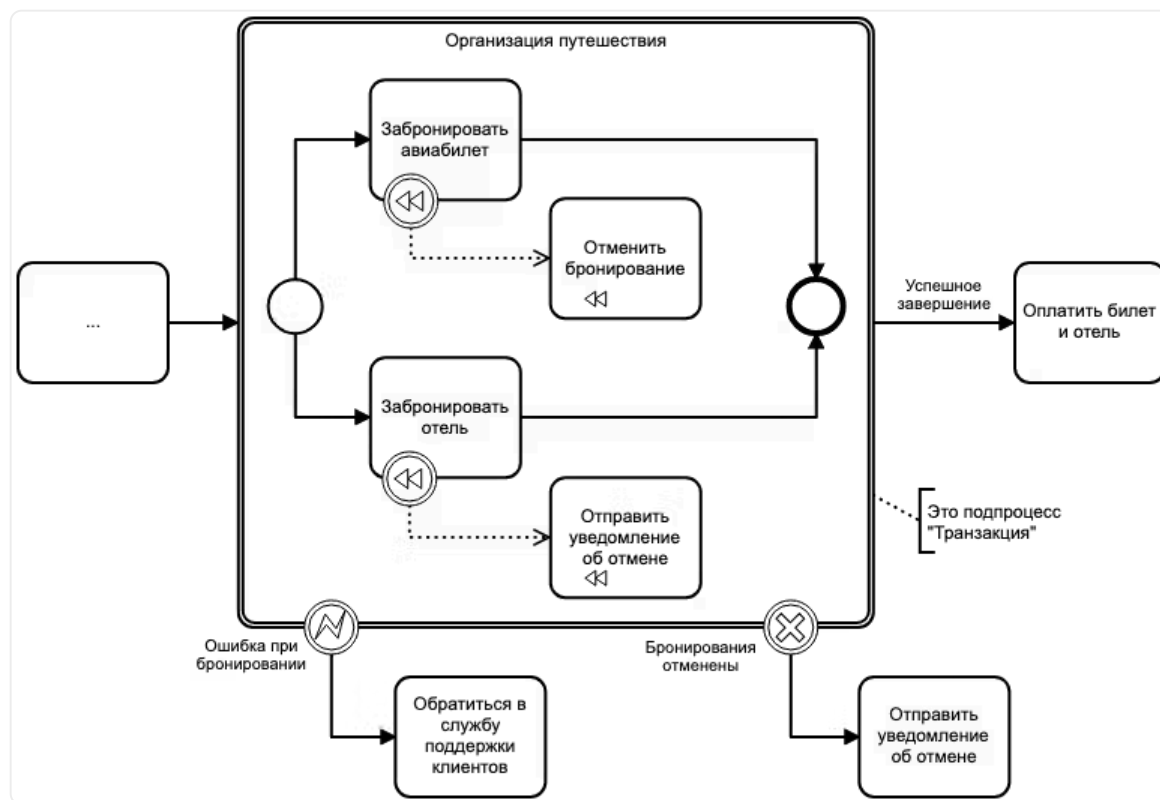
Транзакция BPMN – это подпроцесс, для которого может быть задано несколько вариантов выхода:

1. Успешное завершение – изображается в виде потока операций, выходящего из транзакции.
2. Отмена – изображается с помощью граничного события «Отмена». Это событие может быть использовано только с **подпроцессом «Транзакция» BPMN**. При возникновении события «Отмена» произойдет компенсация («откат назад») определенных действия транзакции, либо возврат к началу процесса.
3. Ошибка – изображается с помощью граничного события «Ошибка». Появление ошибки означает, что **действия Транзакции BPMN** выполняются неверно и становятся невозможными успешное завершение и отмена Транзакции. Выполнение Транзакции при этом завершается без компенсации, а поток операций родительского процесса продолжается от граничного события «Ошибка».

