

ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

UML 2

1



Диаграмма вариантов использования – исходная **концептуальная** модель системы в процессе её проектирования и разработки

Цели диаграммы вариантов использования:

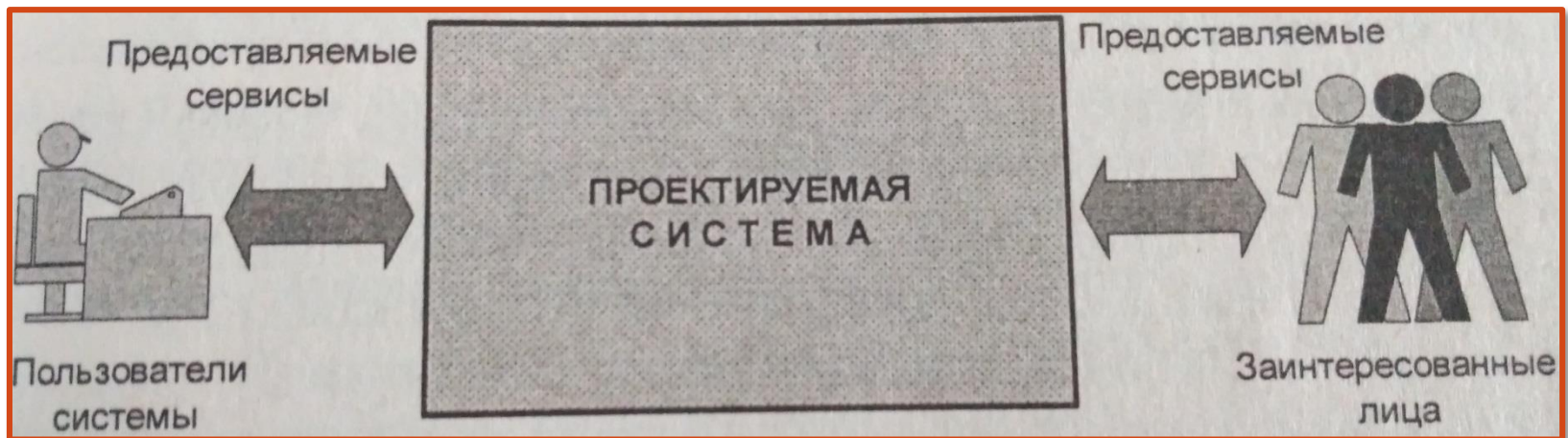
Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы

Сформулировать общие требования к функциональному поведению проектируемой системы

Разработать исходную концептуальную модель системы для её последующей детализации в форме логических и физических моделей

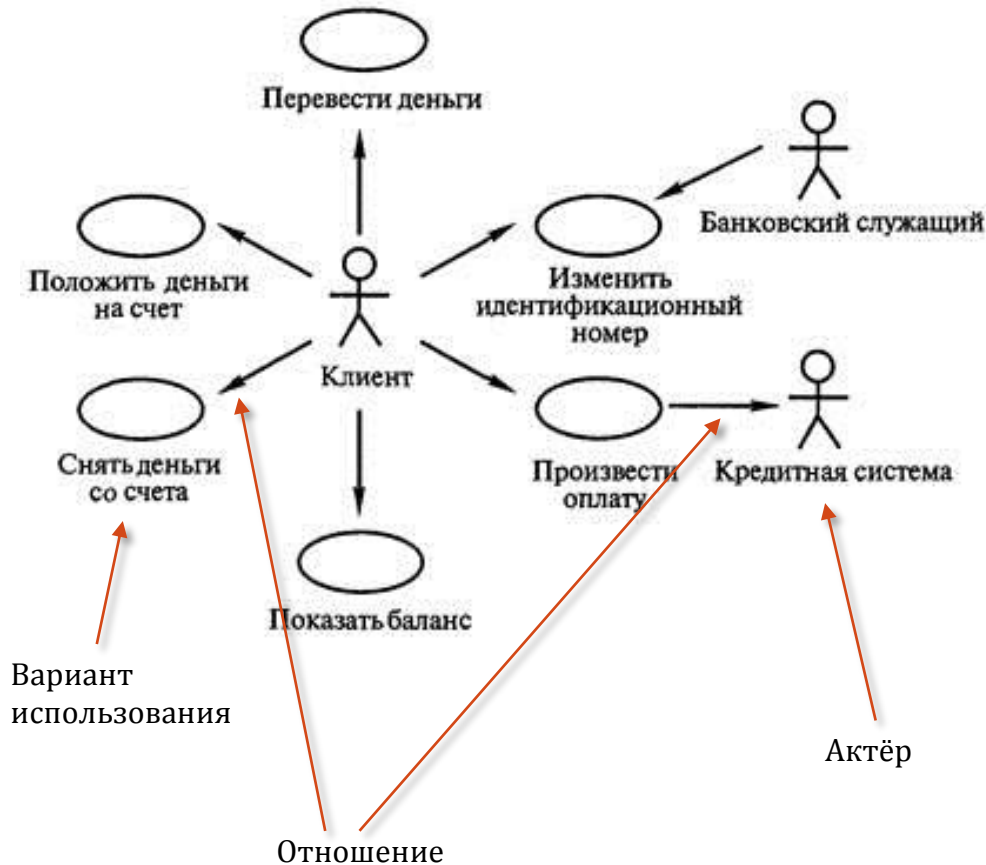
Подготовить исходную документацию для взаимодействия разработчиков системы с её заказчиками и пользователями

