

ЛЕКЦИЯ 39

Тема: Объектно-ориентированная информационная система, моделирование UML диаграмм. Use-case диаграммы.

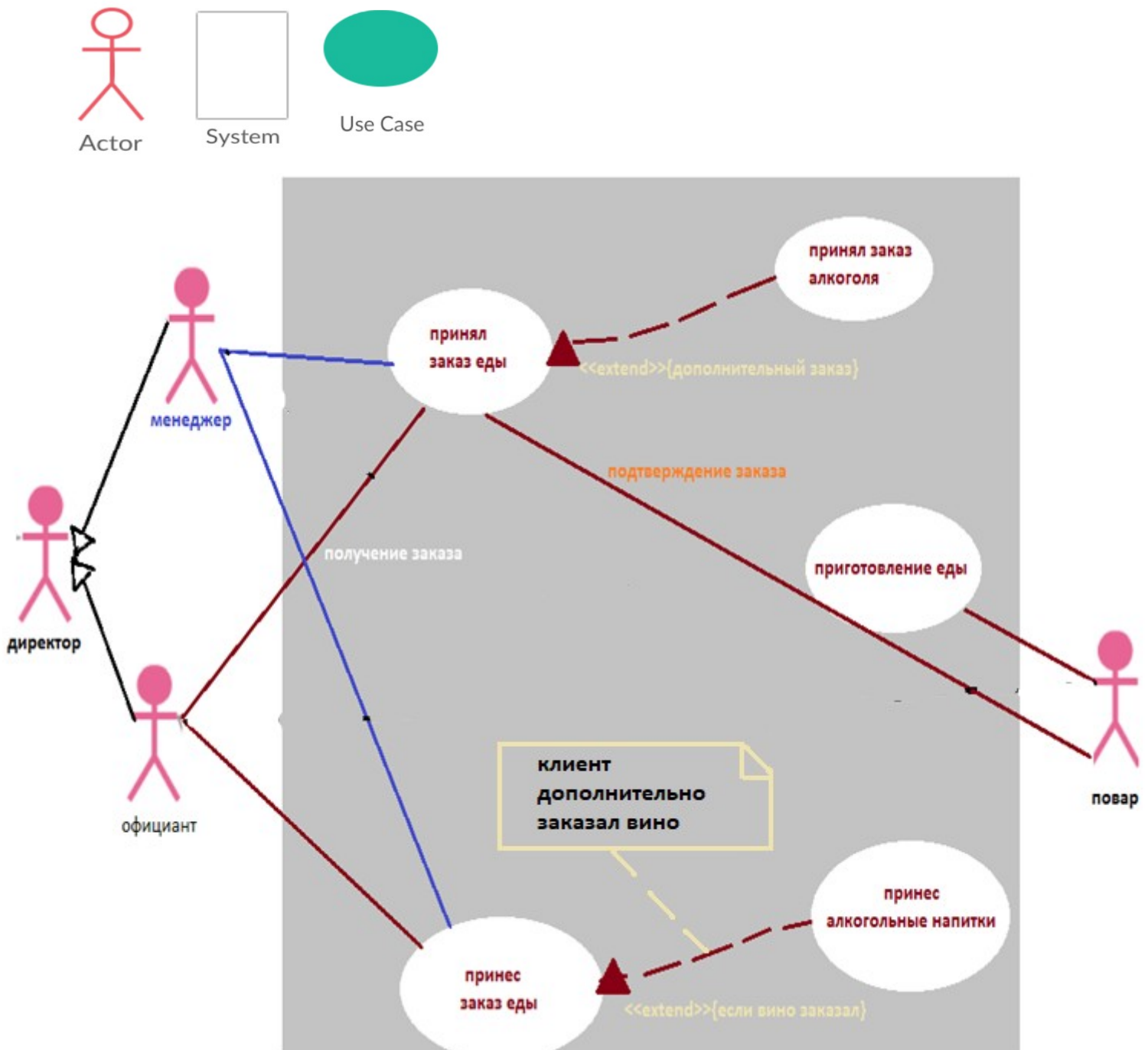
Цель: научиться строить UML диаграммы

К числу средств визуального моделирования объектно-ориентированных информационных систем (ИС) относится **Rational Rose**.