Успешное завершение Транзакции отличается от завершения обычного подпроцесса. Когда все действия Транзакции завершаются конечными событиями (кроме отмены, либо ошибки) моментального возврата к потоку операций

родительского процесса не происходит. Протокол транзакции сначала проверяет состояние всех участников, завершивших Транзакцию. Если обнаруживается, что хотя бы один из них имеет проблемы при завершении действий, то транзакция активирует событие «Ошибки» или «Отмены». При этом поток операций родительского процесса направляется к соответствующему событию Транзакции.

Транзакция изображается на диаграмме в виде прямоугольника с закругленными углами, выполненного двойной тонкой линией:



В качестве примера Транзакции приведен подпроцесс «Организация путешествия». Подпроцесс инициируется из родительского процесса и заключается в параллельном выполнении двух действий: бронирование авиабилета и бронирование отеля. При успешном выполнении обоих действий подпроцесс завершается успешно и родительский процесс переходит к задаче оплаты бронирования.

Если хотя бы одно из действий «Забронировать авиабилет» или «Забронировать отель» выполняется с ошибкой (например, ошибка системы бронирования на сайте), то подпроцесс «Транзакция» последовательно выполняет следующие действия:

- Иницирует граничное прерывающее событие «Ошибка при бронировании». При этом события-компенсации не происходят и задачи-компенсации не выполняются.
- Родительский процесс продолжает выполнение через задачу «Обратиться в службу поддержки клиентов».

Если на любом этапе бронирования потребовалась отмена бронирования (например, клиент пожелал отменить бронирование, то **подпроцесс «Транзакция» BPMN** последовательно выполняет следующие действия:

- Иницирует все граничные события-компенсации и выполняет все задачи-компенсации. Это задачи «Отменить бронирование» и «Отправить уведомление об отмене брони».
- Иницирует граничное прерывающее событие «Бронирования отменены». При этом родительский процесс продолжает выполнение через задачу «Отправить уведомление об отмене».

Сложные события BPMN

Ошибка (Error)

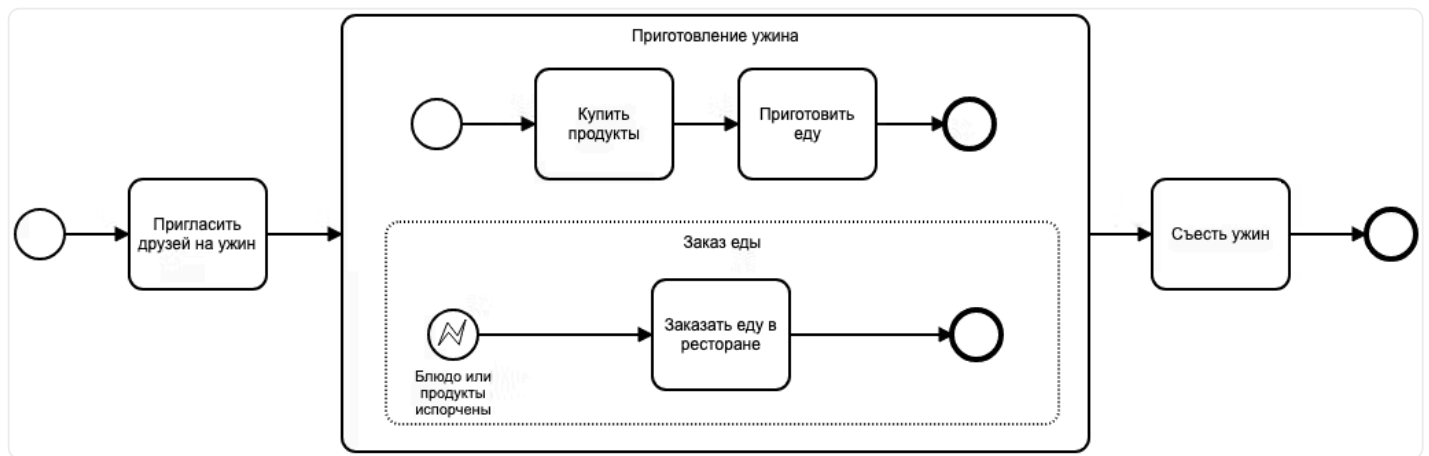
Событие BPMN с типом «Ошибка» используется для моделирования возможных ошибок при выполнении процесса, а также для отображения последовательности действий по устранению этих ошибок.

