UML диаграммы

1994 г.

Активное использование анализа и проектирования

Средства UML

язык для определения, представления, проектирования и документирования программных систем, организационно-экономических систем, технических систем и других систем различной природы.

UML содержит стандартный набор диаграмм и нотаций самых разнообразных **видов**:

1) **диаграммы вариантов использования** *(use case diagrams)* – для моделирования бизнес-процессов организации и требований к создаваемой системе);

2) **диаграммы классов** *(class diagrams)* – для моделирования статической структуры классов системы и связей между ними;

3) **диаграммы поведения системы** *(behavior diagrams):*

3.1. диаграммы взаимодействия *(interaction diagrams*):

3.1.1. диаграммы последовательности *(sequence diagrams)*

3.1.2. кооперативные диаграммы *(collaboration diagrams)* – для моделирования процесса обмена сообщениями между объектами;

3.2. диаграммы состояний *(statechart diagrams)* – для моделирования поведения объектов системы при переходе из одного состояния в другое;

3.3. диаграммы деятельностей *(activity diagrams)* – для моделирования поведения системы в рамках различных вариантов использования, или моделирования деятельностей;

4) диаграммы реализации *(implementation diagrams):*

4.1. диаграммы компонентов *(component diagrams)* – для моделирования иерархии компонентов (подсистем) системы;

4.2. диаграммы размещения *(deployment diagrams)* – для моделирования физической архитектуры системы.

Диаграммы вариантов использования

Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом).

Действующее лицо (actor)

– это роль, которую пользователь играет по отношению к системе.

Действующие лица делятся на три основных типа:

- пользователи системы,
- другие системы, взаимодействующие с данной,
- время.

Диаграмма для банкомата (Automated Teller Machine, ATM)

