

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
Харківський національний автомобільно-дорожній університет

До друку і в світ дозволяю  
Заступник ректора

\_\_\_\_\_ І.П. Гладкий

Шевченко В.О.

## **КОНСПЕКТ ЛЕКЦІЙ**

з дисципліни **"Теорія систем та системний аналіз"**  
за напрямом підготовки 6.050201

Усі цитати, цифровий, фактичний  
матеріал і бібліографічні відомості  
перевірені, написання

\_\_\_\_\_   
одиниць відповідає стандартам

Затверджено  
Радою Факультету МТЗ  
протокол № \_\_\_\_ від

Укладач

В.О. Шевченко

Відповідальний за випуск

Харків 2015

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
Харківський національний автомобільно-дорожній університет

Шевченко В.О.

**КОНСПЕКТ ЛЕКЦІЙ**

з дисципліни **"Теорія систем та системний аналіз"**  
за напрямом підготовки 6.050201

Харків 2015

## Содержание

Лекция 1. Основные понятия теории систем.....	3
Лекция 2. Классификация и свойства систем.....	10
Лекция 3. Системный анализ как инструмент познания систем.....	13
Лекция 4. Методология системного анализа.....	18
Лекция 5. Моделирование систем.....	22
Лекция 6. Методы моделирования.....	28
Лекция 7. Методы формализации моделей.....	37
Лекция 8. Графические методы формализации моделей.....	44
Лекция 9. Объектно-ориентированный анализ и проектирование сложных систем.....	54
Лекция 10. Введение в язык визуального моделирования UML.....	63
Лекция 11. Диаграмма вариантов использования.....	72
Лекция 12. Диаграммы UML.....	78
Лекция 13. Особенности реализации языка UML в CASE-инструментарии Rational Rose.....	85
Литература.....	88

## Лекция 1.

### ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ СИСТЕМ

#### Понятие системы и системного подхода

В настоящий момент термины теория систем и системный анализ все еще не нашли общепринятого, стандартного истолкования. Причина этого факта заключается в динамичности процессов в области человеческой деятельности и в принципиальной возможности использовать системный подход практически в любой решаемой человеком задаче.

**Системный подход** – направление методологии исследования, которое реализует на практике принципы целостности рассмотрения явлений и процессов во всей сложности, взаимосвязи и взаимообусловленности их развития.

**Теория систем (ТС)** – научная дисциплина, изучающая различные явления, отвлекаясь от их конкретной природы и основываясь лишь на формальных взаимосвязях между различными составляющими их факторами и на характере их изменения под влиянием внешних условий. При этом результаты всех наблюдений объясняются лишь взаимодействием их компонентов, например характером их организации и функционирования. Другими словами, теория систем изучает самые фундаментальные понятия и аспекты систем.

Основателем системной концепции под названием "Общая теория систем" считается австрийский биолог Карл Людвиг фон Берталанфи, который ее предложил в 30-х годах прошлого столетия. Основной идеей общей теории систем, является признание изоморфизма законов, управляющих функционированием системных объектов.



**Карл Людвиг фон Берталанфи** (19 сентября 1901, Вена – 12 июня 1972, Нью-Йорк) – австрийский биолог, постоянно проживавший в Канаде и США с 1949 года. Первооснователь обобщенной системной концепции под названием "Общая теория систем". Постановщик системных задач — прежде всего, в сфере разработки математического аппарата описания типологически несходных систем.

**Изоморфизм** – понятие математической логики, означающее соотношение между двумя любыми объектами тождественной структуры.

Для ТС объектом исследования является не "физическая реальность", а "система", т.е. абстрактная формальная взаимосвязь между основными признаками и свойствами.

**Система** – это полный, целостный набор элементов (компонентов), взаимосвязанных и взаимодействующих между собой так, чтобы могла реализоваться функция системы.

**Объект.** Объектом познания является часть реального мира, которая выделяется и воспринимается как единое целое в течение длительного времени. Объект может быть материальным и абстрактным, естественным и искусственным. Реально объект обладает бесконечным набором свойств различной природы. Практически в процессе познания взаимодействие осуществляется с ограниченным множеством свойств, лежащих в пределах возможности их восприятия и необходимости для цели познания. Поэтому система как образ объекта задаётся на конечном множестве отобранных для наблюдения свойств.

**Внешняя среда.** Понятие "система" возникает там и тогда, где и когда мы материально или умозрительно проводим замкнутую границу между неограниченным или некоторым ограниченным множеством элементов. Те элементы, которые попадают внутрь, – образуют систему.

Те элементы, которые остались за пределами границы, образуют множество, называемое в теории систем "системным окружением" или "внешней средой".

Из этих рассуждений вытекает, что нелегко рассматривать систему без ее внешней среды. Система формирует и проявляет свои свойства в процессе взаимодействия с окружением, являясь при этом ведущим компонентом этого воздействия.

**Элементом** системы является часть системы с однозначно определёнными свойствами, выполняющие определённые функции и не подлежащая дальнейшему разбиению в рамках решаемой задачи.

Понятие элемент, подсистема, система взаимопреобразуемы, система может рассматриваться как элемент системы более высокого порядка (метасистема), а элемент при углубленном анализе, как система. При системном подходе объект исследования представляется как система.

Таким образом, система  $S$  представляет собой упорядоченную пару  $S = (A, R)$ , где  $A$  – множество элементов;  $R$  – множество отношений между  $A$ .

**Связь** – совокупность зависимостей свойств одного элемента от свойств других элементов системы. Связь показывает взаимодействие элементов системы между собой и внешней средой.

Система как единое целое существует именно благодаря наличию связей между ее элементами, т.е., иными словами, связи выражают законы функционирования системы. Связи различают по характеру взаимосвязи как прямые и обратные.

**Прямые связи** предназначены для заданной функциональной передачи вещества, энергии, информации или их комбинаций – от одного элемента к другому в направлении основного процесса.

**Обратные связи**, в основном, выполняют осведомляющие функции, отражая изменение состояния системы в результате управляющего воздействия на нее. Открытие принципа обратной связи явилось выдающимся событием в развитии техники и имело исключительно важные последствия. Процессы управления,

адаптации, саморегулирования, самоорганизации, развития невозможны без использования обратных связей. Обратная связь предназначена для выполнения следующих операций:

- сравнение данных на входе с результатами на выходе с выявлением их качественно-количественного различия;
- оценка содержания и смысла различия;
- выработка решения, вытекающего из различия;
- воздействие на ввод.

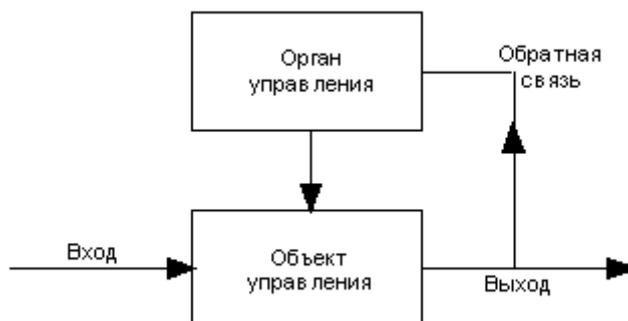


Рисунок 1 – Пример обратной связи

С помощью обратной связи сигнал (информация) с выхода системы (объекта управления) передается в орган управления. Здесь этот сигнал, содержащий информации о работе, выполненной объектом управления, сравнивается с сигналом, задающим содержание и объем работы (например, план). В случае возникновения рассогласования между фактическим и плановым состоянием работы принимаются меры по его устранению.

### Функционирование системы

**Функционирование** любой произвольно выбранной системы состоит в переработке входных (известных) параметров и известных параметров воздействия окружающей среды в значения выходных (неизвестных) параметров с учетом факторов обратной связи.

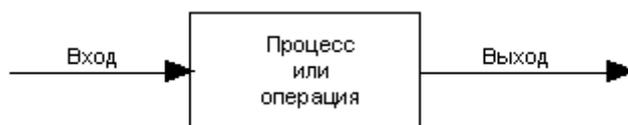


Рисунок 2 – Функционирование системы

**Вход** – это различные точки приложения воздействия внешней среды на систему. Входами системы могут быть информация, вещество, энергия и т.д., которые подлежат преобразованию.

**Выход** – это различные точки приложения воздействия системы на внешнюю среду. Выход системы представляет собой результат преобразования информации, вещества и энергии.

**Процесс** – это преобразование входа в выход.

Управление системой связано с понятиями прямой и обратной связи и ограничениями.

**Ограничения** определяют условия функционирования системы и обеспечивают соответствие между выходом системы и требованием к нему. Если заданное требование не выполняется, ограничение не пропускает его через себя. Ограничение, таким образом, играет роль согласования функционирования данной системы с целями потребителя.

**Критерии** – признаки, по которым производится оценка соответствия функционирования системы желаемому результату (цели) при заданных ограничениях.

Определение функционирования системы связано с понятием "проблемной ситуации", которая возникает, если имеется различие между необходимым (желаемым) выходом и существующим (реальным) входом. Решить проблему – значит скорректировать старую систему или сконструировать новую, желаемую.

При изменении структуры системы происходит развитие системы. **Развитием** называется деятельность системы со сменой цели системы.

**Цель системы** – это представление некоторого наиболее предпочтительного состояния системы, которое позволяет решать задачу или проблему при данных ресурсах.

**Задача** представляет собой множество исходных данных, описание цели, определенной над множеством этих данных, и описание возможных стратегий достижения этой цели или возможных промежуточных состояний исследуемого объекта. Решить задачу означает определить ресурсы и пути достижения указанной цели при исходных посылах.

**Проблема** – это описание ситуации, в которой определены цель, достигаемые результаты, ресурсы и стратегия достижения цели. Проблема проявляется поведением системы.

Если система способна переходить из одного состояния в другое, то говорят, что она обладает **поведением**. Этим понятием пользуются, когда неизвестны закономерности перехода из одного состояния в другое. Тогда говорят, что система обладает каким-то поведением, и выясняют его характер, алгоритм.

**Состоянием системы** называется совокупность свойств, которыми система обладает в каждый момент времени.

### **Структура системы**

Любая задача системного анализа начинается с построения модели исследуемой системы. Для решения задачи построения модели необходимо вначале произвести изучение структуры системы, выполнить анализ ее компонентов, выявить взаимосвязи между отдельными элементами.

**Структура системы** – это совокупность элементов системы и связей между ними.

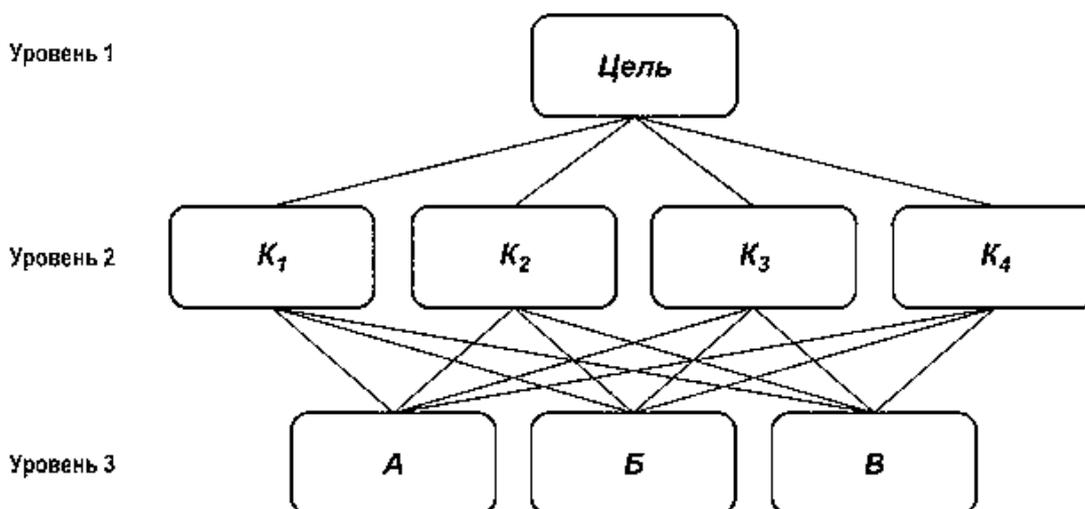
Структурные представления являются средством исследования систем. Одну и ту же систему можно представить различными структурами, необходимый

выбор которых обусловлен содержанием исследований, проводимых на данном этапе. Рассмотрим основные способы представления структур.

**1. Линейная структура** – это структура со строго упорядоченным взаимоотношением элементов "один к одному". Пример: структура станций метро на одной линии в одном направлении.



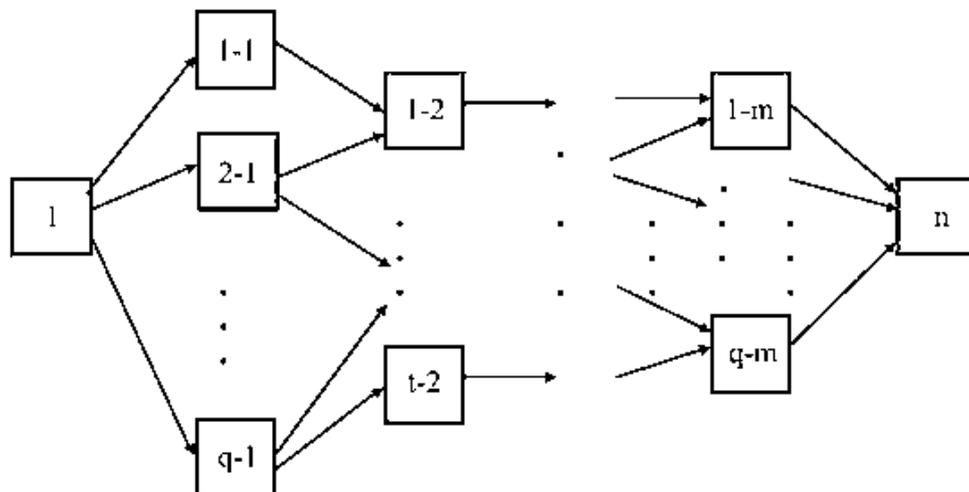
**2. Иерархическая структура** – многоуровневая форма организации элементов со строгой соотнесенностью элементов нижнего уровня определенным элементам верхнего уровня.