BPMN не приводит какой-либо классификации возможных ошибок. Бизнес-аналитик сам выбирает какая ошибка может возникнуть в проектируемом процессе. **Событие «Ошибка»** может быть стартовым, промежуточным и конечным.

Правильное использование: Ошибка используется для идентификации и обработки ошибок, с которыми может столкнуться процесс, и принятия мер по их исправлению или обработке.

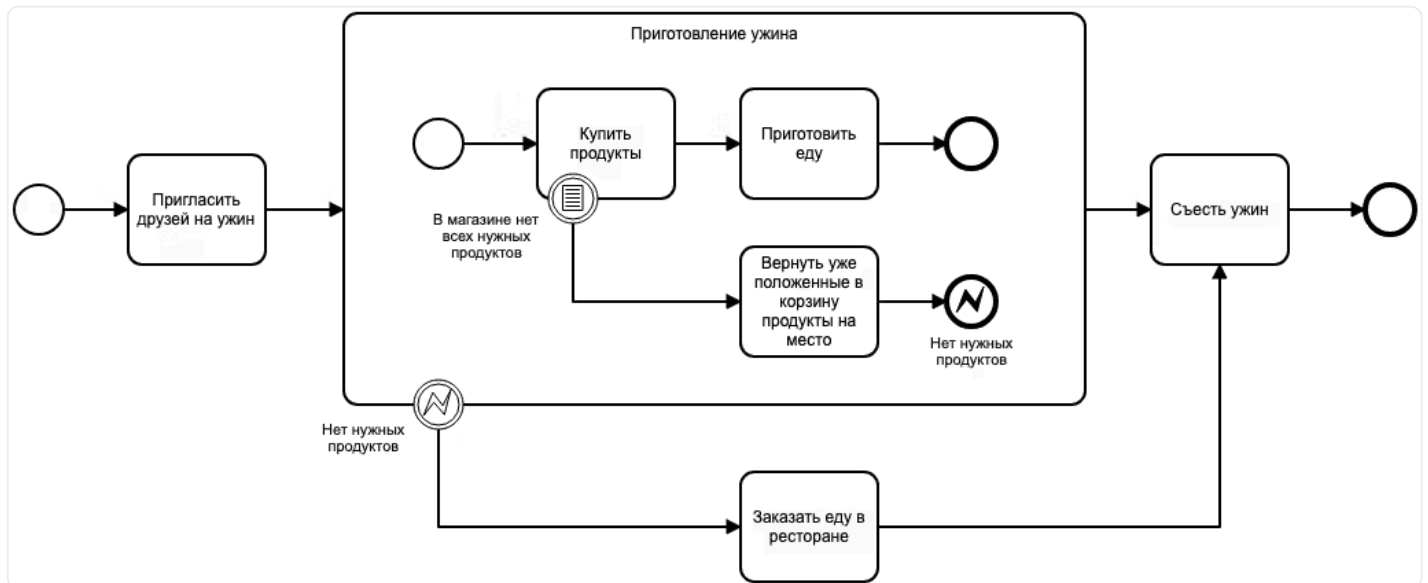
Неправильное использование: Ошибка не должна использоваться для обычных или ожидаемых ситуаций, которые не являются ошибками или исключительными ситуациями.

Стартовое **событие BPMN «Ошибка»** используется только для запуска событийного подпроцесса. Событийный подпроцесс, начинающийся с ошибки, всегда прерывает родительский процесс.



- Промежуточное **событие BPMN «Ошибка»** всегда является граничным-прерывающим. Это означает, что ошибка прерывает выполнение действия, в котором она произошла, и поток операций идет по другому маршруту.
- Конечное **событие BPMN «Ошибка»** показывает, что в результате выполнения процесса произошла ошибка.

Рассмотрим пример. Развернутый подпроцесс на диаграмме ниже показывает, что приготовление ужина состоит из двух задач: «Купить продукты» и «Приготовить еду». Однако, в магазине могут отсутствовать все необходимые продукты. Эта ситуация обрабатывается граничным прерывающим событием с типом «Условие». При срабатывании данного события нужно вернуть все уже положенные в корзину продукты на место и перейти к завершающему событию, которое генерирует ошибку «Нет нужных продуктов». Это событие в свою очередь обрабатывается граничным прерывающим событием «Нет нужных продуктов».