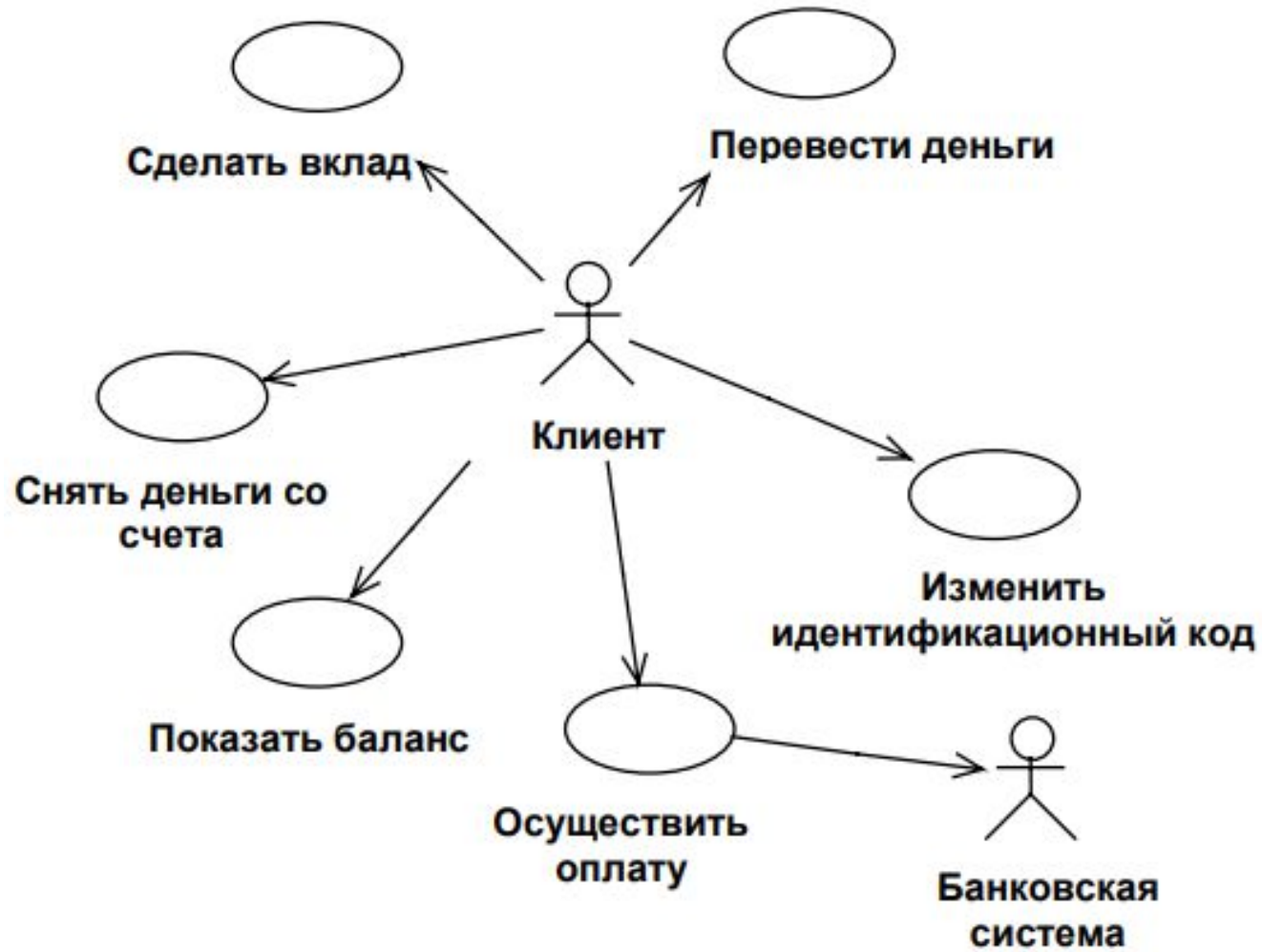


Диаграмма вариантов использования иллюстрирует *требования к системе*.

Все варианты использования связаны с внешними требованиями к функциональности системы. Варианты использования всегда следует анализировать вместе с действующими лицами системы, определяя при этом реальные задачи пользователей и рассматривая альтернативные способы решения этих задач

Конкретная цель диаграмм вариантов использования – это документирование вариантов использования (всё, входящее в сферу применения системы), действующих лиц (всё вне этой сферы) и связей между ними.

Правила диаграммы вариантов использования:

1) *Не моделируйте связи между действующими лицами.* По определению действующие лица находятся вне сферы действия системы. Это означает, что связи между ними также не относятся к её компетенции
2) *Не соединяйте сплошной стрелкой (коммуникационной связью) два варианта использования непосредственно.* Диаграммы данного типа описывают только, какие варианты использования доступны системе, а не порядок их выполнения. Для отображения порядка выполнения вариантов использования применяют диаграммы деятельности.
3) *Вариант использования должен быть инициирован действующим лицом.* Это означает, что должна быть сплошная стрелка, начинающаяся на действующем лице и заканчивающаяся на варианте использования.

Хорошим источником для идентификации вариантов использования служат **внешние события**. Следует начать с перечисления всех событий, происходящих во внешнем мире, на которые система должна каким-то образом реагировать

Варианты использования начинают описывать, что должна будет делать система.

«поток событий» (flow of events)

Целью потока событий является документирование процесса обработки данных, реализуемого в рамках варианта использования

Цель – описать, что будет делать система, а не как она будет делать это. Обычно поток событий включает: – краткое описание; – предусловия (pre-conditions); – основной поток событий; – альтернативный поток событий (или несколько альтернативных потоков); – постусловия (post-conditions).

Описание

Каждый вариант использования должен иметь связанное с ним короткое описание того, что он будет делать. Например, вариант использования *«Перевести деньги»* системы ATM может содержать следующее описание:

Вариант Использования «Перевести деньги» позволяет клиенту или служащему банка переводить деньги с одного счета до востребования или сберегательного счета на другой.

Предусловия

Предусловия варианта использования – это такие условия, которые должны быть выполнены, прежде чем вариант использования начнет выполняться сам.

Например, таким условием может быть выполнение другого варианта использования или наличие у пользователя прав доступа, требуемых для запуска этого. Не у всех вариантов использования бывают предварительные условия.

Основной и альтернативный потоки событий

включают следующее описание:

– способ запуска варианта использования;

– различные пути выполнения варианта использования;

– нормальный, или основной, поток событий варианта использования;

– отклонения от основного потока событий (так называемые альтернативные потоки);

– потоки ошибок;

– способ завершения варианта использования.

(ПРИМЕР С АТМ)

Постусловия

Это такие условия, которые всегда должны быть выполнены после завершения варианта использования.

Как и для предусловий, с помощью постусловий можно вводить информацию о порядке выполнения вариантов использования системы.

Если, например, после одного из вариантов использования должен всегда выполняться другой, это можно описать как постусловие. Такие условия имеются не у каждого варианта использования.

Связи между вариантами использования и действующими лицами

Виды:

- связи коммуникации (communication),
- включения (include),
- расширения (extend) и
- обобщения (generalization).

Связь коммуникации

– это связь между вариантом использования и действующим лицом. На языке UML связи коммуникации показывают с помощью однонаправленной ассоциации (сплошной линии со стрелкой).