- **Функциональность** моделируемой системы представляется в форме **вариантов использования (ВИ)**.
- С ВИ взаимодействуют **актёры** (внешние сущности).
- **Совокупность** всех ВИ, рассматриваемых в контексте поведения проектируемой системы, заключается в **границу** описываемой системы или образует её **субъект**.
- Диаграмму ВИ можно рассматривать как «**чёрный ящик**»:



Проектируемая система и её окружение

Пример диаграммы ВИ



- Диаграмма ВИ (**визуально**) – граф специального вида, содержащий специальные условные изображения ВИ, актёров и отношений между ними.
- Диаграмма ВИ (**формально**) – специализация диаграммы классов, для которой изображаемые классификаторы ограничиваются только актёрами и ВИ.
- Все ВИ на отдельной диаграмме заключаются в прямоугольник, который обозначает границу проектируемой системы (**субъект**).

- В UML2 диаграмма ВИ - модель поведения, которым могут обладать различные элементы проектируемой системы.
- **Субъектом (subject)** в контексте UML2 называется любой элемент модели, который обладает функциональным поведением.
- Субъект вариантов использования может быть некоторой **физической системой** или любым другим элементом с поведением, таким как подсистема, класс или компонент.
- Субъект ≠ Актёр!
- Каждый субъект имеет имя, которое должно соответствовать имени элемента моделируемой системы, владеющего данным поведением. Изображение субъекта на модели **не** является **обязательным**.
- При моделировании сложной системы её можно декомпозировать на несколько подсистем, чтобы упростить процесс проектирования; следовательно, диаграмм ВИ может быть больше одной.
- В UML2 предполагается, что субъект или его части реализуют все ВИ, которые применяются в контексте этого субъекта. Однако необязательно, чтобы все ВИ относились к некоторому субъекту. Субъект может, но не обязан владеть ВИ, которые содержатся в нём.



ВАРИАНТ ИСПОЛЬЗОВАНИЯ



- Вариант использования (**use case**, ВИ, прецедент, функция, юзкейс) – спецификация общих особенностей поведения или функционирования моделируемой системы без рассмотрения внутренней структуры этой системы
- ВИ может сопровождаться дополнительным текстом, раскрывающим смысл выполняемых действий (текст-сценарий или сценарий).
- ВИ обозначается на диаграмме эллипсом.
- Имя ВИ задаётся в форме существительного или глагола с пояснительными словами.
- Текст должен начинаться с заглавной буквы.

Графическое обозначение ВИ (пример)

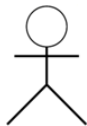


- Цель ВИ – зафиксировать некоторый аспект или фрагмент поведения проектируемой системы без указания особенностей реализации данной функциональности.
- Диаграмма ВИ должна содержать конечное множество ВИ, которые в целом определяют все возможные состояния ожидаемого поведения системы.

9 АКТЁР



- Актёр (actor) - любая внешняя по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует её функциональные возможности для достижения определённых целей или решения частных задач.
- Актёры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой.
- Стандартным графическим обозначением актёра на диаграммах является фигурка «человечка», под которой записывается конкретное имя актёра:



Клиент банка

- Имена актёров должны начинаться с заглавной буквы.
- Имя актёра должно быть достаточно информативным (кассир, менеджер, клиент, сотовый телефон и т.д.).
- Не рекомендуется давать актёрам имена собственные.
- Актёры используются для моделирования внешних по отношению к проектируемой системе сущностей, которые взаимодействуют с ней.
- Актёрами могут быть другие системы, подсистемы проектируемой системы или её отдельные классы.
- Внутренняя структура актёра не определяется.

ВЫЯВЛЕНИЕ АКТЁРОВ

- ⊙ Какие организации или лица будут использовать проектируемую систему?
- ⊙ Кто будет получать пользу от применения системы?
- ⊙ Кто будет использовать информацию от системы?
- ⊙ Будет ли система использовать внешние ресурсы?
- ⊙ Может ли один пользователь играть несколько ролей при взаимодействии с системой?
- ⊙ Могут ли различные пользователи играть одну роль при взаимодействии с системой?
- ⊙ Будет ли система взаимодействовать с законодательными, исполнительными, налоговыми или другими органами?