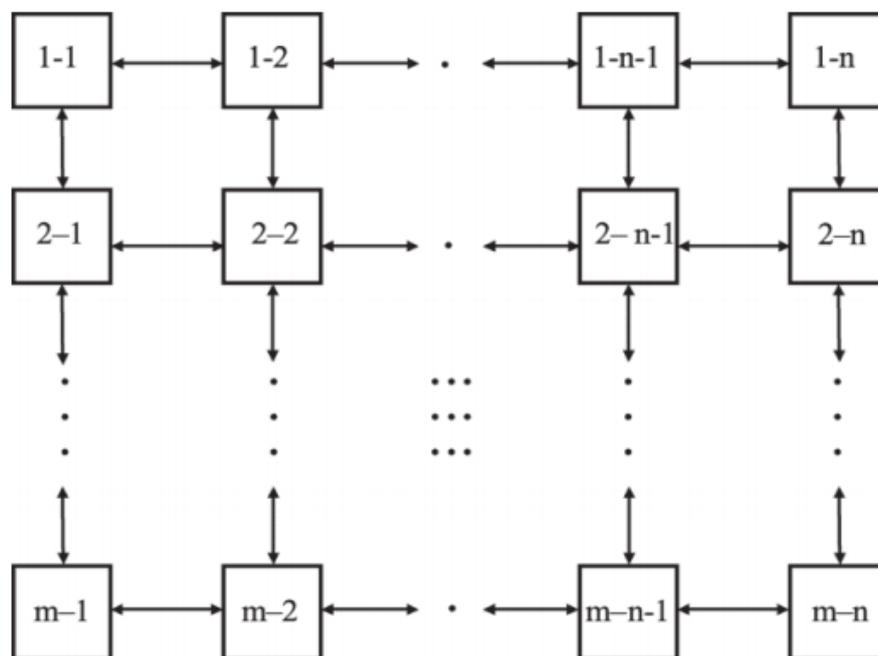
**3. Древоподобная структура** – это иерархическая структура, в которой каждый элемент нижележащего уровня подчинён только одному элементу (одной вершине) вышестоящего.



**4. Сетевая структура** представляет отображение взаимосвязи между элементами одного иерархического уровня, например, параллельная работа станков в цеху.



**5. Матричная структура** – представляет систему, в которой все элементы равнозначны, например, кристаллическая решетка твердого тела.



Структура является статической моделью системы и характеризует только строение системы, не учитывая множества свойств (состояний) её элементов.

## Лекция 2

### КЛАССИФИКАЦИЯ И СВОЙСТВА СИСТЕМ

#### Классификация систем

**Классификацией** называется разбиение на классы по наиболее существенным признакам. Под классом понимается совокупность объектов, обладающих некоторыми признаками общности. Признак (или совокупность признаков) является основанием (критерием) классификации.

Система может быть охарактеризована одним или несколькими признаками и соответственно ей может быть найдено место в различных классификациях, каждая из которых может быть полезной при выборе методологии исследования. Обычно цель классификации ограничить выбор подходов к отображению систем, выработать язык описания, подходящий для соответствующего класса.

Пример классификации систем:

<b>Основание (критерий) классификации</b>	<b>Классы систем</b>
По содержанию	Реальные - естественные - искусственные Абстрактные
По назначению	Производящие Управляющие Обслуживающие Обучающие
По взаимодействию с внешней средой	Открытые Закрытые Смешанные
По структуре	Простые Сложные Большие
По характеру функций	Специализированные Многофункциональные
По характеру развития	Стабильные Развивающиеся
По степени организованности	Хорошо организованные Плохо организованные
По сложности поведения	Автоматизированные Автоматические С интегрированным интеллектом
По характеру связи между элементами	Детерминированные Стохастические

По характеру структуры управления	Централизованные Децентрализованные Смешанные
По характеру функционирования	Непрерывно функционирующие Дискретно функционирующие
По динамике функционирования	Статические Динамические

**Реальные** – это материальные системы, существующие объективно.

**Искусственные** – это системы, которые созданы человеком для достижения конкретной цели.

**Естественные** – это системы, которые существуют в объективной действительности (живая и неживая природа, общество).

**Абстрактные** (концептуальные, идеальные) системы являются продуктом мышления человека и отображают реальную действительность, объективный мир.

**Открытые** – это системы, в которых процессы определяются влиянием внешней среды и сами предоставляют на него существенное действие.

**Закрытые** – это системы, которые в процессе функционирования используют только ту информацию, которая производится внутри самой системы.

**Простые системы** – это системы, не имеющие разветвленных структур, состоящие из определенного количества элементов и связей как внутри системы, так и со средой. Функционирование простых систем можно исследовать без разбивки их на более мелкие подсистемы.

**Сложными** называют системы с большим числом элементов и внутренних связей, которые характеризуются неоднородностью и разнокачественностью, структурным разнообразием и выполняют сложные функции. Компоненты сложных систем могут рассматриваться как подсистемы, каждая из которых может быть детализирована еще более простыми подсистемами и т.д. до тех пор, пока не будет получен элемент.

**Большими системами** называют системы, результат функционирования которых в пространстве и времени невозможно наблюдать одному человеку или группе людей. Большие системы являются подклассом сложных систем.

**Хорошо организованные системы** – это системы, для которых можно определить отдельные элементы, связи между ними, правила объединения в подсистемы и оценить связи между компонентами системы и ее целями. Такие системы могут описываться в виде математических зависимостей, которые связывают цель и средства ее достижения критерием эффективности или оценками функционирования. В хорошо организованных системах используется, в основном, количественная информация.

**Плохо организованные системы** – это системы, в которых закономерности связей между элементами можно определить при получении статистических закономерностей с некоторыми показателями вероятности. Примеры: системы управления, экономические системы. В плохо организованных системах используется, в основном, качественная информация.

**Детерминированные** – это системы, в которых переход из одного состояния в другое (то есть поведение системы) является определенным.

**Стохастические** – системы, в которых переход из одного состояния в другое (то есть поведение системы) не является четко определенным и рассматривается как случайный процесс.

**Статические системы** – это системы с одним возможным состоянием.

**Динамические системы** – это системы с множеством состояний, в которых со временем происходит переход из одного состояния в другое. Практически все системы являются динамическими.

### **Свойства систем.**

**1. Целостность и делимость.** Система есть, прежде всего, целостной совокупностью элементов. Это означает, что, с одной стороны, система – это целостное образование, а с другой – в системе четко могут быть выделены отдельные целостные объекты (элементы). Но не компоненты образуют целое (систему), а наоборот, при разделении целого выявляют компоненты системы. **Первичность целого – главный постулат теории систем.**

**2. Эмерджентность системы** (несводимость свойств отдельных элементов к свойствам системы в целом). Система не сводится к простой совокупности элементов; разделяя систему на части, исследуя каждую из них отдельно, невозможно познать все свойства системы в целом.

**3. Иерархичность системы** – наличие нескольких уровней, при которых каждую систему можно рассматривать как элемент системы высшего порядка (надсистемы), а элементы рассматриваемой системы – как подсистемы.

**4. Взаимосвязанность структуры и среды** – формирование и проявление свойств системы в процессе ее взаимодействия со средой. Система развивается под воздействием внешней среды, но при этом пытается сохранить качественную определенность и свойства, которые обеспечивают относительную стойкость и адаптивность ее функционирования.

**5. Ориентированность** – зависимость каждого элемента системы от его места, функций внутри целого.

**6. Конфигурируемость** – возможность изменения структур элементов и подсистем сложной системы с целью оптимизации ее работы, повышение функциональных возможностей и эффективности функционирования.

**7. Надежность** – способность сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения.

**8. Адаптируемость** – способность изменять поведение или структуру с целью сохранения, улучшения или приобретения новых качеств в условиях изменения внешней среды.

**9. Размерность системы** – количество компонентов системы и связей между ними.

**10. Эффективность** – способность к достижению поставленных целей за определенный период времени при использовании определенного количества ре-

сурсов. Эффективность определяет соответствие между реально полученными (фактическими) и требуемыми результатами функционирования системы. По сути, определяет степень достижения цели.

### Лекция 3

## СИСТЕМНЫЙ АНАЛИЗ КАК ИНСТРУМЕНТ ПОЗНАНИЯ СИСТЕМ

### Задачи исследований системного анализа

**Системный анализ** – научный метод познания, представляющий собой последовательность действий по установлению структурных связей между переменными или элементами исследуемой системы.

**Анализ системы** состоит в определении основных процессов, факторов как внутри системы, так и во внешней среде, влияющих на функционирование и развитие системы. Для этого возможно использование следующих видов анализа:

1. Функционально-структурный анализ существующей системы, позволяющий сформулировать требования к создаваемой системе. Он включает уточнение состава и законов функционирования элементов, алгоритмов функционирования и взаимовлияний подсистем, разделение управляемых и неуправляемых характеристик, анализ целостности системы, формулирование требований к создаваемой системе.

2. Морфологический анализ – анализ взаимосвязи компонентов.

3. Генетический анализ – анализ предыстории, причин развития ситуации, имеющих тенденций, построение прогнозов.

4. Анализ аналогов.

5. Анализ эффективности (по результативности, ресурсоемкости, оперативности). Он включает выбор шкалы измерения, формирование показателей эффективности, обоснование и формирование критериев эффективности, непосредственно оценивание и анализ полученных оценок.

6. Формирование требований к создаваемой системе, включая выбор критериев оценки и ограничений.

Системный анализ допускает рассмотрение широкого спектра разнообразных задач для всесторонних исследований и анализа систем.

**Главная задача системных исследований** – поиск эффективных методов и средств исследования и управления объектами и их взаимодействия с внешней средой.

К основным задачам, которые решаются с помощью системного анализа и теории систем, можно отнести такие:

- определение или выбор оптимальной структуры систем;
- выявление цели функционирования и развития систем;
- изучение организации взаимодействия между подсистемами и элементами;
- учет влияния внешней среды;
- влияние системы на внешнюю среду;
- выявление и четкая формулировка проблемы (задачи) в условиях неопре-

деленности;

- выбор оптимальных алгоритмов функционирования систем.

Разнообразие задач системного анализа требует развитой методологии проведения исследований. Методология системного анализа направлена на формализацию процесса исследования системы, процесса постановки и решения проблемы.

**Проблема** – ситуация несоответствия желаемого и существующего.

В системном анализе проблемы разделяются на три класса:

**Проблемы первого класса** – хорошо структурированные, выражены количественно. Эти проблемы хорошо поддаются формализации и решаются с использованием математического аппарата, позволяющего получать количественные оценки, показывающие эффективность разрешения проблемной ситуации. При решении таких проблем могут быть использованы методы элементарной математики (алгебры, геометрии и т.д.), теории вероятностей, математической статистики, интегрального и дифференциального исчисления, математического программирования и т.д. К ним могут быть отнесены некоторые технические, экономические и организационные проблемы, например, решение вопроса выпуска продукции в условиях ограниченности складского хранения – что, в каком количестве производить, если склад готовой продукции «не резиновый».

**Второй класс** – слабоструктурированные проблемы. Они, как правило, являются комбинированными, содержащими как количественные, так и качественные характеристики проблемных ситуаций, что обуславливает применение малоизвестных и разработку новых методов, обеспечивающих эффективное решение проблемы. Пример проблем: решение вопроса поступления в ВУЗ после окончания средней школы, поиск и устройство на работу и т.п.

Класс слабоструктурированных проблем является основной темой изучения СА. Здесь для решения проблемы используются как методы вычислительной математики, так и математической логики в совокупности с методами мягких вычислений, предложенных Лотфи Заде.

**Мягкие вычисления** – термин, введенный Лотфи Заде в 1994 году, обозначающий совокупность неточных, приближенных методов решения задач. В область мягких вычислений входят такие методы как нечеткая логика, нейронные сети, генетические алгоритмы, методы теории хаоса и роевого интеллекта.

**Нечёткая логика** (англ. *fuzzy logic*) – раздел математики, являющийся обобщением классической логики и теории множеств, базирующийся на понятии нечёткого множества, впервые введённого Лотфи Заде в 1965 году. Понятие нечёткого множества – это попытка математической формализации нечеткой информации для построения математических моделей. В основе этого понятия лежит представление о том, что каждый элемент какого-либо множества, принадлежит к этому множеству с определенной степенью вероятности.

**Нейронные сети** – математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма.

**Генетический алгоритм** (англ. *genetic algorithm*) – эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомым параметров с использованием механизмов, аналогичных естественному отбору в природе.

**Теория хаоса** – математический аппарат, описывающий поведение некоторых нелинейных динамических систем, подверженных при определённых условиях явлению, известному как хаос (динамический хаос, детерминированный хаос). Поведение такой системы кажется случайным, даже если модель, описывающая систему, является детерминированной. Примеры: атмосфера, турбулентные потоки и др.

**Роевой интеллект** (англ. *Swarm intelligence*) представляет собой много-агентную систему, обладающую самоорганизующимся поведением, которое, суммарно, должно проявлять некоторое разумное поведение. Системы роевого интеллекта, как правило, состоят из множества агентов (боидов), локально взаимодействующих между собой и с окружающей средой. Идеи поведения, как правило, исходят от природы, а в особенности, от биологических систем. Каждый боид следует очень простым правилам и, несмотря на то, что нет какой-то централизованной системы управления поведением, которая бы указывала каждому из них, что ему следует делать, локальные и, в некоторой степени, случайные взаимодействия приводят к возникновению интеллектуального глобального поведения, неконтролируемого отдельными боидами.