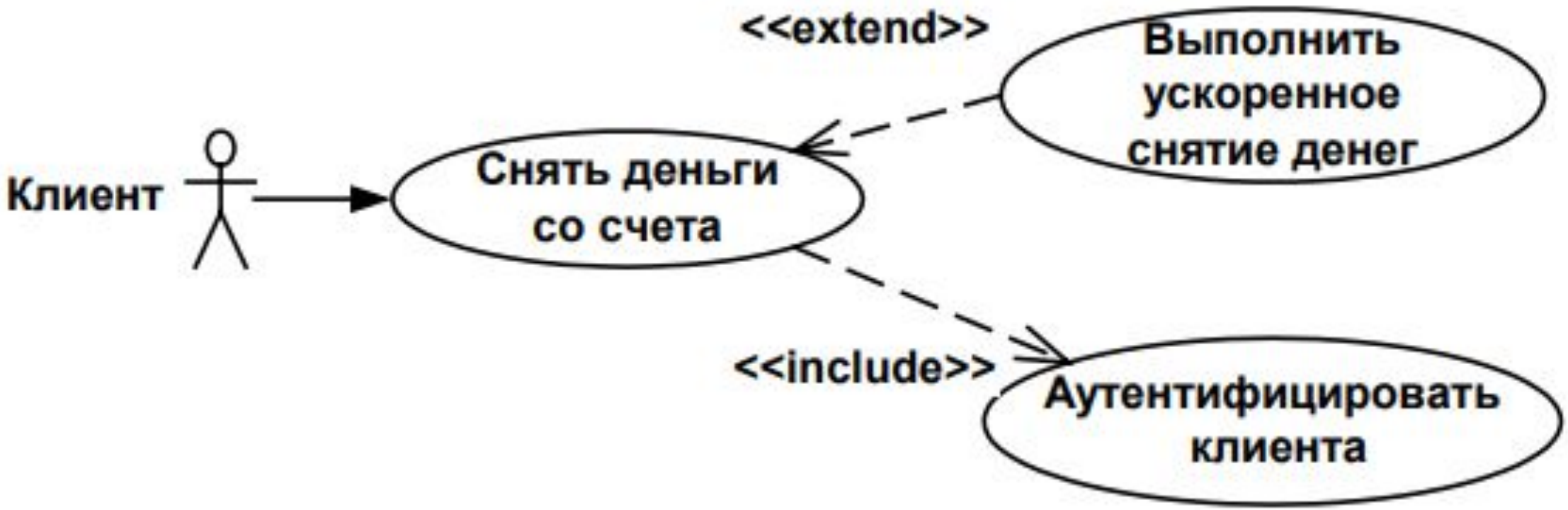
Связь включения

применяется в тех ситуациях, когда имеется какой-либо ***фрагмент поведения системы***, который повторяется более чем в одном варианте использования. С помощью таких связей обычно моделируют многократно используемую функциональность.

В примере АТМ варианты использования «Снять деньги» и «Положить деньги на счет» должны опознать (аутентифицировать) клиента и его идентификационный номер перед тем, как допустить осуществление самой транзакции.

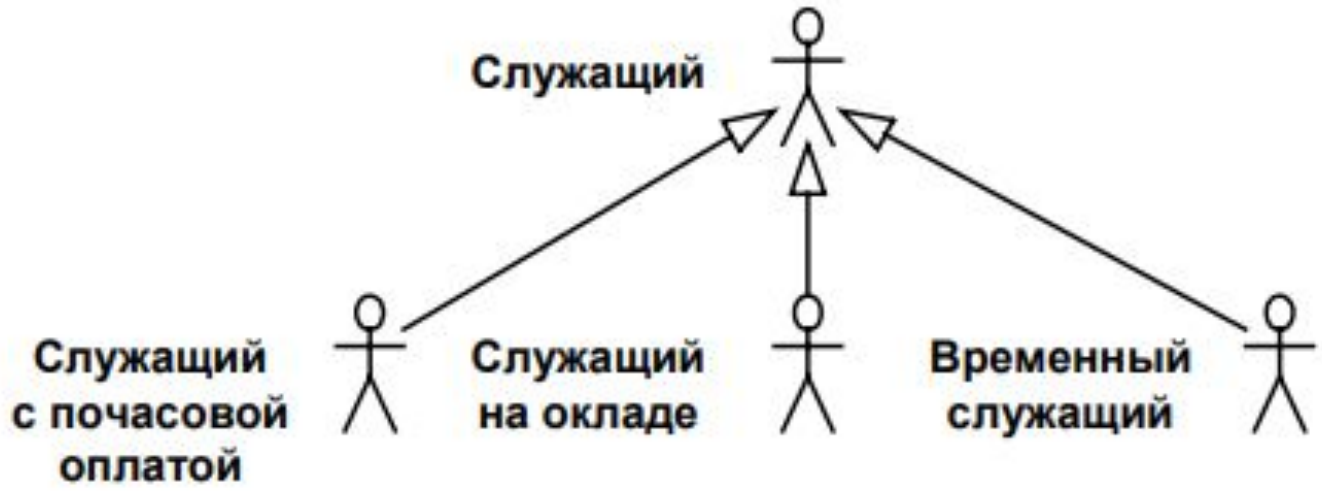
Связь расширения

применяется при описании изменений в нормальном поведении системы. Она позволяет варианту использования только при необходимости использовать функциональные возможности другого.