КОММЕНТАРИЙ



Комментарий (**comment**) в UML2 предназначен для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта.





ОТНОШЕНИЯ



Отношение (**relationship**) в UML2 – произвольная семантическая взаимосвязь между отдельными элементами модели.

Отношение (в общем случае) – абстрактное понятие (метакласс). Ссылается на один или более связанных с ним элементов модели.

Не имеет специальной семантики или общей нотации: различные подклассы отношения имеют собственную семантику и графическую нотацию.

На диаграмме ВИ отношения могут связывать: актёров с ВИ, ВИ с ВИ и актёров с актёрами.

Отношения на диаграмме ВИ:

- ассоциации
- обобщения
- включения
- расширения



ОТНОШЕНИЕ АССОЦИАЦИИ



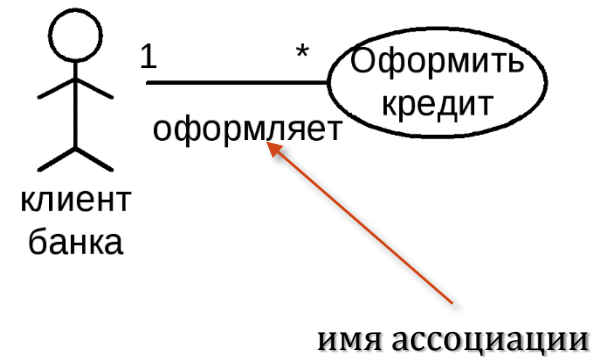
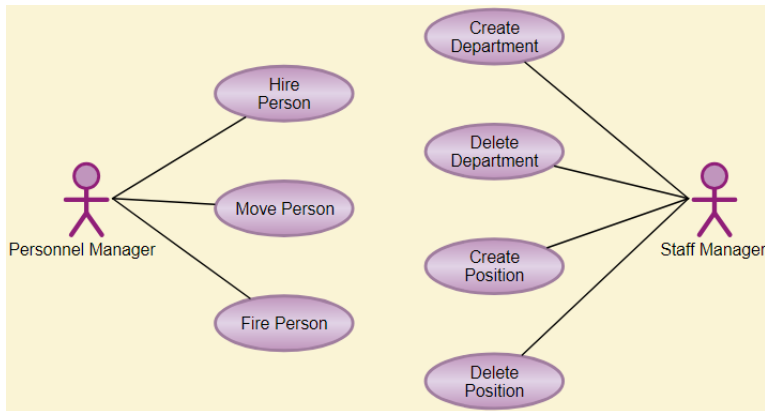
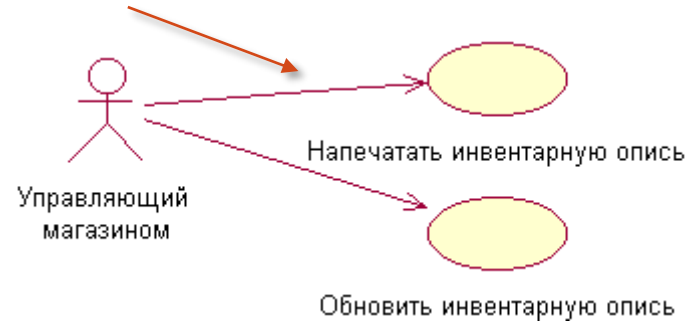
- Ассоциация (**association**) - одно из фундаментальных понятий в UML и в той или иной степени используется при построении всех графических моделей систем в форме канонических диаграмм.
- В данном случае служит для обозначения специфической роли актёра в отдельном ВИ.
- Обозначается сплошной линией (ненаправленная ассоциация) или линией со стрелкой (направленная ассоциация) между актёром и вариантом использования.
- Направленная (от актёра к варианту использования) используется, если необходимо явно указать инициатора взаимодействия (главный или основной актёр).
- Направленная (от варианта использования к актёру) может указывать на то, что актёру предоставляется справочная или отчётная информация.
- Может иметь дополнительные условные обозначения, такие, например, как имя и кратность.