Эскалация (Escalation)

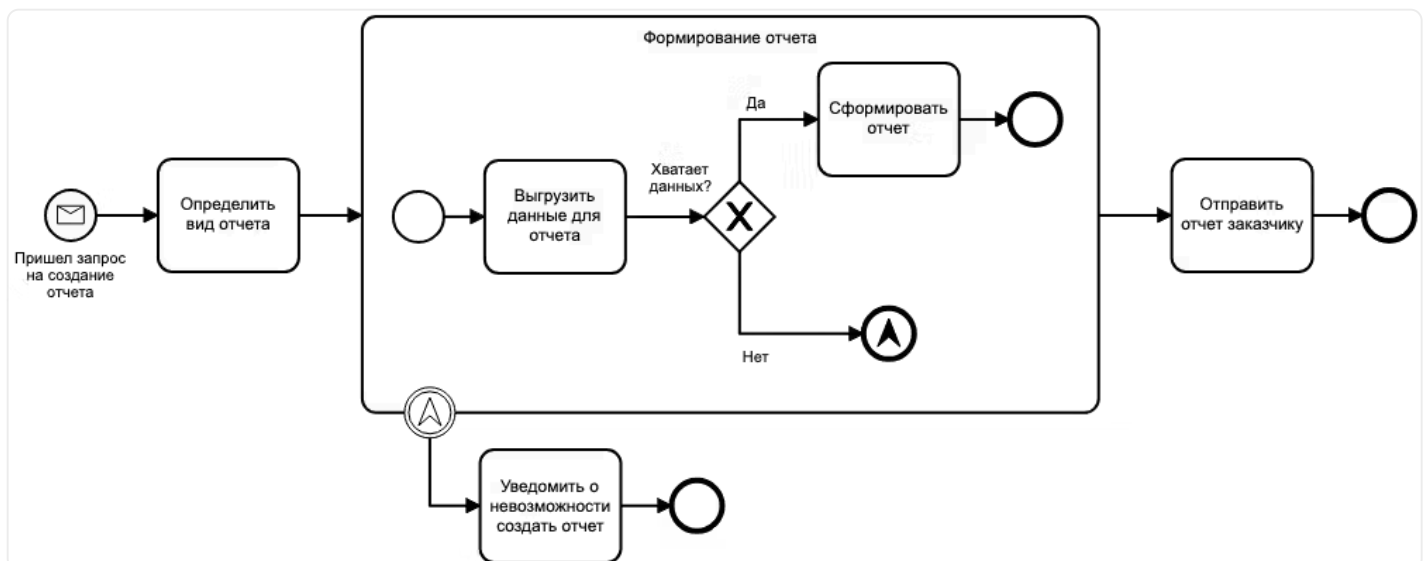
Событие BPMN с типом «Эскалация» указывает на то, что возникла ситуация, требующая вмешательства на более высоком уровне.

Эскалация используется для передачи информации или сигнала верхнему уровню или специалистам, когда процесс не может быть выполнен на текущем уровне или требуется принятие решения на более высоком уровне.

Правильное использование: Эскалация применяется, когда возникают проблемы или ситуации, которые не могут быть решены на текущем уровне и требуют вмешательства на более высоком уровне управления.

Неправильное использование: Эскалация не должна использоваться для передачи информации, которая может быть обработана и решена на текущем уровне.

Рассмотрим пример. На диаграмме показан развернутый подпроцесс «Формирование отчета». В этом подпроцессе, после выгрузки данных выполняется проверка, хватает ли их для построения отчета? Если да, то отчет сначала формируется, а потом отправляется заказчику. Если нет – то срабатывает завершающее **событие BPMN с типом «Эскалация»**. Оно обрабатывается соответствующим граничным событием, и затем заказчик отчета уведомляется о невозможности создать отчет и процесс завершается.