Клиент
Снять деньги
со счета
<<extend>>
Выполнить
ускоренное
снятие денег
<<include>>
Аутентифицировать
клиента

Связь обобщения

показывает, что у нескольких действующих лиц имеются общие черты. Например, клиенты могут быть двух типов: корпоративные и и[illegible]



Варианты использования являются необходимым средством на стадии формирования требований к ПО. Каждый вариант использования – это потенциальное требование к системе, и пока оно не выявлено, невозможно запланировать его реализацию.

Диаграмма взаимодействия

Диаграммы взаимодействия (interaction diagrams) описывают поведение взаимодействующих групп объектов.

Диаграмма взаимодействия охватывает поведение объектов в рамках только одного варианта использования. На такой диаграмме отображается ряд объектов и те сообщения, которыми они обмениваются между собой.

Сообщение (message) – это средство, с помощью которого объект-отправитель запрашивает у объекта получателя выполнение одной из его операций.

Информационное (informative) сообщение – это сообщение, снабжающее объект-получатель некоторой информацией для обновления его состояния.

Сообщение-запрос (interrogative) – это сообщение, запрашивающее выдачу некоторой информации об объекте-получателе.

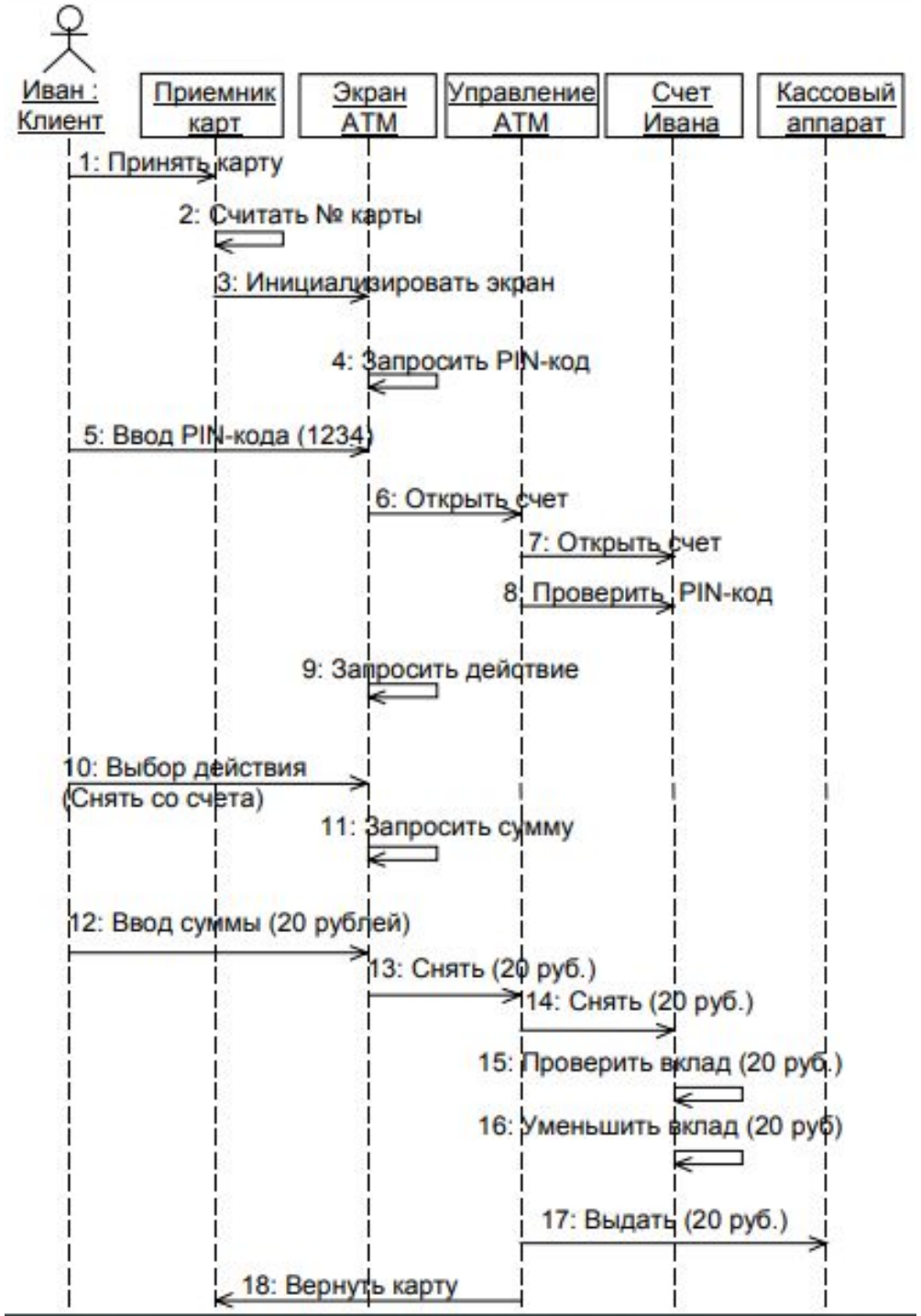
Императивное (imperative) сообщение – это сообщение, запрашивающее у объекта-получателя выполнение некоторых действий.