**Проблемы третьего класса** – неструктурированные. Для решения этих проблем, обычно применяются эвристические (интуитивно-логические) методы решения, с помощью которых неструктурированная проблема переводится в класс слабоструктурированных, после чего можно использовать методы системного анализа, который позволяет либо найти правильное решение, либо выявить причины появления проблемы. Пример: проблема экологии мегаполиса (страны, континента, планеты), дефицит пресной воды в мире и т.п.

### **Этапы исследования проблемы**

В общем виде системное исследование проблемы состоит из таких этапов:

1. Формулировка проблемы.
2. Определение цели.
3. Формулировка критериев.
4. Определение имеющихся ресурсов для достижения цели.
5. Генерация альтернатив и сценариев.

**Первый этап.** Системное исследование произвольной проблемы начинается из формулировки и описания проблемной ситуации. Для начала необходимо определить внешние признаки проявления проблемы, ее временные параметры, источники возникновения проблемы, обуславливающие ее факторы, характер влияния проблемы на основные элементы объекта исследования, возможные исследования развития проблемной ситуации.

В качестве инструмента исследования можно использовать анализ логической структуры проблемы, изучить ее в ретроспективном аспекте, оценить воз-

возможные пути развития в будущем и во взаимосвязи с другими проблемами. При анализе сложных проблем используют приемы декомпозиции (разделение целого на части), модели типа "черного ящика" и др.

**Декомпозиция** – это научный метод, использующий структуру задачи и позволяющий заменить решение одной большой задачи решением серии меньших задач, пусть и взаимосвязанных, но более простых. Декомпозиция, как процесс расчленения, позволяет рассматривать любую исследуемую систему как сложную, состоящую из отдельных взаимосвязанных подсистем, которые, в свою очередь, также могут быть расчленены на части. В качестве систем могут выступать не только материальные объекты, но и процессы, явления и понятия.

**Чёрный ящик** – термин, используемый для обозначения системы, внутреннее устройство и механизм работы которой очень сложны, неизвестны или неважны в рамках данной задачи. "Метод черного ящика" – метод исследования таких систем, когда вместо свойств и взаимосвязей составных частей системы, изучается реакция системы, как целого, на изменяющиеся условия.

**Второй этап.** Цель – это желаемое состояние системы или результат ее деятельности. Определение целей есть зеркальное отражение формулирования проблемы, поскольку проблема – это несоответствие между необходимым (желаемым) и фактическим положением дел. Правильная постановка целей может быть равносильна "половине" решения проблемы.

При системном анализе цель можно рассматривать с позиций субъекта и объекта исследования.

Цель с позиции субъекта определяет цель анализа, описания, проектирования (создания или реорганизации) и управления:

- цель анализа объекта – выявить наличие и место противоречий (проблемной ситуации), причин их возникновения и способов устранения;
- цель описания объекта – представить проблемную ситуацию в виде, удобном для анализа;
- цель проектирования – разрешить проблемную ситуацию с помощью нового объекта или реорганизации старого;
- цель управления – разрешить проблемную ситуацию путем удержания функционирования объекта в заданном состоянии или перевода его в новое состояние.

Цель с позиции объекта определяет цель его функционирования (существования), которая может быть заложена при его создании либо формироваться внутри него.

На практике, как правило, существует несколько целей и потому важно, кроме определения главной цели, не выпустить некоторые из существенных среди других. Для этого применяют метод построения дерева целей. Преимуществом этого метода является то, что он делает возможным разделение сложного задания, которое трудно формализовать, на совокупность более простых заданий, для решения которых существуют проверенные приемы и методы.

**Третий этап.** Задача формирования критериев решается непосредственно после того, как сформулированы цели системного анализа. Ситуация становится

понятной, если к критериям относиться как к количественным моделям качественных целей. Задача системного аналитика состоит в том, чтобы формализовать проблемную ситуацию, возникающую в ходе системного анализа. Этой цели как раз и служит этап формирования критериев. Сформированные критерии в некотором смысле должны заменять цели. От критериев требуется как можно большее сходство с целями, чтобы оптимизация по критериям соответствовала максимальному приближению к целям.

При решении задач системного анализа, возникает ситуация, когда невозможно предложить один критерий, адекватно отражающий цель исследования. Даже одну цель редко удается выразить одним критерием, хотя к этому необходимо стремиться. Еще более усложняется задача в случае, когда сформулировано несколько целей системного анализа, отражающих разные системы ценностей. В этом случае исследователь тем более вынужден формировать несколько критериев и в последующем решать многокритериальную задачу. Таким образом, можно отметить, что многокритериальность является способом повышения адекватности описания цели. Однако введение многокритериальности в задачах системного анализа не должно быть самоцелью. Качество постановки задачи заключается не только и не столько в количестве критериев, сколько в том, чтобы они достаточно адекватно описывали цель системного анализа. Критерии должны описывать по возможности все важные аспекты цели, но при этом желательно минимизировать число необходимых критериев. Основные критерии, наиболее часто встречающиеся в анализе сложных технических систем: экономические критерии – прибыль, рентабельность, себестоимость; технико-экономические – производительность, надежность, долговечность; технологические – выход продукта, характеристики качества и пр.

**Четвертый этап.** При постановке и решении задач системного анализа необходимо учитывать не только цели, но и возможности, имеющиеся для решения поставленных задач. В первую очередь необходимо учитывать ресурсы, требуемые для решения задач (денежные, людские, материальные и др.). Чтобы удовлетворить условиям непревышения количества имеющихся ресурсов, в постановку задачи системного анализа вводят ограничения. Между целевыми критериями и ограничениями имеется сходство и различия. Общее заключается в том, что и критерий и ограничения являются математической формулировкой некоторых условий. В некоторых задачах оптимизации они могут выступать равноправно. Однако на этапе формирования критериев целевой критерий открывает возможности для генерирования новых альтернатив в поисках лучшей из них, а ограничение заведомо уменьшает их число, запрещая некоторые из них. Одними целевыми критериями можно жертвовать ради других, ограничения же исключить нельзя, они должны четко соблюдаться.

**Пятый этап.** На данном этапе необходимо сгенерировать множество альтернатив, из которых затем будет осуществляться выбор наилучшего пути развития системы. На данном этапе используют методы экспертных оценок.

## Лекция 4

### МЕТОДОЛОГИЯ СИСТЕМНОГО АНАЛИЗА

Под **методологией системного исследования** понимают совокупность системных методов и средств, направленных на решение сложных и комплексных проблем. Методология изучает методы научного исследования.

**Метод** – это путь познания, опирающийся на некоторую совокупность ранее полученных общих знаний. Метод является научным средством исследования объекта.

В системном анализе применяется комплекс общенаучных, экспериментальных, статистических и математических методов, а также методы экспертных оценок.

**Общенаучные методы** используются в теоретических и эмпирических исследованиях. Они включают:

**Анализ** – метод исследования, который включает в себя изучение предмета путем мысленного расчленения его на составные элементы (части объекта, его признаки, свойства, отношения). Каждая из выделенных частей анализируется отдельно в пределах единого целого. Например, анализ производительности труда рабочих производится по предприятию в целом и по каждому цеху.

**Синтез** – метод изучения объекта в его целостности, в единстве и взаимной связи его частей. В процессе научных исследований синтез связан с анализом, поскольку он позволяет соединить части предмета, расчлененного в процессе анализа, установить их связь и познать предмет как единое целое (производительность труда по производственному объединению в целом).

**Абстрагирование** – процесс мысленного отвлечения от ряда свойств и отношений изучаемого явления с одновременным выделением интересующих исследователя свойств (прежде всего существенных, общих). Он применяется в экономических исследованиях для перспективного планирования, когда на основании изучения работы предприятий за прошедший период прогнозируется развитие отрасли или региона на будущий период.

**Идеализация** – мыслительная процедура, связанная с образованием абстрактных (идеализированных) объектов, принципиально несуществующих в действительности с целью предельного упрощения исследуемых объектов и явлений (точка, идеальный газ, абсолютно черное тело и т.п.).

**Конкретизация** – метод исследования предметов во всей их разносторонности, в качественном многообразии реального существования в отличие от абстрактного, отвлеченного изучения предметов. При этом исследуется состояние предметов в связи с определенными условиями их существования и исторического развития. Так, перспективы развития отрасли определяются на основании конкретных расчетов применения новой техники и технологии, сбалансированности трудовых и материальных ресурсов и др.

**Индукция** – метод исследования, при котором общий вывод о признаках множества элементов делается на основе изучения этих признаков у части элементов одного объекта. Так, изучаются факторы, отрицательно влияющие на производительность труда по каждому отдельному предприятию, а затем обобщаются в целом по объединению, в состав которого входят эти предприятия как производственные единицы.

**Дедукция** – метод логического умозаключения от общего к частному, т.е. сначала исследуется состояние объекта в целом, а затем – его составных элементов. Применительно к предыдущему примеру сначала анализируется производительность труда в целом по объединению и далее по его производственным единицам.

**Аналогия** (соответствие, сходство) – умозаключение на основе переноса знания об одном объекте на менее изученный объект, сходный с первым в некотором отношении. Например, производительность труда в объединении может исследоваться не по каждому предприятию, а лишь по выбранным в качестве аналога, выпускающим однородную с другими предприятиями продукцию и имеющим одинаковые условия для производственной деятельности. При этом полученные результаты распространяются на все аналогичные предприятия.

**Моделирование** – метод исследования определенных объектов путем воспроизведения их характеристик на другом объекте – модели, которые представляют собой аналог того или иного фрагмента действительности (материального или мыслительного) – оригинала модели.

**Экспериментальные методы** применяются для точного описания и тщательного изучения рассматриваемых явлений и процессов. К ним относятся:

**Наблюдение** – это метод познания, при котором объект изучают без вмешательства в него; фиксируют, измеряют лишь свойства объекта и характер их изменения.

**Опрос** – это метод сбора первичной информации, источником которой служит словесное или письменное суждение человека. Опрос может проводиться в виде беседы, тестирования или анкетирования. Цель опроса – выявление количественно-качественных характеристик объекта исследования.

**Эксперимент** – это метод познания, в котором создаются и изменяются специальные условия для того, чтобы вызвать изучаемый процесс и влиять на его течение, что дает возможность получить количественные и качественные характеристики исследуемого явления.

**Испытание** – это опытное определение количественных и (или) качественных свойств предмета испытаний в результате воздействий на него.

**Статистические методы** направлены на изучение количественных закономерностей, проявляющихся в структуре, динамике и взаимосвязях изучаемых явлений и процессов. К ним относятся методы математической статистики и теории ошибок, например, методы дисперсионного, корреляционного, факторного анализа и другие, а также методы непараметрической статистики.

**Математическая статистика** – наука, разрабатывающая математические методы систематизации и использования статистических данных для научных и практических выводов.

**Теория ошибок** – раздел математической статистики, посвященный построению уточненных выводов о численных значениях приближенно измеренных величин, а также об ошибках (погрешностях) измерений.

**Непараметрическая статистика** – это раздел математической статистики, представляющий комплекс методов обработки статистических данных, не требующих, чтобы распределение вероятностей было описано каким-либо параметрическим законом распределения (например, нормальным).

**Математические методы** применяются для решения хорошо структурированных, количественно выражаемых проблем. К ним относятся методы исследования операций, например, задачи линейного, нелинейного, динамического программирования, задачи теории массового обслуживания, теории игр и др.

**Экспертные методы** используются для организации и проведения экспертиз для оценки сложных систем, генерации альтернатив и выбора сценария развития системы.

К методам оценки систем относятся:

**Ранжирование.** Метод представляет собой процедуру упорядочения объектов, выполняемую экспертом. На основе знаний опыта эксперт располагает объекты в порядке предпочтения, руководствуясь одним или несколькими выбранными показателями сравнения. В зависимости от вида отношений между объектами возможны различные варианты упорядочения объектов.

**Парное сравнение.** Этот метод представляет собой процедуру установления предпочтения объектов при сравнении всех возможных пар. В отличие от ранжирования, в котором осуществляется упорядочение всех объектов, парное сравнение объектов является более простой задачей.

**Множественные сравнения.** Они отличаются от парных тем, что экспертам последовательно предъявляются не пары, а тройки, четверки и т.д. объектов. Эксперт их упорядочивает по важности или разбивает на классы в зависимости от целей экспертизы. Множественные сравнения занимают промежуточное положение между парными сравнениями и ранжированием.

**Непосредственная оценка.** Метод заключается в присваивании объектам числовых значений в шкале интервалов. Эксперту необходимо поставить в соответствие каждому объекту точку на определенном отрезке числовой оси. При этом необходимо, чтобы эквивалентным объектам приписывались одинаковые числа.

Методы генерации альтернатив и выбора сценария развития системы:

**Методы коллективной генерации идей.** Известны, так же и как метод **мозгового штурма**, или **мозговой атаки**. Данный метод является методом систематической тренировки творческого мышления, нацеленным на открытие новых

идей и достижение согласия группы людей на основе интуитивного мышления. Техника мозгового штурма состоит в следующем.

Собирается группа лиц, отобранных для генерации альтернатив. Главный принцип отбора заключается в подборе специалистов разных профессий, опыта работы и квалификации. Данная группа обсуждает проблему, причем заранее оговаривается, что приветствуются любые идеи, возникшие как индивидуально, так и по ассоциации при выслушивании предложений других участников.

**Разработка сценариев.** В некоторых проблемах искомое решение должно описывать реальное поведение объекта в будущем, определять реальный ход событий. В таких случаях альтернативами являются различные последовательности действий и вытекающих из них событий, которые могут произойти с системой в будущем. Эти последовательности имеют общее начальное состояние и различные траектории движения развития системы. Это различие и приводит к проблеме выбора. Такие гипотетические альтернативные описания поведения системы в будущем называются сценариями.