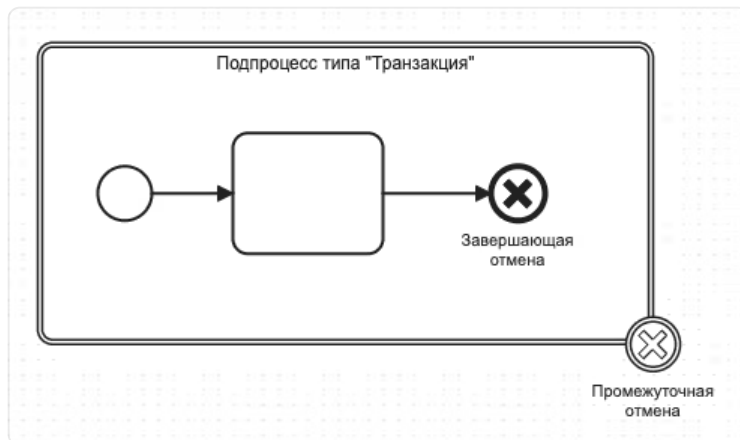


Отличия от события Ошибка:

- Во-первых, событие-обработчик с типом «Эскалация» всегда должно находиться на родительском уровне относительно связанного события-инициатора. Таким образом, при наступлении инициирующего события-эскалации управление всегда передается родительскому процессу.
- Во-вторых, наступление события-эскалации, в отличие от события-ошибки, может приводить к положительному результату, в то время как событие «Ошибка» подразумевает, что в процессе произошла нештатная ситуация.

Отмена (Cancel)

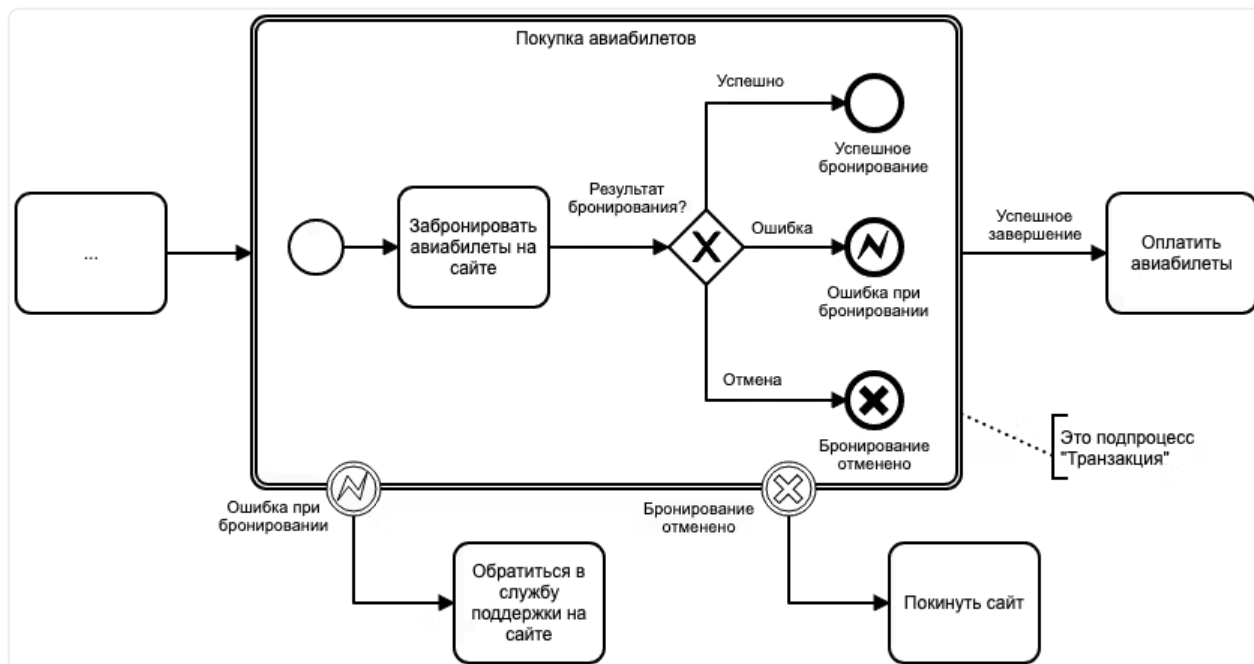
Событие BPMN с типом «Отмена» используется только в подпроцессах "Транзакция" для инициирования и обработки отмены транзакций. Графически **событие BPMN «Отмена»** отображается в виде круга с триггером косого креста внутри. Ниже приведены все возможные виды событий с типом «Отмена».