Диаграмма последовательности

Диаграммы взаимодействия

Диаграммы последовательности отражают поток событий, происходящих в рамках варианта использования.

Диаграммы последовательност и отражают поток событий, происходящих в рамках варианта использования.

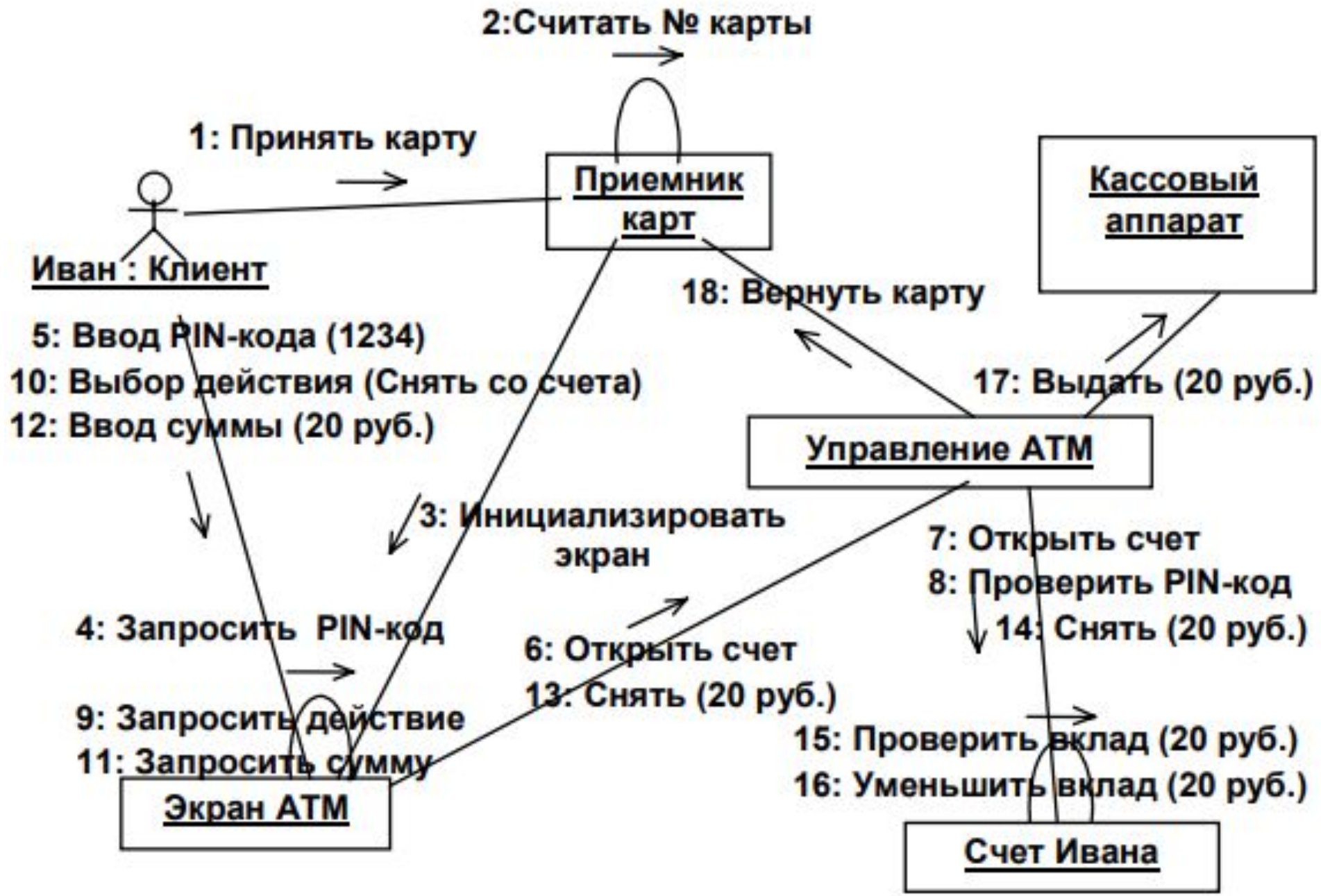


Коорпоративные диаграммы

Диаграммы взаимодействия

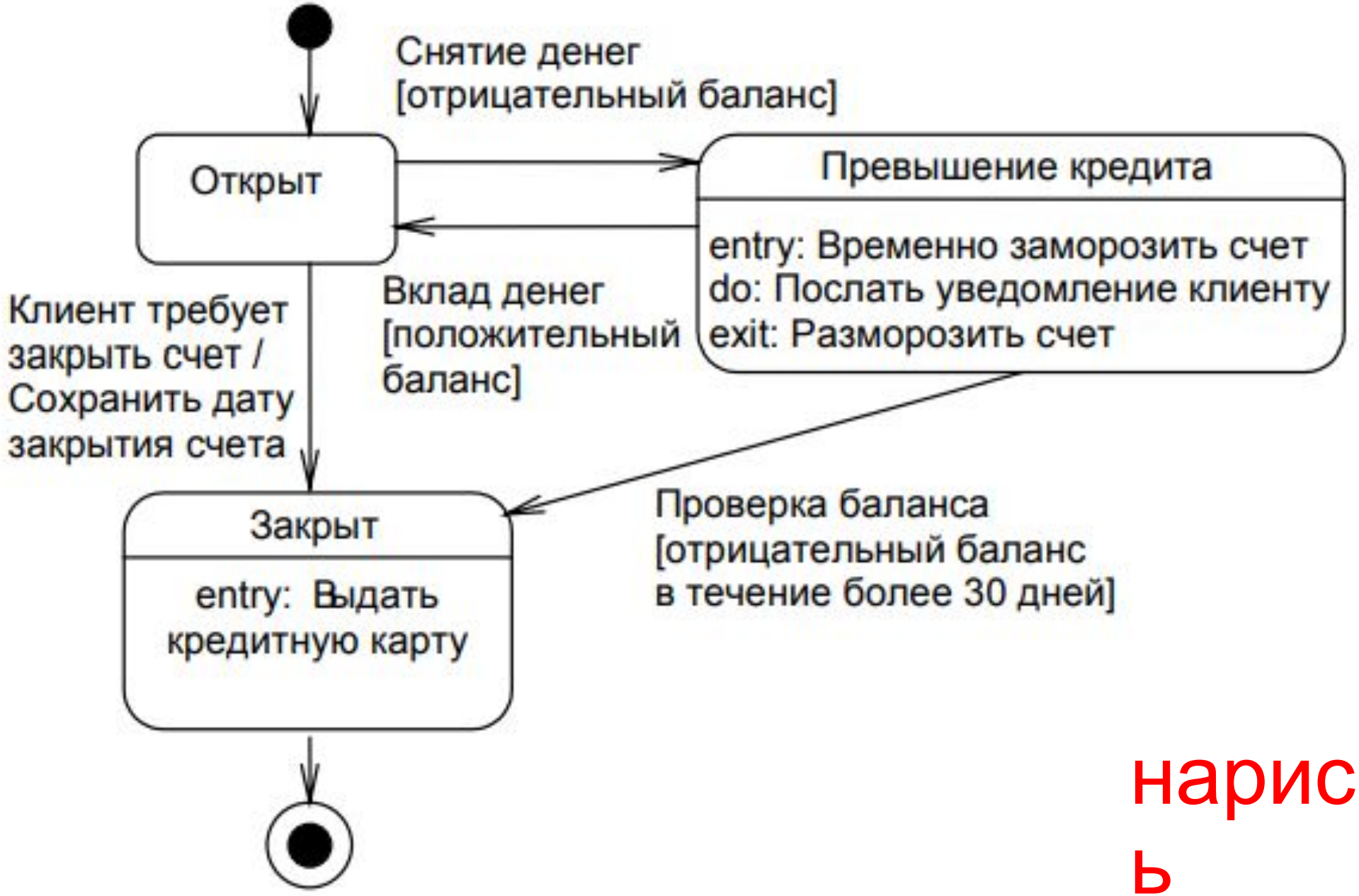
Подобно диаграммам последовательности, кооперативные диаграммы отображают поток событий через конкретный сценарий варианта использования.