Примеры отношения ассоциации

ненаправленная ассоциация



направленная ассоциация





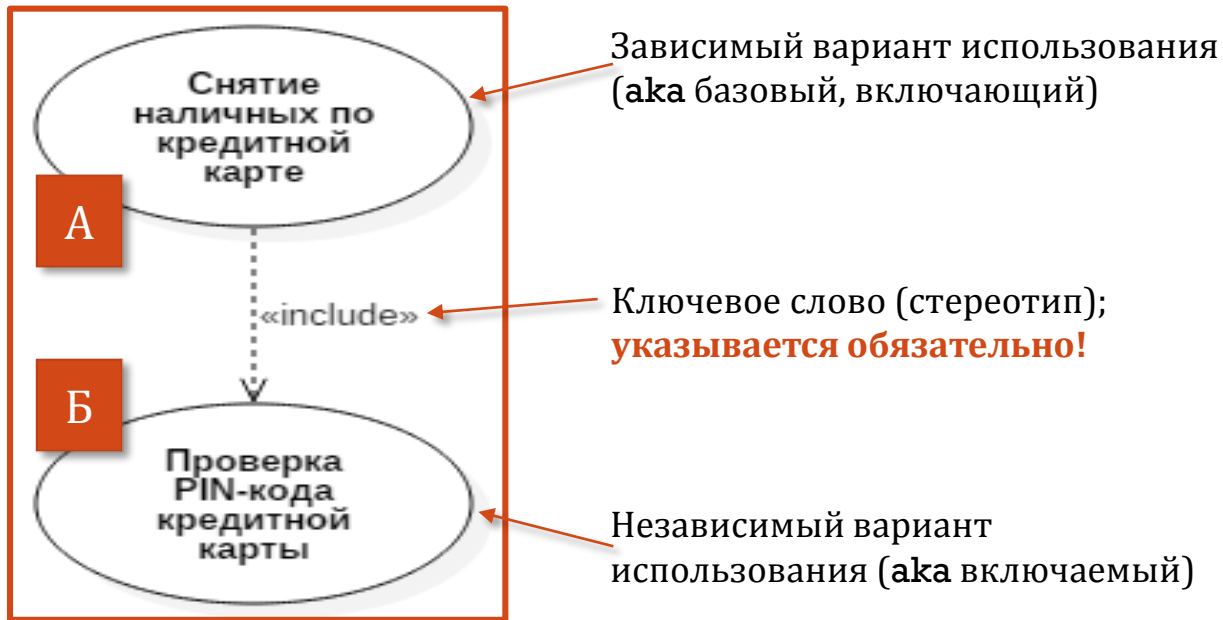
ОТНОШЕНИЕ ВКЛЮЧЕНИЯ



- Отношение включения между двумя вариантами использования в UML2 – частный случай общего отношения зависимости.
- Отношение зависимости (**dependency**) – форма взаимосвязи между двумя элементами модели, предназначенная для спецификации того обстоятельства, что изменение одного элемента модели приводит к изменению некоторого другого элемента.
- Зависимость (в общем случае) – направленное бинарное отношение, связывающее между собой только два элемента модели – независимый и зависимый.
- Отношение включения (**include**) – специфицирует тот факт, что некоторый вариант использования содержит поведение, определённое в другом варианте использования. Устанавливается только между двумя вариантами использования. Обозначается пунктирной линией со стрелкой.

отношение зависимости:
----->





У зависимого варианта использования (ЗВИ) может быть 1+ независимых вариантов использования (НВИ). При НВИ = 1 лучше функционал НВИ включить в ЗВИ, то есть **не выделять** его из ЗВИ в отдельный вариант использования.