Правильное использование: Отмена используется, когда процесс или задача не может быть выполнены или не имеют смысла продолжать их выполнение, и требуется принятие мер по отмене или прерыванию.

Неправильное использование: Отмена не должна использоваться в случаях, когда процесс или задача могут быть успешно выполнены без проблем.

Рассмотрим пример. На диаграмме показан подпроцесс-транзакция «Покупка авиабилетов», который выполняется на сайте компании по продаже билетов. Этот подпроцесс может завершаться одним из трех завершающих событий, среди которых есть завершающее **событие BPMN с типом «Отмена»** - «Бронирование отменено». Это завершающее событие обрабатывается соответствующим граничным событием с таким же названием. В случае срабатывания данных событий пользователь покидает сайт по продаже авиабилетов, поскольку необходимость в них пропала.



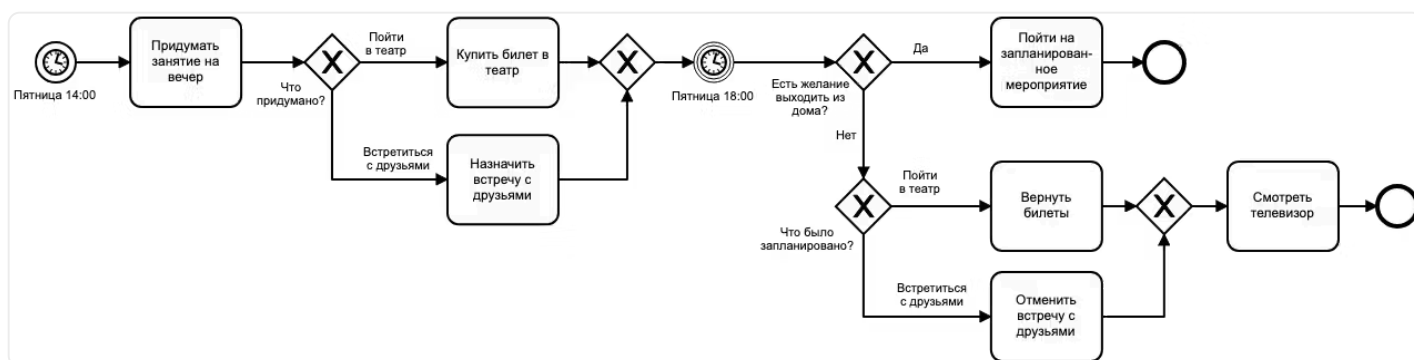
Компенсация (Compensation)

Событие BPMN с типом «Компенсация» используется для обеспечения отката или восстановления состояния после выполнения процесса или задачи, если происходят ошибки или несоответствия.

Правильное использование: Компенсация используется для восстановления состояния или отмены выполненных действий, если процесс или задача не могут быть выполнены успешно или возникла ошибка.

Неправильное использование: Компенсация не должна использоваться для обработки ошибок или исключительных ситуаций, которые могут быть обработаны с помощью других типов событий.

Ниже приведен пример процесса, в котором участник процесса выбирает, что он будет делать в пятницу вечером. Он выбирает один из двух вариантов: пойти в театр или встретится с друзьями. Однако, к вечеру желание куда-либо идти может пропасть. В этом случае нужно сдать билеты в театр обратно, либо отменить встречу с друзьями. После этого можно остаться дома и посмотреть телевизор.



Эта же ситуация может быть смоделирована с использованием **события BPMN «Компенсация»**, как показано ниже:



Сравните это с похожим процессом, в котором используются события с типом «Сообщение». Обратите внимание, что сообщения имеют конкретных отправителя и получателя.