Диаграммы последовательности упорядочены *по времени*, а кооперативные диаграммы больше внимания заостряют на связях *между объектами*.


2:Считать № карты
1: Принять карту
Приемник карт
Кассовый аппарат
Иван : Клиент
18: Вернуть карту
5: Ввод PIN-кода (1234)
10: Выбор действия (Снять со счета)
12: Ввод суммы (20 руб.)
17: Выдать (20 руб.)
Управление ATM
3: Инициализировать экран
7: Открыть счет
8: Проверить PIN-код
14: Снять (20 руб.)
4: Запросить PIN-код
6: Открыть счет
13: Снять (20 руб.)
9: Запросить действие
11: Запросить сумму
15: Проверить вклад (20 руб.)
16: Уменьшить вклад (20 руб.)
Экран ATM
Счет Ивана

Диаграммы состояний

Диаграммы состояний определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий.


Снятие денег
[отрицательный баланс]
Открыт
Превышение кредита
entry: Временно заморозить счет
do: Послать уведомление клиенту
exit: Разморозить счет
Вклад денег
[положительный
баланс]
Клиент требует
закрыть счет /
Сохранить дату
закрытия счета
Проверка баланса
[отрицательный баланс
в течение более 30 дней]
Закрыт
entry: Выдать
кредитную карту

нарисовать

На диаграмме имеются два специальных состояния – **начальное (start)** и **конечное (stop)**.

Начальное состояние выделено *черной точкой*, оно соответствует состоянию объекта, когда он только что был создан. Конечное состояние обозначается *черной точкой в белом кружке*, оно соответствует состоянию объекта непосредственно перед его уничтожением.

На диаграмме состояний может быть **одно** и **только одно** начальное состояние.

С состоянием можно связывать данные пяти типов:

- деятельность,
- входное действие,
- выходное действие,
- событие и
- история состояния.

Деятельностью (activity) называется поведение, реализуемое объектом, пока он находится в данном состоянии.

Деятельность – это прерываемое поведение.

Деятельность изображают внутри самого состояния, ей должно предшествовать слово do (делать) и двоеточие.

Входным действием (entry action) называется поведение, которое выполняется, когда объект переходит в данное состояние.

В отличие от деятельности, входное действие рассматривается как непрерываемое. Входное действие также показывают внутри состояния, ему предшествует слово **entry (вход)** и **двоеточие.**

Выходное действие (exit action) подобно входному. Однако, оно осуществляется как составная часть процесса выхода из данного состояния.

Как и входное, выходное действие является непрерываемым. Выходное действие изображают внутри состояния, ему предшествует слово *exit (выход) и двоеточие.*

Do: ^Цель.Событие(Аргументы)

Здесь Цель – это объект, получающий событие,

Событие – это посылаемое сообщение, а

Аргументы являются параметрами посылаемого сообщения.

Переходом (Transition) называется перемещение из одного состояния в другое. Совокупность переходов диаграммы показывает, как объект может перемещаться между своими состояниями.

Переходы могут быть *рефлексивными*. Объект может перейти в то же состояние, в котором он в настоящий момент находится. Рефлексивные переходы изображают в виде стрелки, начинающейся и завершающейся на одном и том же состоянии.

Событие (event) – это то, что вызывает переход из одного состояния в другое. Событие размещают на диаграмме вдоль линии перехода.

Большинство переходов должны иметь события, так как именно они заставляют переход осуществиться.

НО! Бывают и автоматические переходы, не имеющие событий. При этом объект сам перемещается из одного состояния в другое со скоростью, позволяющей осуществиться входным действиям, деятельности и выходным действиям.

Ограждающие условия (guard conditions) определяют, когда переход может, а когда не может осуществиться.

Ограждающие условия изображают на диаграмме вдоль линии перехода после имени события, заключая их в квадратные скобки. Ограждающие условия задавать необязательно.

Действием (action) является непрерываемое поведение, осуществляющееся как часть перехода.

Входные и выходные действия показывают внутри состояний, поскольку они определяют, что происходит, когда объект входит или выходит из него. Большую часть действий изображают вдоль линии перехода, так как они не должны осуществляться при входе или выходе из состояния.

Диаграммы состояний не надо создавать для каждого класса, они применяются только в сложных случаях.