Для составления сценариев привлекаются специалисты, которые должны знать общие закономерности развития систем. При составлении сценариев проводят анализ внутренних и внешних факторов, влияющих на развитие системы, определяют источники этих факторов, целенаправленно анализируют высказывания ведущих специалистов в научных публикациях по рассматриваемой тематике. Сценарий является предварительной информацией, на основе которой проводится дальнейшая работа по прогнозированию развития системы. Сценарий помогает составить представление о проблеме, затем приступают к более тщательным, как правило, количественным процедурам анализа.

**Морфологические методы.** Основная идея морфологических методов состоит в систематическом переборе всех мыслимых вариантов решения проблемы или развития системы путем комбинирования выделенных элементов или их признаков. Системный аналитик определяет все мыслимые параметры, от которых может зависеть решение проблемы и представляет их в виде матриц-строк. Затем в этой матрице определяются все возможные сочетания параметров по одному из каждой строки. Полученные таким образом варианты подвергаются оценке и анализу с целью выбора наилучшего варианта решения проблемной ситуации.

**Деловые игры.** Деловыми играми называется имитационное моделирование реальных ситуаций. В процессе моделирования участники игры ведут себя таким образом, будто они в реальности выполняют порученную им роль. Реальная ситуация в данном случае заменяется некоторой моделью. Чаще всего деловые игры используются для обучения, однако их с успехом применяют и для экспериментального генерирования альтернатив, особенно в слабо формализованных ситуациях. Важная роль в деловых играх отводится руководителю игры, тому, кто управляет моделью, регистрирует ход игры и обобщает ее результаты.

**Методы экспертного анализа.** Методы экспертного анализа разрабатывались для решения задачи структурирования и системной организации процесса получения и кодирования данных и знаний. Суть методов состоит в подборе группы экспертов, являющихся специалистами в рассматриваемой области зна-

ний. Перед ними формулируется задача, скажем, изложить свое мнение по проблеме, требующей решения, предложить пути развития системы, обосновать траекторию изменения состояний системы в будущем и т. п. После получения ответов появляется как бы коллективное мнение, коллективный взгляд на решаемую проблему. В результате обработки экспертных ответов получают наиболее вероятный прогноз по развитию системы.

**Метод "Дельфи"** – итеративная процедура при проведении мозговой атаки, которая должна снизить влияние психологических факторов при проведении обсуждений проблемы и повысить объективность результатов. В отличие от традиционного подхода к достижению согласованности мнений экспертов путем открытой дискуссии метод "Дельфи" предполагает полный отказ от коллективных обсуждений. Это делается для того, чтобы уменьшить влияние таких психологических факторов, как присоединение к мнению наиболее авторитетного специалиста, нежелание отказаться от публично выраженного мнения, следование за мнением большинства. В методе "Дельфи" прямые дебаты заменены тщательно разработанной программой последовательных индивидуальных опросов, проводимых в форме анкетирования. Ответы экспертов обобщаются и вместе с новой дополнительной информацией поступают в распоряжение экспертов, после чего они уточняют свои первоначальные ответы. Такая процедура повторяется несколько раз до получения приемлемой сходимости совокупности высказанных мнений.

**Методы типа дерева целей** подразумевает использование иерархической структуры, полученной путем деления общей цели на подцели, а их, в свою очередь, на более детальные составляющие – новые подцели, функции и т. д. Древоподобные иерархические структуры используются при исследовании вопросов совершенствования организационных систем.

Общим для всех методик системного анализа является осмысление сущности и основной цели жизнедеятельности системы, формирования вариантов представления системы и выбор лучшего варианта. На каждой стадии исследования используются разнообразные научные методы и приемы, состоящие из неодинакового количества этапов анализа, содержание которых зависит от сложности решаемых заданий.

## Лекция 5

### МОДЕЛИРОВАНИЕ СИСТЕМ

#### Понятие модели и моделирования

Под **моделированием** понимают процесс адекватного отображения наиболее существенных сторон исследуемого объекта или явления с точностью, которая необходима для исследования. Основой моделирования является формализованный подход к исследованию сложной системы.

Теоретической базой моделирования является теория подобия. Подобие – это взаимно однозначное соответствие между двумя объектами, при котором из-

вестны функции перехода от параметров одного объекта к параметрам другого, а математические описания этих объектов могут быть преобразованы в тождественные. Таким образом, моделирование – это процесс представления объекта исследования адекватной (подобной) ему моделью и проведения экспериментов с моделью для получения информации об объекте исследования. При моделировании модель выступает и как средство, и как объект исследований, находящийся в отношении подобия к моделируемому объекту.

**Модель** – это специально создаваемый объект, на котором воспроизводятся определенные характеристики реального исследуемого объекта или процесса с целью их использования для управления или изучения объекта.

Моделирование – неотъемлемый этап всякой целенаправленной деятельности. Целевой характер имеет любая деятельность человека, она всегда целесообразна, целенаправленна. Системность деятельности проявляется в том, что она осуществляется по определённому плану, или, как чаще говорят, по алгоритму. То есть алгоритм – образ будущей деятельности, её модель. В алгоритме моделируются все возможные ситуации, в зависимости от различных промежуточных значений параметров; возможные шаги деятельности не выполняются реально, а проигрываются на модели. Моделирование возникает в таких сферах человеческой деятельности, как познание, общение, практическая деятельность. Человека (субъекта моделирования) могут интересовать: внешний вид, структура, поведение объекта моделирования.

Описание внешнего вида объекта сводится к перечислению его признаков. В языке эти признаки часто выражаются прилагательными: красивый, жёлтый, круглый, длинный и т.п. Моделирование внешнего вида объекта необходимо для идентификации (узнавания) объекта.

Описание структуры обычно сводится к перечислению составных элементов объекта и указанию связи между ними. В языке эти элементы и связи часто выражаются именами существительными: электрон, протон, нейтрон, сила притяжения, энергетический уровень (при описании атома). Моделирование структуры объекта необходимо для её наглядного представления, изучения свойств объекта, выявления значимых связей, изучения стабильности объекта и пр.

Поведение объекта характеризуется изменением его внешнего вида и структуры с течением времени в результате взаимодействия с другими объектами. В языке, как правило, оно выражается глаголами: сохраняется, развивается, укрупняется, перестраивается, преломляется, превращается и т.д. Моделирование поведения необходимо для прогнозирования, установления связей с другими объектами, управления, конструирования технических устройств и пр.

### **Базовые модели системного анализа**

К базовым моделям относятся модель черного ящика, модель состава системы и модель структуры системы.

Наиболее простой моделью является **модель «чёрного ящика»**. «Чёрный ящик» – это система, в которой входные и выходные величины известны, а внутреннее устройство ее и процессы, происходящие в ней, не известны.

Понятие «черный ящик» было предложено Уильямом Россом Эшби (английский психиатр, специалист по кибернетике, пионер в исследовании сложных систем). Модель «чёрного ящика» позволяет изучать поведение систем, т. е. их реакции на разнообразные внешние воздействия, и в то же время абстрагироваться от их внутреннего устройства. Таким образом, система изучается не как совокупность взаимосвязанных элементов, а как нечто целое, взаимодействующее со средой на своих входах и выходах.



Уильям Росс Эшби  
(1903-1972)

Метод «черного ящика» используется при недоступности внутренних процессов системы для исследования. Примерами могут служить изучение деятельности новых лекарственных средств или проверка на готовность к эксплуатации автоматической телефонной станции, которая проводится путем «прозванивания», а не непосредственно проверкой всех блоков, схем и т.д.

Исследование с помощью метода «черного ящика» заключается в том, что осуществляется предварительное наблюдение за взаимодействием системы с внешней средой и составляется список входных и выходных воздействий.

На следующем этапе производится воздействие на входы системы и регистрация ее выходов. В процессе изучения наблюдатель и «черный ящик» образуют систему с обратной связью, а результаты исследования – множество пар состояний входа и выхода, анализ которых позволяет установить между ними причинно-следственную связь.

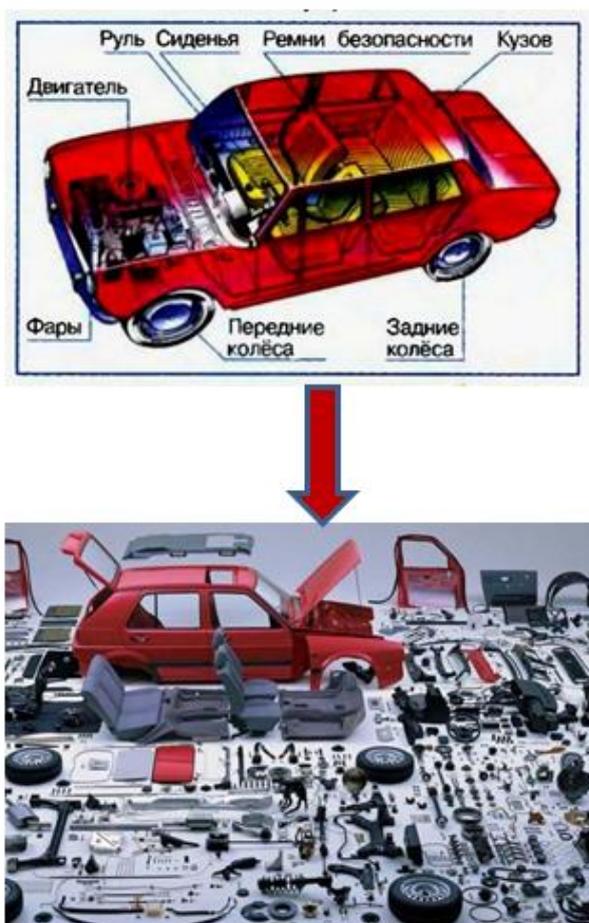
**Модель состава системы** описывает основные составные части системы, просматривает элементы системы как неделимые части и подсистемы, т.е. модель состава иллюстрирует иерархию составных частей системы, *но не рассматривает связи между ними.*

Сложность построения модели состава системы – в ее неоднозначности. Если дать разным экспертам задание определить состав одной и той же системы, то результаты их работы будут различаться, и иногда довольно значительно. Главная трудность в построении модели состава заключается в том, что деление целостной системы на части является относительным, условным, зависящим от целей моделирования (это относится не только к границам между частями системы, но и к границам самой системы). Кроме того, относительным является и выбор

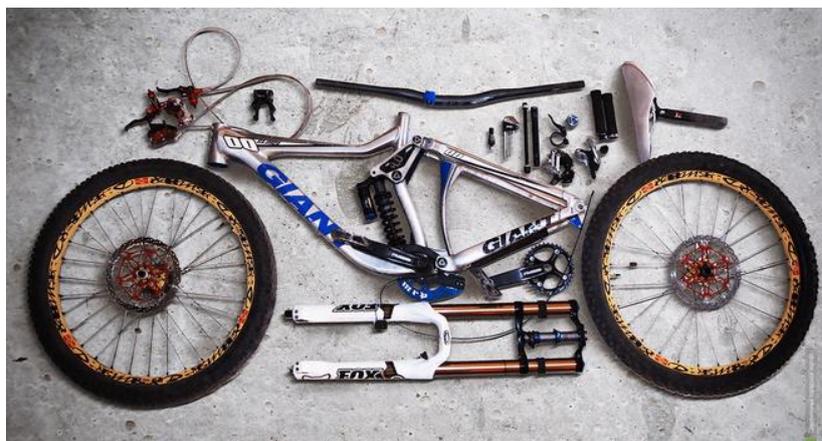
уровня дробления, определение самой малой части – элемента. Для удобства обычно применяют иерархические модели состава системы, состоящие из нескольких уровней детализации.

Пример.

Автомобиль на верхнем уровне состоит из корпуса, салона, движущей части. В зависимости от цели анализа к этому уровню также можно отнести пассажиров и груз. На уровне движущей части он состоит из двигателя, колёс, трансмиссии и т.д. На уровне колеса он состоит из ступицы, резиновой камеры, болтов крепления. Необходимость дальнейшего дробления и рассмотрения, например, из чего состоит болт в колесе, зависит от цели анализа автомобиля.



Для достижения многих практических целей достаточно модели черного ящика или модели состава. Однако, очевидно, есть вопросы, решить которые с помощью этих моделей нельзя. Например, чтобы получить велосипед, недостаточно иметь ящик со всеми отдельными его деталями. Необходимо еще правильно соединить все детали между собой, то есть установить между элементами определенные связи – отношения. Совокупность необходимых и достаточных для достижения цели отношений между элементами называется структурой системы. **Модель структуры системы** отображает связи между компонентами модели.



### **Классификация моделей**

Различают физические и информационные (абстрактные) модели.

**Физические** – это модели, которые сконструированы человеком искусственно или взяты из природы в качестве образцов. Физические модели образуются из совокупности материальных объектов. Для их построения используются различные физические свойства объектов, причём природа применяемых в модели материальных элементов не обязательно та же, что и в исследуемом объекте. Примером физической модели является макет.

К физическим моделям относятся наглядные, аналоговые и натурные.

**Наглядные** модели отражают, главным образом, структуру или геометрические характеристики оригинала. Например: макет здания, чертеж детали, макет самолета. Эти модели, как правило, есть уменьшенные копии реальных объектов той же физической природы.

**Аналоговые** модели используют сходство некоторых физических процессов, протекающих в оригинале и модели. Так, разряд электрического контура моделирует колебания маятника, аналоговые вычислительные машины моделируют технологические процессы и т.п.

**Натурные** – это модели, где исследования проводят на реальном объекте или его уменьшенной копии с последующей обработкой результатов эксперимента на основе теории подобия.