ОТНОШЕНИЕ РАСШИРЕНИЯ



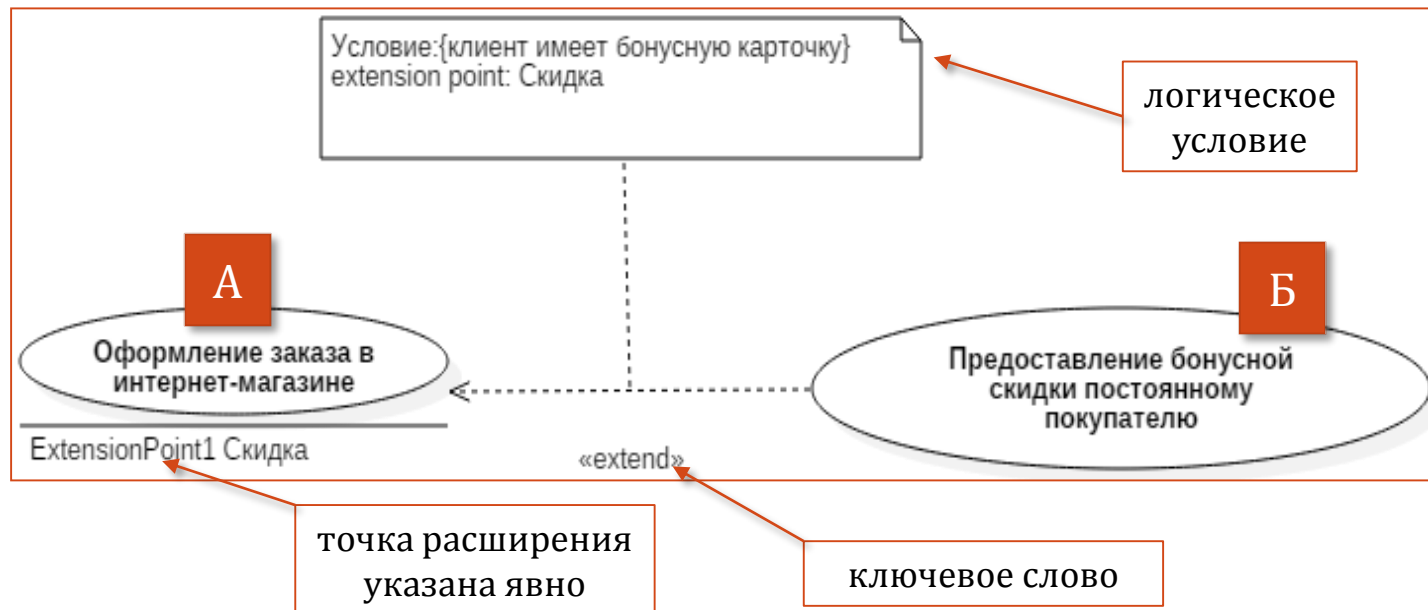
- Отношение расширения между двумя вариантами использования в UML2 – частный случай общего отношения зависимости.
- Отношение расширения (**extend**) определяет взаимосвязь одного варианта использования с некоторым другим вариантом использования, функциональность или поведение которого задействуется первым не всегда, а только при выполнении некоторых дополнительных условий.
- Графически данное отношение обозначается как отношение зависимости в форме пунктирной линии с V-образной стрелкой, направленного от зависимого варианта использования к независимому и соединённой с ним в т.н. точке расширения (**extension point**).



независимый (aka базовый, расширяемый)

зависимый (aka расширяющий)

- Чтобы расширение функциональности ВИ имело место, должно быть выполнено специальное логическое условие. Наличие данного условия всегда предполагает проверку некоторого условия и ссылку на точку расширения в варианте использования.
- Формально точка расширения может иметь некоторое имя, которое должно быть уникальным в базовом варианте использования. Точка расширения является ссылкой на местоположение в варианте использования, в которое должно быть вставлено поведение расширяющего варианта использования.
- Условие можно уточнить текстом в произвольной форме.



- Базовый вариант использования может иметь несколько точек расширения.
- С каждой из них должен быть связан расширяющий вариант использования.
- Один расширяющий вариант использования может быть связан отношениями расширения с несколькими базовыми вариантами использования.
- Поведение любого базового варианта использования не должно зависеть от поведения его расширений.



ОТНОШЕНИЕ ОБОБЩЕНИЯ

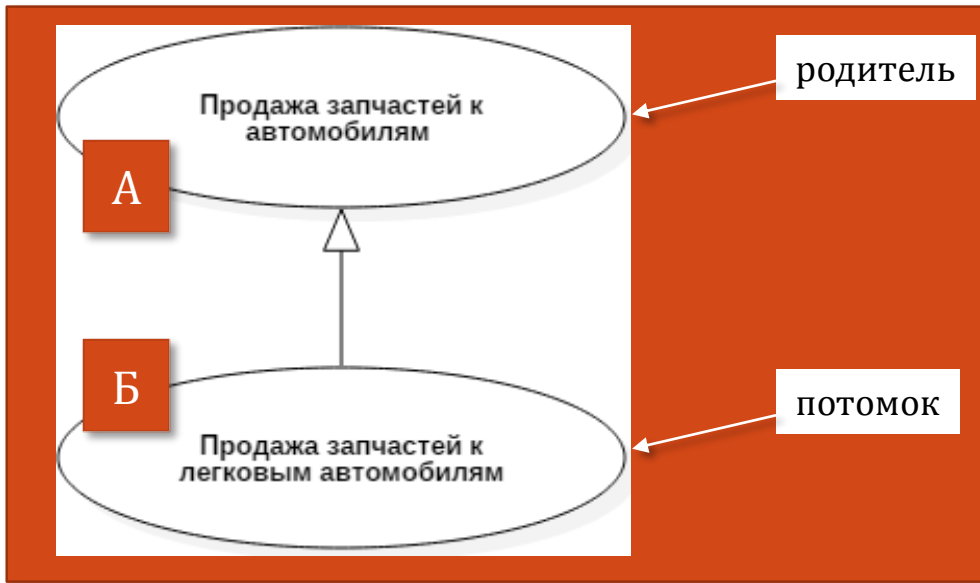


Отношение обобщения (**generalization**) предназначено для спецификации того факта, что один элемент модели является специальным или частным случаем другого элемента модели.