Идея унификации привела к появлению языков 3-го поколения. В качестве стандартного языка третьего поколения был принят Unified Modeling Language (UML)

прецедент (Use case) - это описание последовательности выполняемых системой действий, которая производит наблюдаемый результат, значимый для какого-то определенного актера (Actor).

USE-CASE состоит из:



Use Case (диаграмма прецедентов)

UML — стандартный язык для написания моделей анализа, проектирования и реализации объектно-ориентированных программных систем UML может использоваться для визуализации, спецификации, конструирования и документирования результатов программных проектов.

UML — это не визуальный язык программирования, но его модели прямо транслируются в текст на языках программирования и даже в таблицы для реляционной БД.

Словарь UML образуют три разновидности строительных блоков: предметы, отношения, диаграммы. Предметы — это абстракции, которые являются основными элементами в модели, отношения

Предметы в UML

В UML имеются четыре разновидности предметов: »

1. структурные предметы; »
2. предметы поведения; »
3. группирующие предметы; »
4. поясняющие предметы.

Эти предметы являются базовыми объектно-ориентированными строительными блоками. Они используются для написания моделей.

Структурные предметы являются существительными в UML-моделях.

Они представляют статические части модели — понятийные или физические элементы.

Перечислим восемь разновидностей структурных предметов.

1. **Класс** — описание множества объектов, которые разделяют одинаковые свойства, операции, отношения и семантику (смысл).

Класс реализует один или несколько интерфейсов.

графически *класс отображается в виде прямоугольника*, обычно включающего секции с именем, свойствами (атрибутами) и операциями.

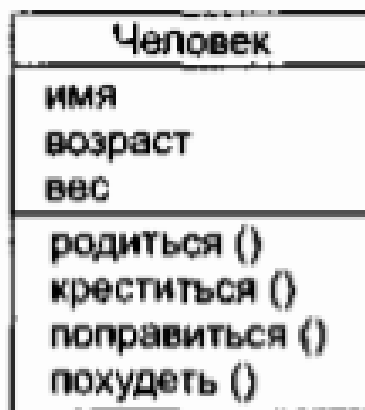


Рис. Классы

2. **Интерфейс.**— набор операций, которые определяют услуги класса или компонента описывает поведение элемента, видимое извне.

конструкция, определяющая методы и свойства, предоставляемые классом.

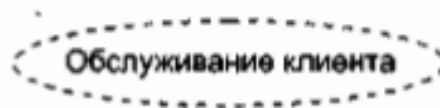
Имя интерфейса обычно начинается с буквы «I».



Интерфейс редко показывают самостоятельно. Обычно его присоединяют к классу или компоненту, который реализует интерфейс.

3. **Кооперация** (сотрудничество) определяет взаимодействие и является совокупностью ролей и других элементов, которые работают вместе для обеспечения коллективного поведения более сложного, чем простая сумма всех элементов.

Конкретный класс может участвовать в нескольких кооперациях. Эти кооперации представляют реализацию паттернов (образцов), которые формируют систему.



4. **Актер** — набор согласованных ролей, которые могут играть пользователи при взаимодействии с системой (ее элементами Use Case). Каждая роль требует от системы определенного поведения.



5. **Элемент Use Case (Прецедент)** — описание последовательности действий, выполняемых системой в интересах отдельного актера и производящих видимый для актера результат



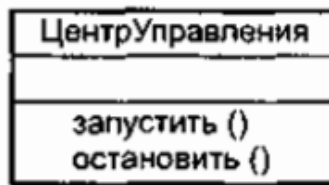
6.

Элемент Use Case — это последовательность взаимодействий в диалоге, выполняемом актером и системой. Запускается элемент Use Case актером, поэтому удобно выявлять элементы Use Case с помощью актеров.