**Информационные (абстрактные)** модели – это описание объекта исследований на каком-либо языке. Абстрактность модели проявляется в том, что её компонентами являются понятия, а не физические элементы (например, словесные описания, чертежи, схемы, графики, таблицы, алгоритмы или программы, математические описания).

Среди информационных (абстрактных) моделей различают гносеологические, инфологические, кибернетические, сенсуальные (чувственные), концептуальные, математические.

**Гносеологические** модели направлены на изучение объективных законов природы (например, модели солнечной системы, биосферы, мирового океана, катастрофических явлений природы).

**Инфологическая (информационно-логическая)** модель – параметрическое представление изучаемого процесса с целью проведения автоматизированного исследования, выполненное с использованием специальных языковых средств, не зависящих от используемых в дальнейшем программных средств. Широко применяются при проектировании баз данных. Для описания инфологических моделей используются как языки аналитического типа (языки спецификаций), так и графические средства. Язык спецификаций – формальный язык, предназначенный для декларативного описания структуры, связей, свойств данных и способов их преобразований. В отличие от языков программирования, используемых при реализации компьютерных программ, языки спецификаций применяются для проведения системного анализа, разработки архитектуры создаваемых программных систем и формальной верификации программного обеспечения. Примеры языков спецификаций: UML, LOTOS, сеть Петри и др.

Верификация – это процесс определения, выполняют ли программные средства и их компоненты требования, предъявляемые к программной системе, другими словами, подтверждение соответствия конечного продукта предопределённым эталонным требованиям.

**Сенсуальная** модель – модель каких-то чувств, эмоций, либо модель, оказывающая воздействие на чувства человека, например, музыка, живопись, поэзия.

**Концептуальная** модель – это модель, выявляющая причинно-следственные связи, присущие исследуемому объекту и существенные в рамках определённого исследования. Один и тот же объект может представляться различными концептуальными моделями, которые строятся в зависимости от цели исследования. Так, одна концептуальная модель может отображать временные аспекты функционирования системы, иная – влияние отказов на работоспособность системы.

**Математическая** модель – абстрактная модель, представленная на языке математических отношений. Она имеет форму функциональных зависимостей между параметрами, учитываемыми соответствующей концептуальной моделью. Эти зависимости конкретизируют причинно-следственные связи, выявленные в концептуальной модели, и характеризуют их количественно.

**Кибернетические** модели раскрывают внешние функциональные зависимости систем от среды, не затрагивая внутренних причинных связей. При таком моделировании в основном исследуется характеристика поведения сложной динамической системы в определённой среде. Моделирование в кибернетике обычно является вместе с тем и математическим моделированием, которое осуществляется с помощью аппарата математики и реализацией которого являются процедуры в современных вычислительных машинах и автоматах. Примеры кибернетических моделей: модели искусственного интеллекта, нейронные сети, роботы и т.д.

Все вышеперечисленные модели могут подразделяться:

- в зависимости от полноты описания системы – на полные, неполные, приближенные;
- по мере изменений во времени – на статические и динамические;

- по степени стабильности процессов – на детерминированные и стохастические.

**Полные модели** реализуют максимально возможное число свойств и элементов объекта. **Неполные модели** обладают только частью свойств оригинала. При **приближенном моделировании** учитываются только важнейшие аспекты системы.

**Статические модели** применяются для описания состояния системы в фиксированный момент времени, например, глобус. **Динамические модели** применяются для описания поведения систем во времени, например, звёздное небо в планетарии.

**Детерминированные модели** отражают процессы, для которых характерно отсутствие случайных воздействий, например, модель перемещения твёрдого тела.

**Стохастические** учитывают случайные процессы и события, например, модель броуновского движения.

## Лекция 6

### МЕТОДЫ МОДЕЛИРОВАНИЯ

#### Формализация моделей

Построение модели системы относится к числу системных задач, при решении которых синтезируют решения на базе огромного числа исходных данных. Использование системного подхода в этих условиях позволяет не только построить модель реального объекта, но и на базе этой модели выбрать необходимое количество управляющей информации в реальной системе, оценить показатели её функционирования и тем самым на базе моделирования найти наиболее эффективный вариант построения и оптимальный режим функционирования реальной системы.

Постановка любой задачи заключается в том, чтобы перевести её словесное, вербальное описание в формальное. В случае относительно простых задач строится формальная модель, которая представляет собой математическую зависимость между величинами в виде формулы, уравнения, системы уравнений. Если полученная формальная модель опирается на фундаментальный закон или подтверждается экспериментом, то этим доказывается её адекватность отображаемой ситуации, и модель рекомендуется для решения задачи.

По мере усложнения задач получение модели и доказательство её адекватности усложняется. Эксперимент может быть дорогим и опасным или практически нереализуемым (например, при создании сложных технических комплексов, при реализации космических программ и т.д.), вследствие чего задача переходит в класс проблем принятия решений, и перевод вербального описания в формальное становится важной составной частью процесса принятия решения. Для решения проблемы перевода вербального описания в формальное в различных областях деятельности стали развиваться специальные приёмы и методы. Так, возникли

методы типа "мозговой атаки", "сценариев", экспертных оценок, "дерева целей" и т.п.

В свою очередь, развитие математики шло по пути расширения средств постановки и решения трудноформализуемых задач. Наряду с детерминированными, аналитическими методами классической математики возникла теория вероятностей и математическая статистика (как средство доказательства адекватности модели на основе представительной выборки и понятия вероятности правомерности использования модели и результатов моделирования). Для задач с большей степенью неопределённости инженеры стали привлекать теорию множеств, математическую логику, математическую лингвистику, теорию графов, что во многом стимулировало развитие этих направлений. Иными словами, математика стала постепенно накапливать средства работы с неопределённостью, со смыслом, который классическая математика исключала из объектов своего рассмотрения.

Таким образом, между неформальным, образным мышлением человека и формальными моделями классической математики сложился как бы "спектр" методов, которые помогают формализовать вербальное описание проблемной ситуации, с одной стороны, и интерпретировать формальные модели, связывать их с реальной действительностью, с другой.

Первоначально исследователи, развивающие теорию систем, предлагали классификации систем и старались поставить им в соответствие определённые методы моделирования, позволяющие наилучшим образом отразить особенности того или иного класса. Такой подход к выбору методов моделирования подобен подходу прикладной математики. Однако в отличие от последней, в основу которой положены классы прикладных задач, системный анализ может один и тот же объект или одну и ту же проблемную ситуацию (в зависимости от степени неопределённости и по мере познания) отображать различными моделями, организуя, таким образом, как бы процесс постепенной формализации задачи, т.е. "выращивание" её формальной модели. Подход помогает понять, что неверно выбранный метод моделирования может привести к неверным результатам, к невозможности доказательства адекватности модели, к увеличению числа итераций и затягиванию решения проблемы.

### **Аналитические и статистические методы**

Аналитические и статистические методы получили наибольшее распространение в практике проектирования и управления.

Аналитическими названы методы, которые отображают реальные объекты и процессы в виде точек, совершающих какие-либо перемещения в пространстве или взаимодействующих между собой.

На базе аналитических представлений возникли и развиваются математические теории различной сложности – от аппарата классического математического анализа (методов исследования функций, их вида, способов представления, поиска экстремумов функций и т.п.) до таких новых разделов современной математики, как математическое программирование (линейное, нелинейное, динамическое и т.п.), теория игр (матричные игры с чистыми стратегиями, дифференциальные

игры и т.п.). Эти теоретические направления стали основой многих прикладных наук, в том числе теории автоматического управления, теории оптимальных решений и т.д.

Основу статистических представлений составляет отображение явлений и процессов с помощью случайных (стохастических) событий и их поведений, которые описываются соответствующими вероятностными (статистическими) характеристиками и статистическими закономерностями. Статистические отображения системы в общем случае (по аналогии с аналитическими) можно представить как бы в виде «размытой» точки (размытой области) в  $n$ -мерном пространстве. «Размытую» точку следует понимать как некоторую область, характеризующую движение системы (её поведение); при этом границы области заданы с некоторой вероятностью  $p$  («размыты») и движение точки описывается некоторой случайной функцией.

### **Случайные события и величины, их основные характеристики**

Величины, которые могут принимать различные значения в зависимости от внешних по отношению к ним условий, принято называть случайными (стохастическими по природе).

Для случайных величин (СВ) приходится использовать особые, статистические методы их описания. В зависимости от типа самой СВ – дискретная или непрерывная это делается по разному.

Дискретное описание заключается в том, что указываются все возможные значения данной величины и для каждой из них указывается вероятность или частота наблюдений именно этого значения при бесконечно большом числе всех наблюдений.

Доказано, что при увеличении числа наблюдений в определенных условиях за значениями некоторой дискретной величины частота повторений данного значения будет все больше приближаться к некоторому фиксированному значению, которое и есть вероятность этого значения.

Случайное событие с вероятностью 0.5 или 50% в 50 случаях из 100 может произойти или не произойти, если же его вероятность более 0.5 – оно чаще происходит, чем не происходит. События с вероятностью 1 называют достоверными, а с вероятностью 0 – невозможными.

Для случайного события  $X$  сумма вероятности  $P(X)$  (событие происходит) и  $P(\bar{X})$  (событие не происходит) равна 1.

Рассмотрим пример. Если мы наблюдаем за сложным событием, например, выпадением чисел 1..6 на верхней грани игральной кости, то можно считать, что такое событие имеет множество исходов и для каждого из них вероятность составляет  $1/6$  при симметрии кости. Если же кость несимметрична, то вероятности отдельных чисел будут разными, но сумма их равна 1.

Пусть в результате достаточно большого числа наблюдений за игрой с помощью одной и той же кости мы получили следующие данные:

Грани	1	2	3	4	5	6	Итого
Наблюдения	140	80	200	400	100	80	1000

Подобную таблицу наблюдений за СВ называют выборочным распределением. С помощью такой таблицы можно сосчитать среднее значение СВ или для нашего примера, сколько в среднем мы выигрываем за одно бросание кости, если выигрыш соответствует выпавшему числу на грани:

$$1 \cdot 0.140 + 2 \cdot 0.080 + 3 \cdot 0.200 + 4 \cdot 0.400 + 5 \cdot 0.100 + 6 \cdot 0.080 = 3.48$$

Если поставить вопрос иначе – оценить по этим данным наш будущий выигрыш, то ответ 3.48 принято называть **математическим ожиданием** случайной величины, которое в общем случае определяется как

$$M_X = \sum X_i \cdot P(X_i),$$

где  $P(X_i)$  – вероятность того, что  $X$  примет свое очередное значение.

Таким образом, математическое ожидание случайной величины (как дискретной, так и непрерывной) – это то, к чему стремится ее среднее значение при достаточно большом числе наблюдений.

Обращаясь к нашему примеру, можно заметить, что кость несимметрична, в противном случае вероятности составляли бы по 1/6 каждая, а среднее и математическое ожидание составило бы 3.5.

Поэтому уместен следующий вопрос: какова степень асимметрии кости, как ее оценить по итогам наблюдений?

Для этой цели используется специальная величина – мера рассеяния. Так же как мы "усредняли" допустимые значения СВ, можно усреднить ее отклонения от среднего. Но так как разности  $(X_i - M_X)$  всегда будут компенсировать друг друга, то приходится усреднять не отклонения от среднего, а квадраты этих отклонений. Величину

$$D_X = \sum (X_i - M_X)^2 \cdot P(X_i)$$

принято называть **дисперсией** случайной величины  $X$ .

Вычисление дисперсии намного упрощается, если воспользоваться выражением

$$D_X = \sum X_i^2 \cdot P(X_i) - M_X^2$$

т.е. вычислять дисперсию случайной величины через усредненную разность квадратов ее значений и квадрат ее среднего значения.

Выполним такое вычисление для нашего примера:

Грани (X)	1	2	3	4	5	6	Итого
$X^2$	1	4	9	16	25	36	
$P_i$	0.140	0.080	0.200	0.400	0.100	0.080	<b>1.000</b>
$P_i \cdot X^2 \cdot 1000$	140	320	1800	6400	2500	2880	<b>14040</b>

Таким образом, дисперсия составит  $14.04 - 3.48^2 = 1.930$ .

Заметим, что размерность дисперсии не совпадает с размерностью самой СВ и это не позволяет оценить величину разброса. Поэтому чаще всего вместо дисперсии используется квадратный корень из ее значения – т. н. **среднеквадратичное отклонение** или отклонение от среднего значения:

$$\sigma = \sqrt{D(X)}$$

составляющее в нашем случае = 1.389. Много это или мало?

В случае наблюдения только одного из возможных значений (разброса нет) среднее было бы равно именно этому значению, а дисперсия составила бы 0. И наоборот, если бы все значения наблюдались одинаково часто (были бы равновероятными), то среднее значение составило бы  $(1 + 2 + 3 + 4 + 5 + 6) / 6 = 3.500$ ; усредненный квадрат отклонения  $(1 + 4 + 9 + 16 + 25 + 36) / 6 = 15.167$ ; а дисперсия  $15.167 - 12.25 = 2.917$ .

Таким образом, наибольшее рассеяние значений СВ имеет место при ее равновероятном или равномерном распределении. Неслучайная, детерминированная величина имеет математическое ожидание равное ей самой, нулевую дисперсию и нулевой коэффициент вариации, в то время как равномерно распределенная СВ имеет максимальную дисперсию и максимальный коэффициент вариации.

В ряде ситуаций приходится иметь дело с непрерывно распределенными СВ – весами, расстояниями и т. п. Для них идея оценки среднего значения (математического ожидания) и меры рассеяния (дисперсии) остается той же, что и для дис-

кретных СВ. Приходится только вместо соответствующих сумм вычислять интегралы.