или

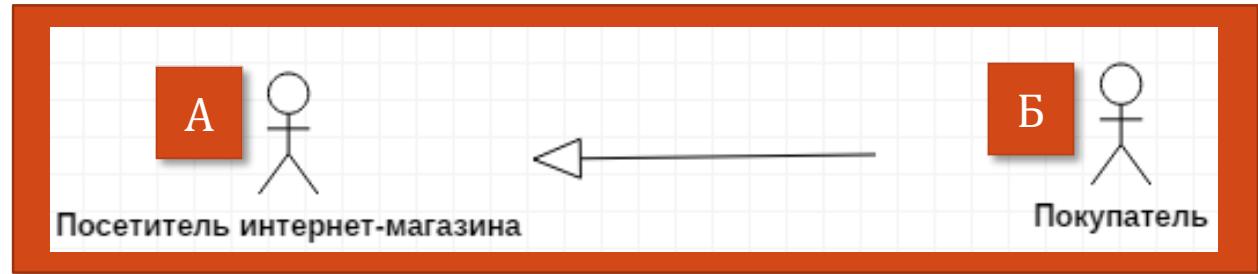
Отношение обобщения служит для указания того факта, что некоторый вариант использования «Б» может быть обобщён до варианта использования «А». Вариант «Б» является специализацией варианта «А». «А» - предок или родитель по отношению к «Б», а вариант «Б» – потомок по отношению к «А».

- Потомок наследует все свойства родителя (+ может обладать дополнительными свойствами).
- Может связывать между собой только элементы **одного** типа!
- Графическое обозначение: сплошная линия с треугольной не закрашенной стрелкой, указывающая на общий элемент модели.



у одного родителя
может быть много
потомков и наоборот

Отношение обобщения может устанавливаться между вариантами использования или актёрами



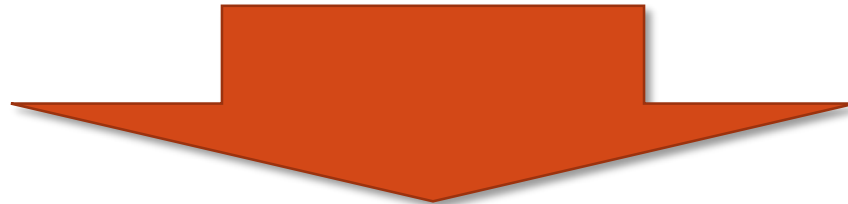


ФОРМАЛИЗАЦИЯ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ К СИСТЕМЕ С ПОМОЩЬЮ ДИАГРАММЫ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



- Варианты использования являются средством для спецификации требований к системе.
- Требуемое поведение системы или субъекта специфицируется одним или несколькими вариантами использования, которые определяются в соответствии с потребностями актёров.
- **Требование (requirement)** – желательное свойство, характеристика или условие, которым должна удовлетворять система в процессе своей эксплуатации.
- Рекомендуется: **один** вариант использования – **одно** требование.

Применительно к программным системам предложена следующая классификация требований - **FURPS+**, что соответствует первым буквам категорий требований на английском языке



функциональные требования

- **Functionality**

требования удобства использования

- **Usability**

требования надёжности

- **Reliability**

требования производительности

- **Performance**

требования удобства сопровождения

- **Supportability**

Символом «+» обозначены дополнительные требования:

проектные ограничения

требования управления системой

требования к GUI

физические требования

юридические требования

Центральное место занимают **функциональные** требования, которые в контексте моделей UML2 должны служить исходной информацией для построения диаграмм вариантов использования

Построение диаграммы вариантов использования является самым **первым** этапом процесса объектно-ориентированного анализа и проектирования, **цель** которого – представить совокупность функциональных требований к поведению проектируемой системы.

Одно из требований UML2 – **самодостаточность** графических диаграмм для представления информации о моделях проектируемых систем.