7. **Активный класс** — класс, чьи объекты имеют один или несколько процессов (или потоков) и поэтому могут инициировать управляющую деятельность.

Активный класс похож на обычный класс за исключением того, что *его объекты действуют одновременно с объектами других классов.*

активный класс изображается как утолщенный прямоугольник, обычно включающий имя, свойства (атрибуты) и операции.

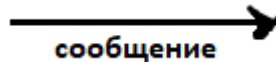


Предметы поведения — динамические части UML.

То что взаимодействует на диаграмме

. Существует две основные разновидности предметов поведения.

1. **Взаимодействие** — поведение, заключающее в себе набор сообщений, которыми обменивается набор объектов в конкретном контексте для достижения определенной цели.



2. **Конечный автомат** — поведение, которое определяет последовательность состояний объекта или взаимодействия, выполняемые в ходе его существования в ответ на события

С помощью конечного автомата может определяться поведение индивидуального класса или кооперации классов.

Элементами конечного автомата являются состояния, переходы (от состояния к состоянию), события (предметы, вызывающие переходы) и действия (реакции на переход).

прямоугольник, обычно включающий его им и его подсостояния (если они есть).

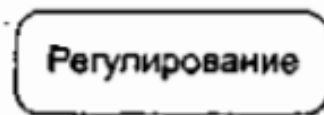
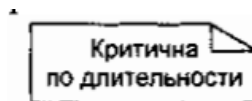


Рис.. Состояния

2.1 **Поясняющие предметы** — разъясняющие части UML-моделей, сообщения, примечания.



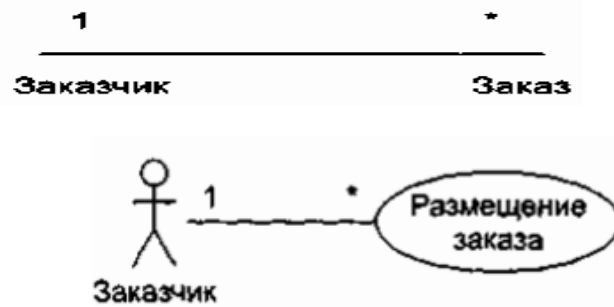
ОТНОШЕНИЯ В UML

Между актером и элементом Use Case возможен только один вид отношения — ассоциация, отображающая их взаимодействие

В UML имеются четыре разновидности отношений:

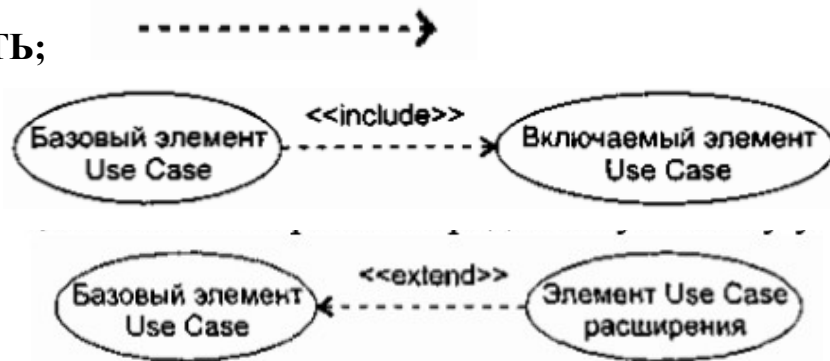
1. Ассоциация
2. Зависимость
3. Обобщение
4. реализация

1) АССОЦИАЦИЯ;



Между актерами допустимо отношение обобщения, означающее, что экземпляр потомка может взаимодействовать с такими же разновидностями экземпляров элементов Use Case, что и экземпляр родителя.

2) ЗАВИСИМОСТЬ;



Между элементами Use Case определены отношение обобщения и две разновидности отношения зависимости — **включения** <<include>> (уточнение элемента Use Case) и **расширения** <<extend>>.