Математическое ожидание для непрерывных величин:

$$m_x = \int_{-\infty}^{\infty} p(x) dx;$$

Дисперсия для непрерывных величин:

$$\sigma_x^2 = \int_{i=1}^n (x_i - m_x)^2 p_i(x_i) .$$

Наиболее полной, исчерпывающей характеристикой случайной величины является закон распределения. **Закон распределения** – функция (таблица, график, формула), позволяющая определять вероятность того, что случайная величина  $X$  принимает определенное значение  $x_i$  или попадает в некоторый интервал. Если случайная величина имеет данный закон распределения, то говорят, что она распределена по этому закону или подчиняется этому закону распределения. Каждый закон распределения – это некоторая функция, полностью описывающая случайную величину с вероятностной точки зрения. На практике о распределении вероятностей случайной величины  $X$  часто приходится судить только по результатам испытаний. Эти распределения иногда называют «теоретическими», поскольку для них разработаны методы расчета всех показателей распределения, зафиксированы связи между ними, построены алгоритмы расчета и т. п.

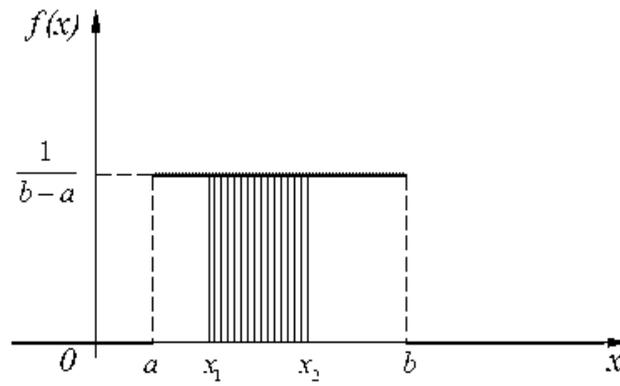
Типичные законы распределения:

**1. Равномерное распределение.** Непрерывная величина  $X$  *распределена равномерно* на интервале  $(a, b)$ , если все ее возможные значения находятся на этом интервале и плотность распределения вероятностей постоянна:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{при } x \in (a, b), \\ 0 & \text{при } x \notin (a, b). \end{cases}$$

где  $f(x)$  – функция распределения случайной величины;  $p(x)$  – плотность вероятностей распределения случайной величины.

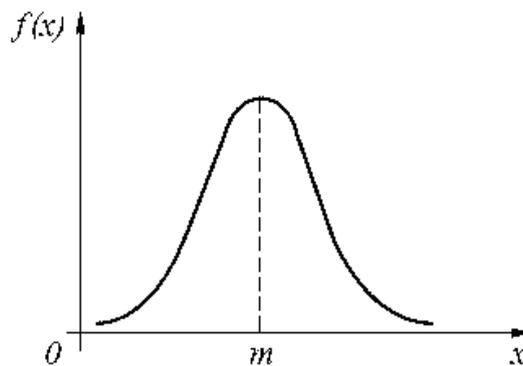
График плотности равномерного распределения:



**2. Нормальное (гауссово) распределение** – распределение с плотностью, описываемой формулой:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

График плотности нормального распределения:



Нормальное распределение является наиболее часто встречающимся в различных случайных явлениях природы. Так, ошибки выполнения команд автоматизированным устройством, ошибки вывода космического корабля в заданную точку пространства, ошибки параметров компьютерных систем и т.д. в большинстве случаев имеют нормальное или близкое к нормальному распределение. Более того, случайные величины, образованные суммированием большого количества случайных слагаемых, распределены практически по нормальному закону.

**3. Распределение Бернулли.** Если производится серия независимых испытаний, в каждом из которых случайное событие может появиться с одинаковой вероятностью, то число появлений события есть случайная величина, распределенная по закону Бернулли или по биномиальному закону. Другими словами, дискретная случайная величина  $X$  имеет *биномиальное распределение*, если ее возможные значения  $0, 1, 2, \dots, m, \dots, n$ , а соответствующие им вероятности равны:

$$P_m = C_n^m p^m q^{n-m},$$

где  $0 < p < 1$ ,  $q = 1 - p$ ;  $m = 0, 1, 2, \dots, n$ .

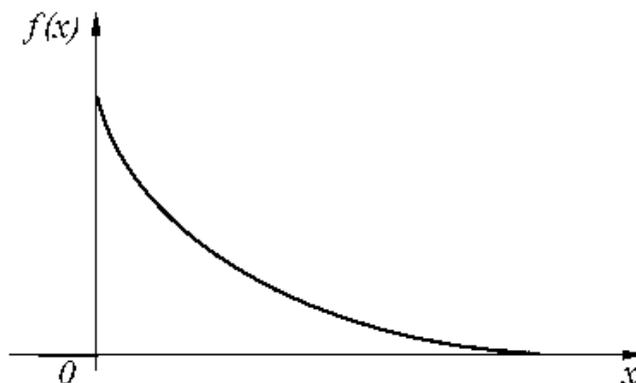
**4. Распределение Пуассона.** Дискретная случайная величина  $X$  имеет *распределение Пуассона*, если она имеет бесконечное счетное множество возможных значений  $0, 1, 2, \dots, m, \dots$ , а соответствующие им вероятности определяются формулой:

$$P_m = \frac{a^m}{m!} e^{-a}, \quad m = 0, 1, 2, \dots$$

**5. Показательное распределение.** Непрерывная случайная величина  $X$  имеет *показательное распределение*, если плотность распределения ее вероятностей выражается формулой:

$$f(t) = \begin{cases} \lambda \cdot e^{-\lambda t} & \text{при } t > 0, \\ 0 & \text{при } t \leq 0. \end{cases}$$

График плотности показательного распределения:



### Методы непараметрической статистики

Использование классических распределений случайных величин обычно называют параметрической статистикой, при этом интересующая нас СВ (дискретная или непрерывная) имеет вероятности, вычисляемые по некоторым формулам или алгоритмам. Однако не всегда полученные экспериментальные данные подчиняются стандартным законам распределения.

В настоящее время в прикладной статистике все большей популярностью пользуются методы т. н. непараметрической статистики, когда вопрос о принадлежности распределения вероятностей данной величины к тому или иному классу

вообще не подымается, но задача оценки самой СВ, получения информации о ней остается.

Непараметрические методы позволяют обрабатывать данные "низкого качества" из выборок малого объёма с переменными, про распределение которых мало что или вообще ничего неизвестно.

Непараметрические методы не основываются на оценке параметров (таких как среднее или стандартное отклонение) при описании выборочного распределения интересующей величины. Поэтому эти методы иногда также называются свободными от параметров или свободно распределенными.

Наиболее часто встречается задача проверки непараметрических гипотез. Под статистической гипотезой понимается любое предположение о свойствах случайных величин или событий.

Основные типы гипотез:

1. Гипотезы о типах вероятностных законов распределения случайных величин.

2. Гипотезы о свойствах тех или иных числовых параметров (медиан, дисперсий и т.д.), характеризующих случайные величины.

3. Гипотезы о равенстве или различии законов распределения случайных величин, характеризующих изучаемое свойство в двух или более совокупностях рассматриваемых явлений.

Прежде чем приступить к проведению эксперимента, исследователь обычно выдвигает две взаимоисключающие гипотезы. Одна из них является статистической гипотезой, которую исследователь обычно предполагает отклонить, и называется **нулевой гипотезой** ( $H_0$ ). Проверка нулевой гипотезы обычно осуществляется путем сравнения ее с другой гипотезой, которая называется альтернативной. **Альтернативная гипотеза** ( $H_1$ ) фактически отрицает нулевую гипотезу.

В математической статистике проверка тех или иных гипотез о случайных величинах или событиях основана на принципе практической невозможности, который заключается в следующем. Задается заранее некоторая, обычно весьма малая вероятность  $\alpha$  (например,  $\alpha=0,1$ ;  $\alpha=0,05$  и т.д.), именуемая уровнем значимости. При этом случайные события, вероятность которых меньше или равна  $\alpha$ , считаются практически невозможными. Следовательно, если некоторое событие с вероятностью меньшей или равной  $\alpha$  все-таки произошло, то это событие следует рассматривать как неслучайное, значимое. Уровень значимости показывает, сколько раз в среднем можно ошибиться, объявив изучаемое событие неслучайным. Так, если уровень значимости выразить в процентах, то  $\alpha=0,05$  допускает в среднем пятипроцентную ошибку, т.е. в 5 случаях из 100.

Всякое правило, на основе которого нулевая гипотеза отклоняется или принимается, называется критерием для проверки этой гипотезы, например, критерий Макнамары, критерий Вилкоксона, критерий знаков, медианный критерий, критерий Колмогорова-Смирнова и др.

## Лекция 7

### МЕТОДЫ ФОРМАЛИЗАЦИИ МОДЕЛЕЙ

#### Теоретико-множественные представления

Теоретико-множественные представления базируются на понятиях множество, элементы множества, отношения на множествах. Понятие множество относится к числу интуитивно постигаемых понятий, которым трудно дать определение. Это понятие содержательно эквивалентно понятиям "совокупность", "собрание", "ансамбль", "коллекция", "семейство", "класс" и другим обобщающим понятиям. Один из основоположников теории множеств Георг Кантор определял множество как "многое, мыслимое нами как единое". Множества могут задаваться следующими способами:

1) списком, перечислением (интенциональным путём); например:

$$\{a_i\} \in A, \text{ где } i = 1, \dots, n,$$

или

$$A = \langle a_1, a_2, \dots, a_i, \dots, a_n \rangle,$$

где  $\in$  – знак вхождения элементов в множество;

2) путём указания некоторого характеристического свойства  $A$  (экстенционально). Например, "множество натуральных чисел", "множество рабочих данного завода", "множество планет солнечной системы", "множество  $A$ " и т.д.

В множестве могут быть выделены подмножества. Вхождение элементов в любое множество или подмножество описывается знаком принадлежит –  $\in$ , а вхождение подмножества в множество записывается  $A \subset B$ . Это означает, что все элементы подмножества  $A$  являются одновременно элементами множества  $B$ .

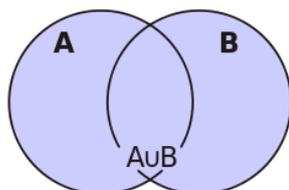
Важным понятием является понятие пустое множество – множество, в котором в данный момент нет ни одного элемента:  $D = \emptyset$ .

При использовании теоретико-множественных представлений в соответствии с концепцией Кантора можно вводить любые отношения. При уточнении этих отношений применительно к множествам удобно пользоваться наглядными диаграммами Эйлера-Венна.

Примеры теоретико-множественных отношений (операции над множествами):

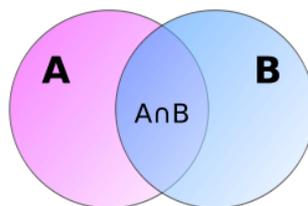
**Объединение множеств** – множество элементов, принадлежащих либо  $A$ , либо  $B$ , либо как  $A$ , так и  $B$ .

$$A \cup B = \{x / x \in A \text{ или } x \in B\}$$



**Пересечение множеств** – множество всех элементов, принадлежащих как А, так и В.

$$A \cap B = \{x / x \in A \text{ и } x \in B\}$$

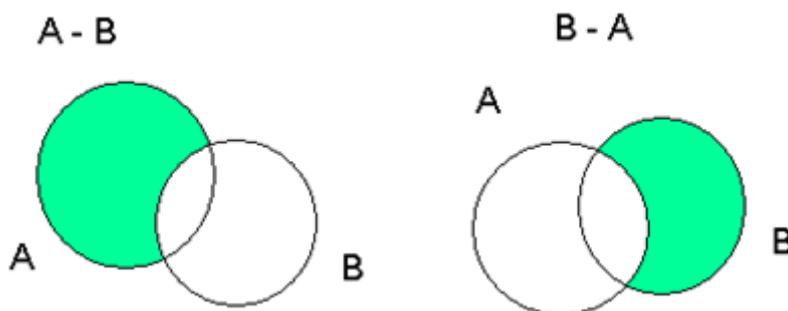


**Вычитание А-В** – множество элементов А, не принадлежащих В.

$$A \setminus B = \{x \in A \text{ и } x \notin B\}$$

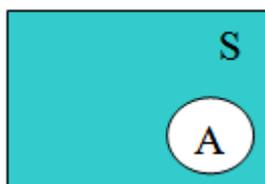
**Вычитание В-А** – множество элементов В, не принадлежащих А.

$$B \setminus A = \{x \in B \text{ и } x \notin A\}$$



Если  $A \subset S$ , то разность  $S \setminus A$  называется **дополнением** множества А до множества S.

$$\bar{A} = \{x | x \in S \text{ и } x \notin A\}$$



### Математическая логика

Формальная логика – одна из древнейших наук. Отдельные фрагменты ее начали разрабатываться в VI в. до нашей эры в Древней Греции и Индии. Основателем этой науки считается гениальный древнегреческий ученый Аристотель, который обстоятельно систематизировал логические формы и правила мышления, и

тем самым заложил начала логики. Логика, основанная на учении Аристотеля, существовала до начала XX в., после чего в ней произошла своеобразная научная революция, связанная с широким применением методов так называемой символической, или математической логики. Идеи последней – о возможности сведения рассуждений к вычислениям – были высказаны еще немецким ученым Г. В. Лейбницем в XVII в. Однако только к началу XX столетия математическая логика (то есть логика, развиваемая математическим методом) оформилась в качестве самостоятельной дисциплины. Характерным для этой дисциплины является использование формальных языков с точным синтаксисом и четкой семантикой, однозначно определяющими понимание формул.

*Семантика* (от др.-греч. σηματικός – обозначающий) – раздел лингвистики, изучающий смысловое значение единиц языка.

Современная математическая логика – это та же самая логика Аристотеля, но только громоздкие словесные выводы заменены в ней математической символикой. Эта дисциплина изучает вопросы применения математических методов для решения логических задач.