НО

Изобразительных свойств UML2 недостаточно, чтобы учесть на диаграммах вариантов использования особенности функционального поведения достаточно сложных систем

[мнение большинства разработчиков и экспертов]

поэтому

Рекомендуется **дополнять** этот тип диаграмм **текстовыми сценариями**, которые уточняют или детализируют последовательность действий, совершаемых системой при реализации поведения



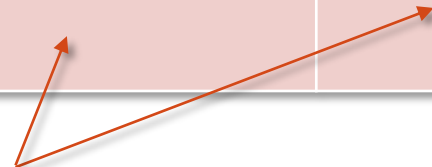
СПЕЦИФИКАЦИЯ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ С ПОМОЩЬЮ ТЕКСТОВЫХ СЦЕНАРИЕВ



- Содержание варианта использования может быть **дополнительно** специфицировано в форме пояснительного текста, который раскрывает смысл выполняемых действий при выполнении варианта использования.
- Такой пояснительный текст применительно к диаграммам вариантов использования получил специальное название – «текст-сценарий» или «сценарий».
- **Сценарий (scenario)** – специально написанный текст, который описывает **поведение** моделируемой системы в форме последовательности выполняемых действий актёров и самой системы.
- Существуют различные шаблоны сценариев. Один из них приведён далее



Главный раздел	Раздел «Типичный ход событий»	Раздел «Исключения»	Раздел «Примечания»
Имя варианта использования	Типичный ход событий, приводящий к успешному выполнению варианта использования	Исключение №1	Примечания
Актёры		Исключение №2	
Цель		Исключение №3	
Краткое описание			
Тип			
Ссылки на другие варианты использования			



Заполняют по необходимости

- Написание сценариев модели начинают с базовых вариантов использованию, после чего рассматриваются сценарии второстепенных или включаемых вариантов использования.
- Сценарии расширяющих вариантов использования обычно помещаются в разделе исключений базового варианта использования.
- Желательно, чтобы каждому **исключению** соответствовал отдельный вариант использования, который соединяется с **базовым** отношением **расширения**. Иначе будет нарушено логическое соответствие между текстами сценариев и диаграммой вариантов использования.

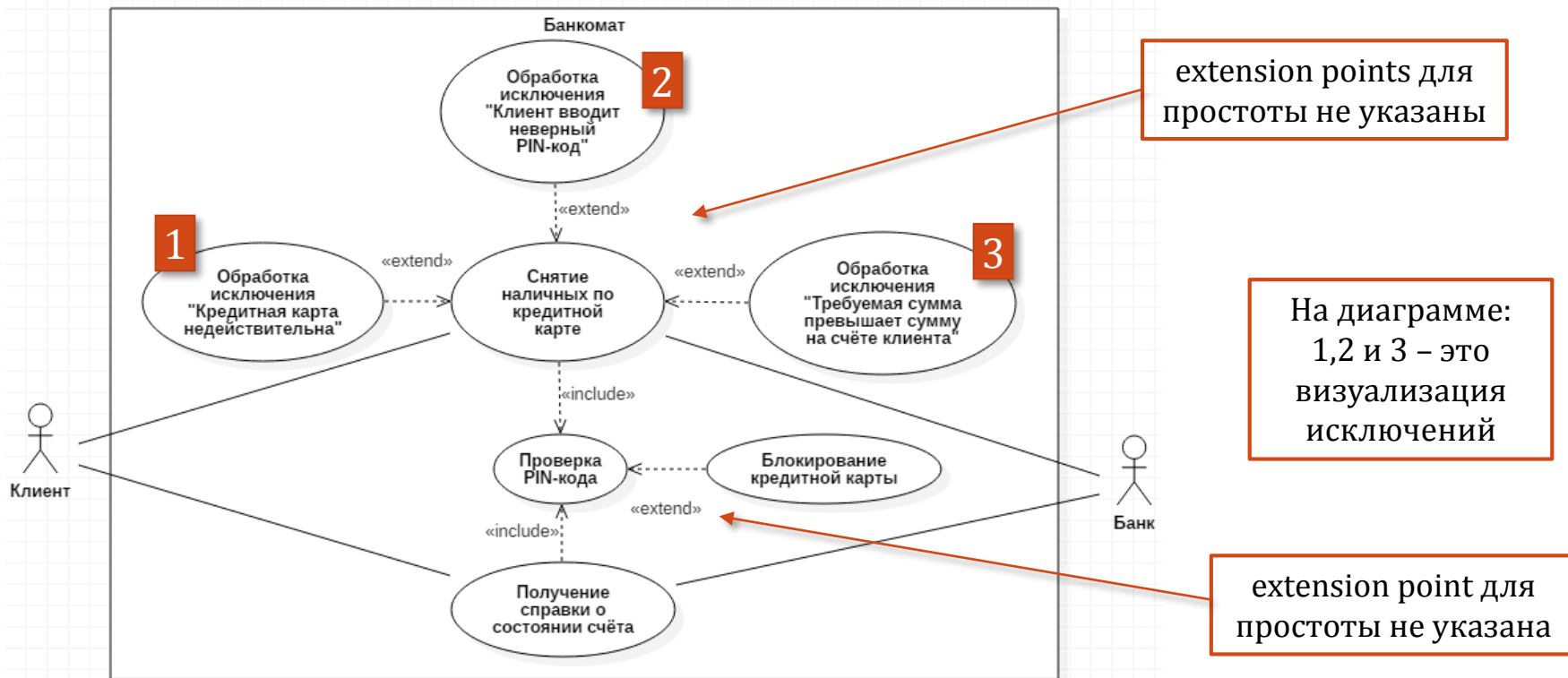
Сценарий должен уточнять и дополнять диаграмму, а не заменять её!