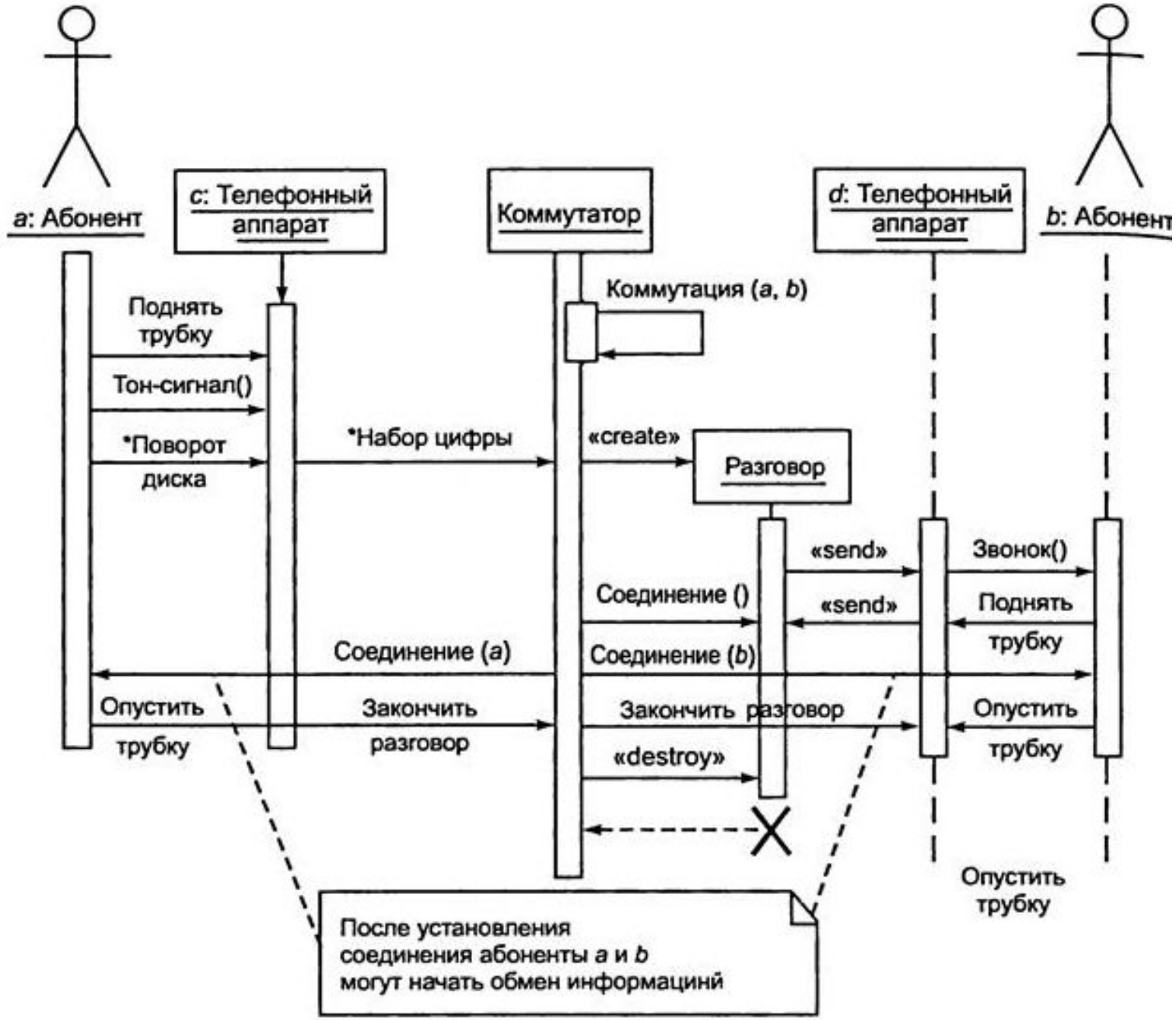
Если объект класса может существовать в нескольких состояниях и в каждом из них ведет себя по-разному, для него может потребоваться такая диаграмма.

Диаграммы деятельности

Самым большим достоинством диаграмм деятельностей является поддержка *параллелизма.* Благодаря этому они являются мощным средством моделирования потоков работ и параллельного программирования.

Диаграммы деятельностей предпочтительнее использовать в следующих ситуациях:

- Анализ варианта использования;
- Анализ потоков работ (workflow) в различных вариантах использования.


a: Абонент
c: Телефонный аппарат
Коммутатор
d: Телефонный аппарат
b: Абонент
Коммутация (a, b)
Поднять трубку
Тон-сигнал()
*Поворот диска
*Набор цифры
«create»
Разговор
«send»
Звонок()
Соединение ()
«send»
Поднять трубку
Соединение (a)
Соединение (b)
Опустить трубку
Закончить разговор
Закончить разговор
Опустить трубку
«destroy»
Опустить трубку
После установления соединения абоненты a и b могут начать обмен информациий

Задание

1. Выбрать тему из файла Проекты СПИ Для примера.xlsx
2. По выбранной теме на сайте http://draw.io/ сделать 3 из перечисленных видом UML-дианраммы.