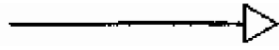
расширения <<extend>>. используют:]

1. для моделирования вариантных частей элементов Use Case;]
2. для моделирования сложных и редко выполняемых альтернативных последовательностей;
3.] для моделирования подчиненных последовательностей, которые выполняются только в определенных случаях;
4.] для моделирования систем с выбором на основе меню.

включения <<include>> Уточнение модели сводится к выявлению одинаковых частей в элементах Use Case и извлечению этих частей. Любые изменения в такой части, выделенной в

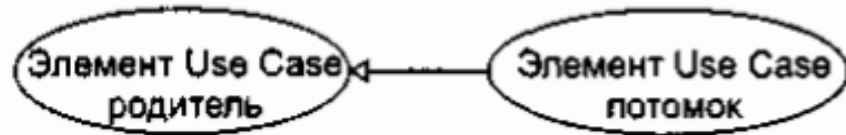
отдельный элемент Use Case, будут автоматически влиять на все элементы Use Case, которые используют ее совместно.

3) ОБОБЩЕНИЕ;

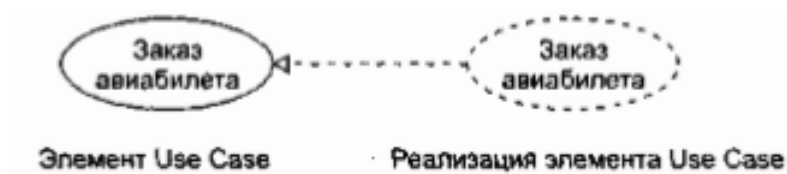


абстрактные элементы Use Case могут использоваться другими абстрактными элементами Use Case. Так образуется иерархия. При построении иерархии абстрактных элементов Use Case руководствуются правилом: выделение элементов Use Case прекращается при достижении уровня отдельных операций над объектами.

Абстрактный актер — это общий фрагмент роли в нескольких конкретных актерах.



4) РЕАЛИЗАЦИЯ.



кооперация реализует конкретный элемент Use Case.

требования к информационной системе авиакассы задаются множеством элементов Use Case, каждый из которых реализуется отдельной кооперацией. Все кооперации применяют одни и те же классы, но все же имеют разную функциональную организацию.

Эти отношения являются базовыми строительными блоками отношений. Они используются при написании моделей.

1. **Зависимость** — семантическое отношение между двумя предметами, в котором изменение в одном предмете (независимом предмете) может влиять на семантику другого предмета (зависимого предмета).

2. **Ассоциация** — структурное отношение, которое набор связей, являющихся соединением между двумя

3. **Обобщение** — отношение специализации/обобщения, в котором объекты специализированного элемента (потомка, ребенка) могут заменять объекты обобщенного элемента (предка, родителя).