Диаграмма вариантов использования – это, в первом приближении, просто перечень функций системы, а не алгоритм её работы!



ПРИМЕР СЦЕНАРИЯ





Главный раздел

Вариант использования

Снятие наличных по кредитной карте

Актёры

Клиент, Банк

Цель

Получение требуемой суммы наличными

Краткое описание

Клиент запрашивает требуемую сумму. Банкомат обеспечивает доступ к счёту клиента. Банкомат выдаёт клиенту наличные

Тип

Базовый

Ссылки на другие варианты использования

Включает в себя варианты использования:

- проверка PIN-кода кредитной карты

Типичный ход событий

Действия актёров	Отклик системы
1. Клиент вставляет кредитную карту в устройство чтения банкомата.	2. Банкомат проверяет кредитную карту. 3. Банкомат предлагает ввести PIN-код. <u>Исключение №1</u> : Кредитная карточка недействительна
4. Клиент вводит PIN-код.	5. Банкомат проверяет PIN-код. 6. Банкомат отображает опции меню. <u>Исключение №2</u> : Клиент вводит неверный PIN-код
7. Клиент выбирает снятие наличных со своего счёта	8. Система делает запрос в Банк и выясняет текущее состояние счёта клиента. 9. Банкомат предлагает ввести требуемую сумму
10. Клиент вводит требуемую сумму. 11. Банк проверяет введённую сумму. <u>Исключение №3</u> : Требуемая сумма превышает сумму на счёте клиента	12. Банкомат изменяет состояние счёта клиента, выдаёт наличные и чек.

13. Клиент получает наличные и чек.

14. Банкомат предлагает клиенту забрать его кредитную карту.

15. Клиент получает свою кредитную карту.

16. Банкомат отображает сообщение о своей готовности к работе.

Исключения

Исключение №1: Кредитная карта недействительна

3. Банкомат отображает информацию о неверно вставленной кредитной карте.

14. Банкомат возвращает клиенту его кредитную карту.

Исключение №2: Клиент вводит неверный PIN-код

6. Банкомат отображает информацию о неверном PIN-коде

4. Клиент вводит новый PIN-код

Исключение №3: Требуемая сумма превышает сумму на счёте клиента

12. Банкомат отображает информацию о превышении кредита

10. Клиент вводит новую требуемую сумму



РЕКОМЕНДАЦИИ ПО РАЗРАБОТКЕ ДИАГРАММ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



Рекомендуемое количество актёров в модели – не более 20, а вариантов использования – не более 50.

1. Определить главных и второстепенных актёров

2. Определить цели главных актёров по отношению к системе

3. Сформулировать базовые варианты использования

4. Рассмотреть все базовые варианты использования в порядке убывания степени риска их реализации

Рекомендуемое количество актёров в модели – не более 20, а вариантов использования – не более 50.

5. Выделить участников, интересы, предусловия и постусловия выбранного варианта использования

6. Написать успешный сценарий выполнения выбранного варианта использования

7. Определить исключения и написать для них сценарии

8. Отметить на диаграмме отношения

9. Проверить диаграмму на наличие дубликатов



РАСШИРЕНИЕ UML ДЛЯ БИЗНЕС- МОДЕЛИРОВАНИЯ



Бизнес-актёр

- **Business actor**

Сотрудник

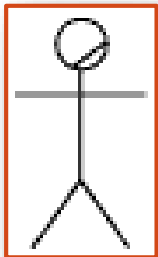
- **Business worker**

Бизнес-вариант использования

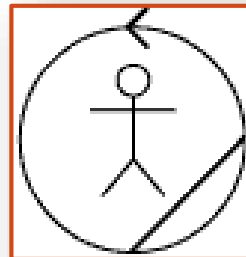
- **Business use case**

РАСШИРЕНИЕ UML ДЛЯ БИЗНЕС-МОДЕЛИРОВАНИЯ

Бизнес-актёр – индивидуум, группа, организация, компания или система, которые взаимодействуют с моделируемой бизнес-системой, но не входят в неё (например, клиенты, покупатели, поставщики, партнёры, ...).



Сотрудник – индивидуум, который действует внутри моделируемой бизнес-системы, взаимодействует с другими сотрудниками и является участником бизнес-процесса.



Бизнес-вариант использования – блок, определяющий функциональность модулируемой системы, ориентированной на выполнение отдельного бизнес-процесса.