Базовыми понятиями математической логики являются высказывание, предикат, логические функции (операции) кванторы, логический базис, логические законы (законы алгебры логики). Под высказыванием в алгебре логики понимается повествовательное предложение (суждение), которое характеризуется определённым значением истинности.

В простейших случаях используется два значения истинности: "истинно" – "ложно", "да" – "нет", "1" – "0". Такая алгебра логики, в которой переменная может принимать только два значения истинности, называется бинарной алгеброй логики или Булевой алгеброй (по имени создателя алгебры логики).

В современной математической логике существуют два подхода: алгебра логики и логические исчисления.

### **Логика (исчисление) высказываний**

Логика высказываний является развитием алгебры логики. Логическими высказываниями являются утвердительные предложения, о которых можно судить, истинны они или ложны. Причем они не могут быть истинными и ложными одновременно. Пример утверждения, не являющегося высказыванием:

**Все, что написано в этой рамке, есть ложь.**

Попытка определить истинностное значение этого "высказывания" приводит к противоречию: если то, что написано, истинно, то это противоречит смыслу слов в рамке. То же противоречие возникает, если предположить, что оно ложно.

Вопросительные, повелительные и бессмысленные предложения не являются логическими высказываниями.

В логике высказываний каждое высказывание рассматривается как двоичная переменная, которая удовлетворяет закону исключения третьего: каждое высказывание может быть истинным или ложным, третьего не дано. При этом счи-

тают, что высказывание не может быть одновременно и истинным и ложным (закон противоречия).

По аналогии с элементарной алгеброй, где любое число является константой, высказывание является логической константой, величина которой равна 1 или 0.

В логике высказываний используют сентенциальные связки: отрицание, конъюнкция, дизъюнкция, импликация, эквиваленция и неравнозначность, с помощью которых из простых высказываний образуются сложные.

Сентенция (от лат. *sententia*, буквально *мнение, суждение*) – изречение нравоучительного характера.

Высказывание называется простым, если оно рассматривается как некое неделимое целое. Сложным называется высказывание, составленное из простых с помощью логических связок.

### Сентенциальные связки логики высказываний

**1. Отрицание (логическая связка "не").** Отрицанием (инверсией) высказывания  $A$  называется высказывание, которое истинно, если высказывание  $A$  ложно, и ложно, когда  $A$  истинно. Записывается:  $\bar{A}$  или  $\neg A$ . Читается: "не  $A$ " ("не верно, что  $A$ ").

Таблица истинности:

$A$	$\neg A$
0	1
1	0

**2. Логическое умножение (конъюнкция).** Конъюнкция двух высказываний  $A$  и  $B$  – это сложное логическое высказывание, которое истинно только в случае истинности всех составляющих высказываний, в противном случае оно ложно. Обозначения:  $A \& B$ ,  $A \wedge B$ . Читается: "А и В".

Таблица истинности:

$A$	$B$	$A \& B$
0	0	0
0	1	0
1	0	0
1	1	1

**3. Логическое сложение (дизъюнкция).** Дизъюнкция двух высказываний  $A$  и  $B$  – это сложное логическое высказывание, которое ложно только в случае ложности всех составляющих высказываний, в противном случае оно истинно. Таким образом, это высказывание считается истинным, когда истинно хотя бы одно из

составляющих высказываний. Обозначается:  $A \vee B$ . Иногда встречается обозначение  $A + B$ . Читается: "А или В".

Таблица истинности:

$A$	$B$	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

**4. Логическое следование (импликация).** В математических доказательствах часто пользуются сложными высказываниями, образованными с помощью слов "если..., то...". Здесь высказывание, расположенное после слова "если", называется основанием или посылкой, а высказывание, расположенное после слова "то", называется следствием или заключением. Импликацией двух высказываний  $A$  и  $B$  называется высказывание, обозначаемое символом  $A \rightarrow B$ , которое ложно тогда и только тогда, когда  $A$  истинно, а  $B$  ложно. Иногда встречается обозначение  $A \supset B$ . Читается: "если  $A$ , то  $B$ " ("А влечет В", "из А следует В").

Таблица истинности:

$A$	$B$	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

**5. Логическое тождество (эквиваленция).** Эквиваленцией (эквивалентностью, равнозначностью) двух высказываний  $A$  и  $B$  называется высказывание, обозначаемое символом  $A \sim B$  (или  $A \leftrightarrow B$ ), которое истинно когда истинностные значения высказываний  $A$  и  $B$  совпадают, и ложно – в противном случае.

Таблица истинности:

$A$	$B$	$A \sim B$
0	0	1
0	1	0
1	0	0
1	1	1

**6. Исключающее "или" (неравнозначность).** Неравнозначностью двух высказываний  $A$  и  $B$  называется высказывание, истинное, когда истинностные значения  $A$  и  $B$  не совпадают, и ложное – в противном случае. Обозначается:  $A \oplus B$ . Читается: "либо  $A$ , либо  $B$ " (понимается – в разделительном смысле).

Таблица истинности:

$A$	$B$	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Примеры высказываний:

Примеры высказываний:

$P$  = "Студенты на занятиях обучают преподавателя"

$P = 0, \bar{P} = 1;$

$R$  = "Преподаватель создал электронные средства обучения"

$Q$  = "Преподаватель использует электронные средства обучения в педагогической практике"

$R \wedge Q$  = "Преподаватель создал электронные средства обучения и использует их в педагогической практике"

$R$  = "Преподаватель способен провести занятие традиционным методом"

$Q$  = "Преподаватель способен провести занятие инновационным методом"

$R \vee Q$  = "Преподаватель способен провести занятие традиционным методом или инновационным методом"

$P$  = "Студент хорошо подготовился к экзамену"

$Q$  = "Студент обязательно получит положительную оценку"

$P \rightarrow Q$  = "Если студент хорошо подготовился к экзамену, то студент обязательно получит положительную оценку"

### Формулы и функции алгебры высказываний

Высказывания представляют собой константы или переменные и обозначаются, как правило, прописными буквами, а для операций используются те же символы, что и в алгебре логики.

Высказывательными переменными называются такие переменные, вместо которых можно подставлять конкретные высказывания. С помощью высказывательных переменных можно построить разнообразные логические формулы.

Определения логической формулы:

1) Всякая высказывательная переменная есть формула.

2) Если  $A$  – формула, то и  $\neg A$  является формулой.

3) Если  $A$  и  $B$  – формулы, то выражения  $(A \& B)$ ,  $(A \vee B)$ ,  $(A \rightarrow B)$ ,  $(A \sim B)$ ,  $(A \oplus B)$  также являются формулами.

4) Других формул, кроме построенных по правилам трех предыдущих пунктов, нет.

Примеры:

$A = \text{"Войтенко – студент ХНАДУ"}$

$B = \text{"Войтенко – студент первого курса"}$

$A \wedge B = \text{"Войтенко – студент первого курса ХНАДУ"}$

$A = \text{"сегодня суббота"}$

$B = \text{"сегодня воскресенье"}$

$A \oplus B = \text{"сегодня суббота или воскресенье"}$

$A = \text{"идет снег"}$

$B = \text{"идет дождь"}$

$A \vee B = \text{"идет снег или дождь"}$

В алгебре высказываний каждая логическая формула задает логическую функцию – функцию от логических переменных, которая сама может принимать только два логических значения: 1 (истина) или 0 (ложь). Логические функции широко используются при описании систем, явлений, формализации рассуждений и пр.

### Логика (исчисление) предикатов

Предикат – это высказывание, содержащее одну или несколько неизвестных, другими словами, в предикат можно подставлять аргументы. Если предикат содержит одну неизвестную – он является одноместным.

Одноместным предикатом называется функция одной переменной, значениями которой являются высказывания об объектах, представляющих значения аргумента. Следовательно, одноместный предикат  $P(x)$  – это произвольная функция переменной  $x$ , определенная на некотором множестве  $M$  и принимающая логические значения из множества  $\{0, 1\}$ . Множество  $M$ , на котором определен предикат  $P(x)$ , называется предметной областью или областью определения предиката, а сама переменная  $x$  – предметной переменной.

Таким образом, одноместный предикат  $P(x)$  – это утверждение об объекте  $x$ , где  $x$  рассматривается как переменная.

Чтобы задать предикат от  $n$  аргументов ( $n$ -местный предикат), прежде всего следует указать множества  $M_1, \dots, M_n$  – области изменения предметных переменных  $x_1, \dots, x_n$ . При этом  $n$ -местным предикатом называется произвольная функция  $n$  переменных  $P(x_1, \dots, x_n)$ , определенная на множестве  $M = M_1 \times \dots \times M_n$  и принимающая логические значения из множества  $\{0, 1\}$ .

Например:

" $x$  – студент" – одноместный предикат  $P(x)$ ;

" $x$  учится в группе  $y$ " – двуместный предикат  $P(x, y)$ ;

" $x$  имеет отличные знания по дисциплине  $y$ , которую преподает  $z$ " – трехместный предикат  $P(x, y, z)$  и т.д.

В случае если аргументы (предметные переменные) замещены конкретными значениями (предметными постоянными), например, фамилия студента, фамилия преподавателя, название учебной дисциплины и другими, то предикат вырождается в высказывание, которое рассматривается как 0-местный предикат.

Предикаты, как и булевы переменные, можно связывать логическими операциями и получать более сложные предикаты. Кроме того, в исчисление предикатов введены операции, которые называются кванторами. Квантор – это общее название для логических операций, ограничивающих область истинности какого-либо предиката. В математической логике наиболее употребительны квантор всеобщности  $\forall$  и квантор существования  $\exists$ .

Пусть  $P(x)$  – одноместный предикат, определенный на множестве  $M$ . Под выражением  $\forall x P(x)$  понимают высказывание, истинное, если  $P(x)$  истинно для каждого элемента  $x \in M$ , и ложное в противном случае. Иными словами, истинность высказывания  $\forall x P(x)$  означает, что область истинности предиката  $P(x)$  совпадает с областью изменения переменной  $x$ . Читается это высказывание: "для всякого  $x$  истинно  $P(x)$ ".

Под выражением  $\exists x P(x)$  понимают высказывание, истинное, если существует  $x \in M$ , для которого  $P(x)$  истинно, и ложное в противном случае. Иными словами, истинность высказывания  $\exists x P(x)$  означает, что область истинности предиката  $P(x)$  не пуста. Читается это высказывание: "существует  $x$ , при котором  $P(x)$  истинно".

Например:

Предикат  $P(x) = "x - студент отличник ХНАДУ"$  определен на множестве студентов ХНАДУ. Подставляя вместо  $x$  фамилии отличников, получим множество высказываний типа  $P(x) = "Петренко - отличник ХНАДУ"$ ,  $P(x) = "Войтенко - отличник ХНАДУ"$  и др., которые являются истинными. Высказывание  $\forall x P(x) = "все студенты являются отличниками ХНАДУ"$  – ложно, а  $\exists x P(x) = "некоторые студенты являются отличниками ХНАДУ"$  – истинно.

Выражения, которые можно записать с применением к предикатам sentенциальных связок и кванторов, представляют собой формулы логики предикатов.

## Лекция 8

### ГРАФИЧЕСКИЕ МЕТОДЫ ФОРМАЛИЗАЦИИ МОДЕЛЕЙ

Графические представления позволяют наглядно отображать структуры сложных систем и процессов, происходящих в них. На основе графических представлений возникли методы, которые позволяют ставить и решать вопросы моделирования процессов организации, управления, проектирования и т.д. К графическим методам относятся традиционная геометрия, теория графов, методы сетевого планирования и управления, семантические сети и др.

## Методы сетевого планирования и управления

В анализе хозяйственной деятельности используется метод сетевого планирования.

**Система сетевого планирования и управления (СПУ)** – совокупность научно обоснованных положений организации и управления производством, основанной на моделировании процесса с помощью сетевого графика на базе применения теории графов, теории вероятностей и компьютерных технологий.

Система СПУ позволяет формировать календарный план реализации сложного комплекса работ, определять и мобилизовать резервы времени, предупреждать возможные срывы в ходе работ, осуществлять оперативную корректировку планов.

Первоначально разработка СПУ вызывалась необходимостью обоснованного прогнозирования срока окончания крупных бизнес-проектов, однако по мере развития этих систем и компьютерных технологий они стали применяться для решения значительно более широкого круга задач. Будучи эффективным средством планирования и управления, сетевые методы вместе с тем отличаются простотой и доступностью, что в немалой степени способствовало их быстрому освоению на практике. В настоящее время возможно применение СПУ как в форме однократного использования сетевых методов и моделей, так и в форме постоянно действующей системы СПУ как составной части более сложных систем управления.

СПУ позволяет:

- сформировать календарный план реализации сложного бизнес-проекта;
- определить и мобилизовать резервы времени, материальных, финансовых, информационных, трудовых ресурсов;
- осуществить реализацию логистического принципа "точно в срок" с прогнозированием и предупреждением возможных срывов в ходе реализации проекта;
- производить оперативную реализацию бизнес-проекта;
- повышать эффективность менеджмента при четком распределении ответственности между руководителями разного уровня и исполнителями и необходимым делегировании полномочий.

Особенностью методов СПУ является не только моделирование всего комплекса работ, но и выявление тех участков, от которых в наибольшей степени зависит выполнение всего бизнес-проекта в установленные сроки. Этот метод учитывает все многообразие связей между отдельными работами, позволяет оценить влияние отклонения от плана на дальнейший ход работы и способствует оптимизации процесса управления всем ходом работ.

Основным элементом системы СПУ является сетевая модель, отображающая с любой степенью детализации план выполнения некоторого комплекса взаимосвязанных работ, заданного в специфической форме сети, наглядное изображение которой представляет собой сетевой график. **Сетевым графиком** называется наглядное изображение последовательности и взаимной логической связи всех работ, выполняемых в процессе разработки и получаемых при этом результа-

тов, вплоть до достижения конечной цели. С помощью сетевых графиков достигается либо оптимизация времени выполнения, либо оптимизация величины себестоимости осуществляемых работ.