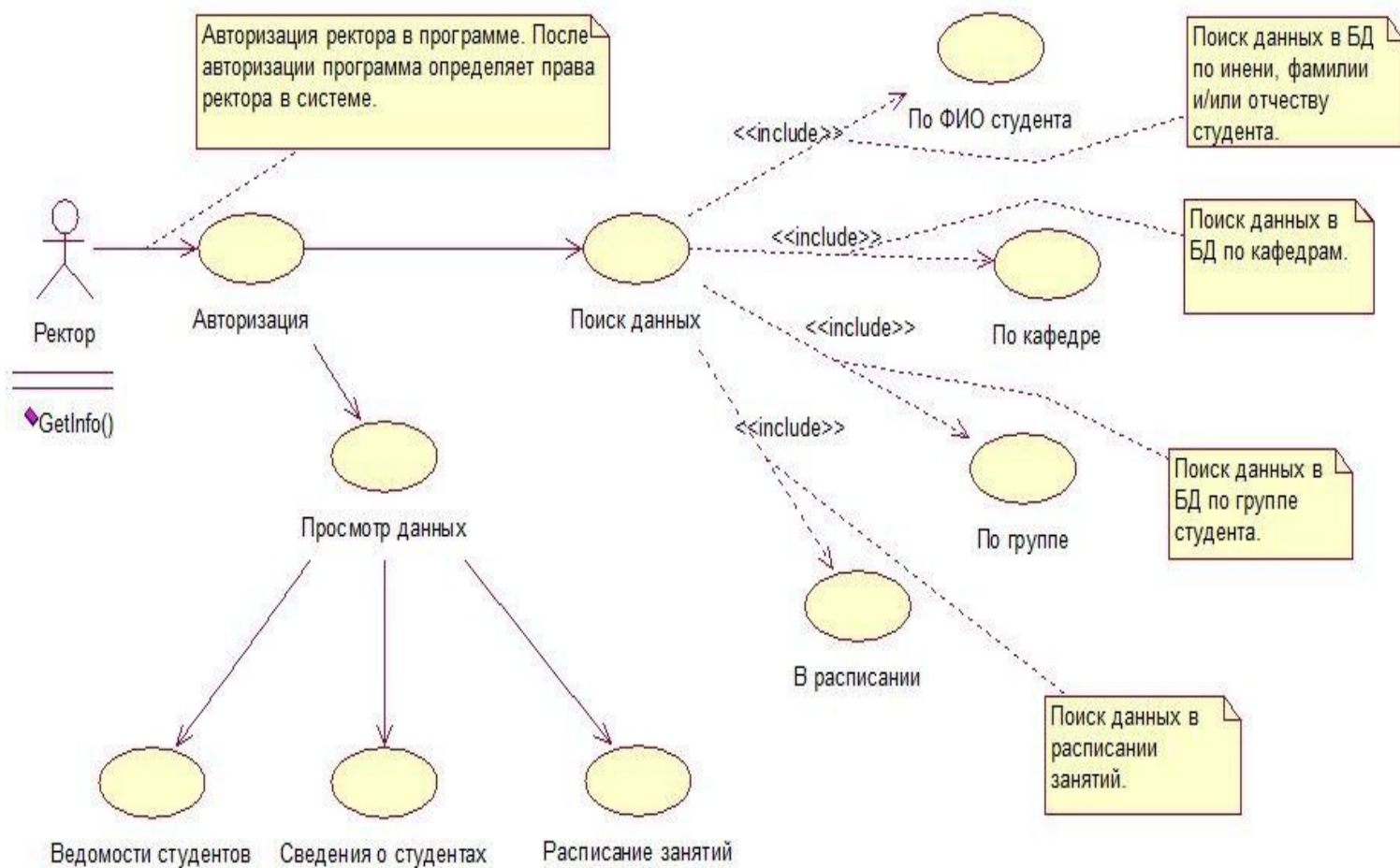
Иначе говоря, потомок разделяет структуру и поведение родителя.

4. **Реализация** — семантическое отношение между классификаторами, где один классификатор определяет контракт, который другой классификатор обязуется выполнять (к классификаторам относят классы, интерфейсы, компоненты, элементы Use Case, кооперации).

Отношения реализации применяют в двух случаях: между интерфейсами и классами (или компонентами), реализующими их; между элементами Use Case и кооперациями, которые реализуют их.

связывают эти предметы, диаграммы группируют коллекции предметов.





Контрольные вопросы:

1. Из чего состоит use-case?
2. Что такое UML?
3. Что такое use-case?
4. Назовите и опишите разновидности предметов в UML
5. Какие отношения в UML вы знаете, как они обозначаются?

Список использованных источников:

1. Технологии разработки программного обеспечения С.А. Орлов
2. Технологии разработки программного обеспечения В.В. Бахтизин, Л.А. Глухова
3. Project Management For Dummies / Управление проектами для "чайников"
4. Л. Н. Боронина З. В. Сенук основы управления проектами
5. http://www.kpms.ru/Automatization/CASE_tools.htm
6. <https://studfiles.net/preview/3828360/page:10/>