ГОД	14	14	14	14	15	15	16	16	17	17	17	18	18
Кол-во мес.	1	3	2	6	6	6	6	6	5	1	6	6	6
Накопленное время, мес.	1	4	6	12	18	24	30	36	41	42	48	54	60
ЭТАП 1	■												
ЭТАП 2		■											
ЭТАП 3			■										
ЭТАП 4		■											
ЭТАП 5		■	■	■	■	■	■						
ЭТАП 6		■			■		■						
ЭТАП 7				■	■								
ЭТАП 8				■	■								
ЭТАП 9						■	■						
ЭТАП 10							■	■					
ЭТАП 11									■				
ЭТАП 12										■		■	
<b>ИТОГО</b>										<b>41 месяц 1 неделя</b>			

Всем моделям свойственны общие принципы:

- по каждому объекту составляются сетевые графики – условные экономико-математические модели, отражающие весь ход выполнения работ от начала до завершения;

- сроки проведения работ по отдельным этапам определяются исходя из конечного срока;

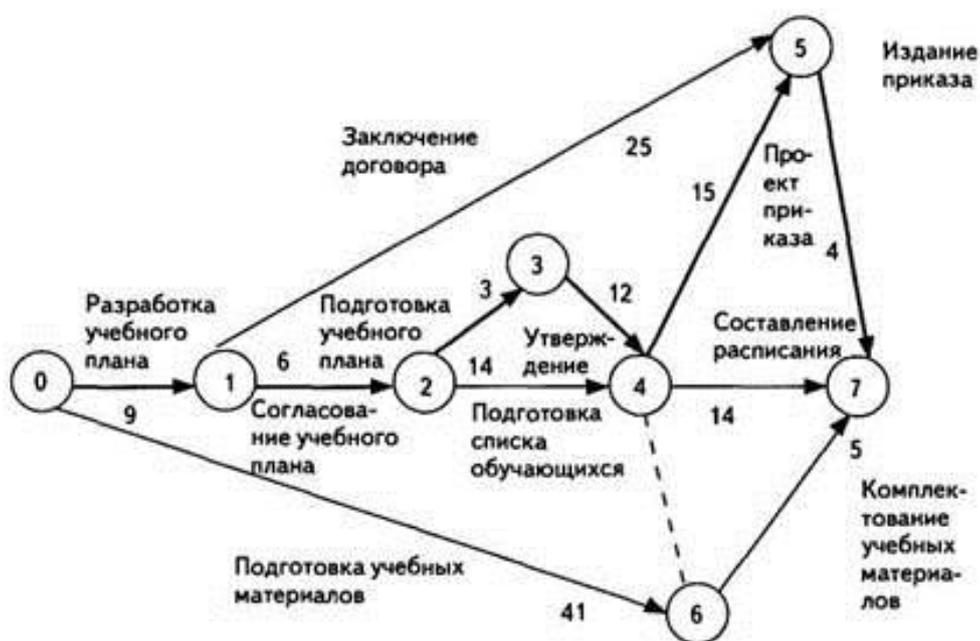
- при составлении сетевого графика используются следующие исходные материалы: задание на проектирование, проектно-конструкторская документация, проекты производства работ, действующие технологические процессы, графики поставок ресурсов, оборудования, документации.

Главными элементами сетевого графика являются понятия событие и работа. Термином "работа" обозначается совокупность приемов и действий, необходимых для выполнения конкретной задачи или достижения определенной цели. Работа подразделяется на работу-действие, работу-ожидание и зависимость (фиктивную работу).

Работа-действие – процесс, происходящий во времени, и требующий затрат ресурсов (материальных, информационных, финансовых, трудовых). Каждая работа-действие конкретна, определена, имеет ответственного исполнителя. Она переводит одно событие в другое и на сетевом графике изображается сплошной линией со стрелкой. Примеры подобной работы: закупка материальных ресурсов, изготовление конечной продукции, испытание конструкции.

Работа-ожидание – процесс, происходящий во времени, но не требующий ресурсных затрат. Работа-ожидание переносит событие во времени и на сетевом графике также изображается сплошной линией со стрелкой. К таким работам относятся процесс сушки изделия естественным путем после покраски, твердение бетона при строительных работах.

Зависимость (фиктивная работа) показывает логическую связь между двумя или несколькими событиями; не требует ресурсных и временных затрат, но указывает на то, что возможность начала одной работы непосредственно зависит от результатов другой. Ее продолжительность принимается равной нулю и на сетевом графике она изображается пунктирной линией со стрелкой.



Под событием понимают результат работы, без которого не могут быть начаты другие работы. Иначе говоря, событие может свершиться только тогда, когда закончатся все работы, ему предшествующие, а последующие работы могут начаться только тогда, когда событие свершится. Для всех непосредственно следующих за ним работ событие является начальным или предшествующим, а для всех непосредственно предшествующих ему работ – конечным или последующим. В сетевых графиках события обозначаются кружками, где внутри пишется номер. Стрелки, помещающиеся между кружками, выражают намеченную последовательность выполнения работ.

Если в сетевой модели нет числовых оценок, то такая сеть называется структурной. Однако чаще всего используются сети, в которых заданы оценки продолжительности работ (указываемые в часах, неделях, месяцах и т.д. над соответствующими стрелками), а также оценки других показателей (трудоемкости, стоимости). Ориентация и размеры стрелок (топология сети) принципиального значения не имеют, так же как сетевой график не имеет масштаба.

При построении сетевого графика необходимо соблюдать следующие правила:

- 1) только исходные события не имеют входящих стрелок, т.е. не должно быть событий (кроме исходного), которым не предшествует хотя бы одна работа;
- 2) только конечные события не имеют выходящих стрелок, т.е. не должно быть событий, из которых не выходит ни одна работа, за исключением завершающего;
- 3) каждая работа должна иметь предшествующее и последующее события;
- 4) не должно быть контуров и петель, соединяющих события с ними же самими, так как это означает, что условием начала некоторой работы является ее же окончание;
- 5) любые два события должны быть непосредственно связаны не более чем одной работой.

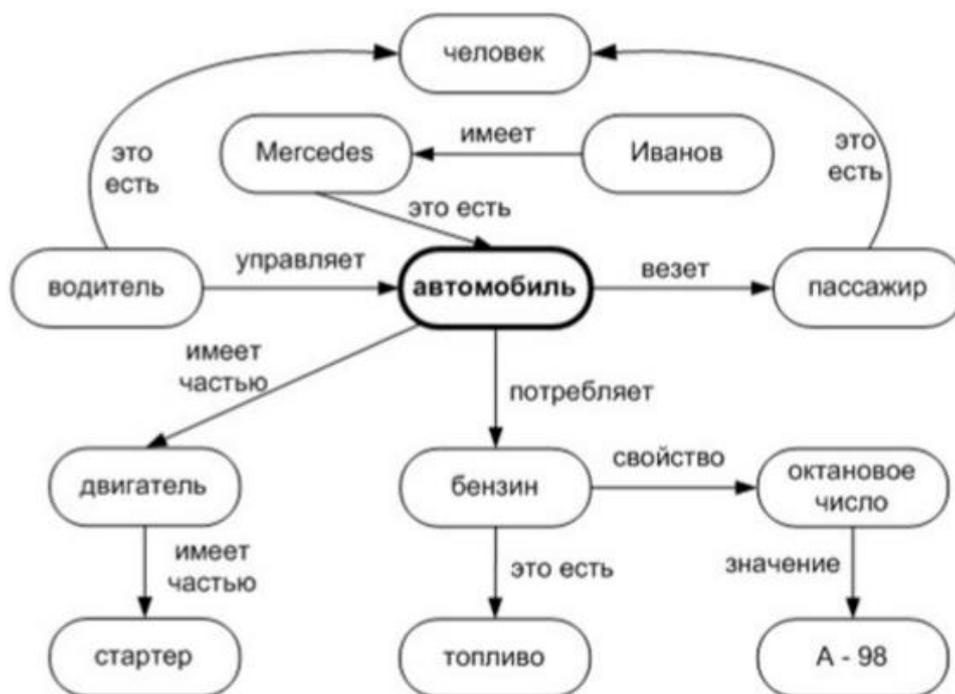
### **Семантические сети**

**Семантика** – раздел языкознания, изучающий значение единиц языка, прежде всего его слов и словосочетаний. В более общем смысле, семантика определяет смысл знаков (обозначений) и их сочетаний.

**Семантическая сеть** – информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (рёбра) задают отношения между ними.

Семантические сети получили свое развитие в рамках научного направления, связанного с представлением знаний для моделирования рассуждений человека. Эта область научных исследований возникла в рамках общей проблематики искусственного интеллекта и была ориентирована на разработку специальных языков и графических средств для представления декларативных знаний о предметной области. Результаты исследований в области семантических сетей в последующем были конкретизированы и успешно использованы при построении концептуальных моделей и схем реляционных баз данных.

В общем случае под семантической сетью понимают некоторый граф, в котором множество вершин и множество ребер разделены на отдельные типы, обладающие специальной семантикой, характерной для той или иной предметной области. Характерной особенностью для семантических сетей является то, что они для образования своей структуры используют два компонента – понятия и отношения. Вершинам сети соответствуют понятия объекты, события, процессы, явления), а дугам, их соединяющим, – отношения между понятиями.



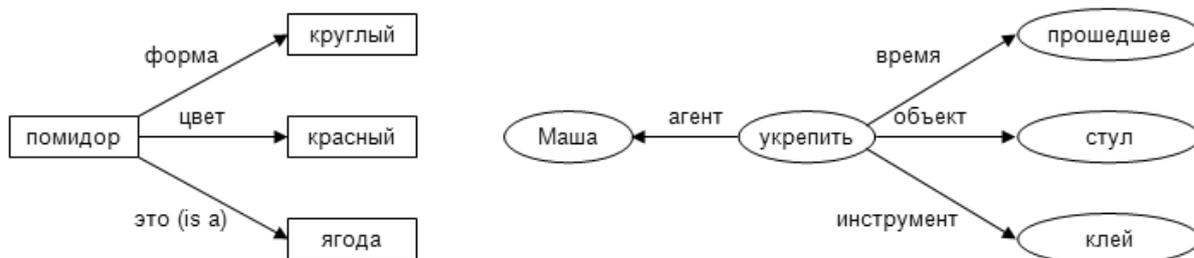
В качестве отношений наиболее часто используются следующие:

- таксономические ("класс – подкласс – экземпляр", "множество – подмножество – элемент" и т.п.);
- структурные ("часть" – целое");
- родовые ("предок" – "потомок");
- производственные ("начальник" – "подчиненный");
- функциональные (определяемые обычно глаголами "производит", "влияет" и т.п.);
- количественные (больше, меньше, равно и т.п.);
- пространственные (далеко от, близко от, за, под, над и т.п.);
- временные (раньше, позже, в течение и т.п.);
- атрибутивные (иметь свойство, иметь значение);
- логические (И, ИЛИ, НЕ);
- казуальные (причинно-следственные).

Отношения можно также классифицировать по арности (степени участия понятий в отношениях):

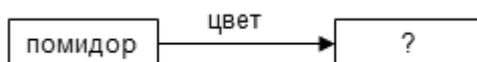
- унарное (рекурсивное) – отношение связывает понятие само с собой;
- бинарное – отношение связывает два понятия;
- N-арное – отношение, связывающее более двух понятий.

Приведем пример двух простых семантических сетей. Одна из них описывает понятие "помидор", а другая описывает факт "Маша укрепила стул клеем".



Важной особенностью семантических сетей является разработка специальных графических обозначений для представления отдельных типов вершин и ребер. При этом вершины не изображаются, как ранее – точками, а имеют вид прямоугольников, овалов, окружностей и других геометрических фигур, конкретный вид которых определяет тот или иной тип сущностей предметной области. Более разнообразным становится и изображение ребер, приобретающих вид различных линий со стрелками или без них, а также имеющих специальные обозначения в виде условных значков.

Проблема поиска решения в семантической сети сводится к задаче поиска фрагмента сети, соответствующего поставленному запросу. Например, вопрос "Какого цвета помидор?" можно графически представить в виде подсети.



Наложение подсети вопроса на сеть, описывающую предметную область, дает ответ – "красный".

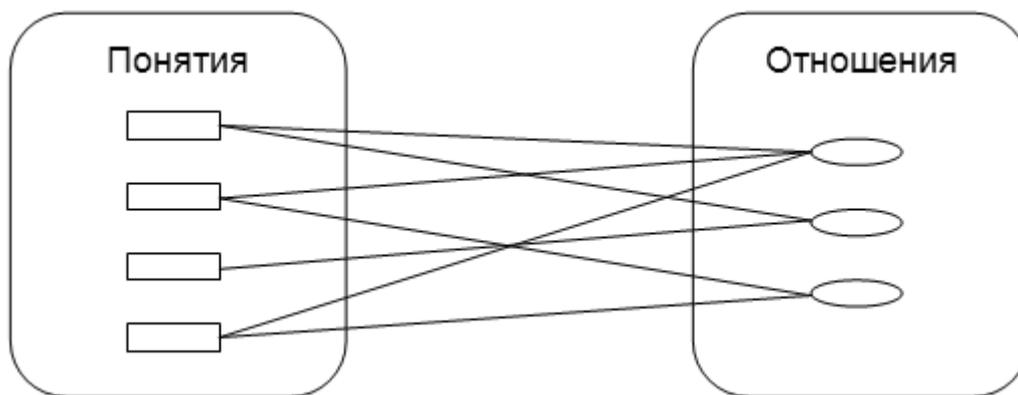
Семантические сети широко используются в экспертных системах в качестве языка представления знаний, в системах распознавания речи и понимания естественного языка.

### Концептуальные графы

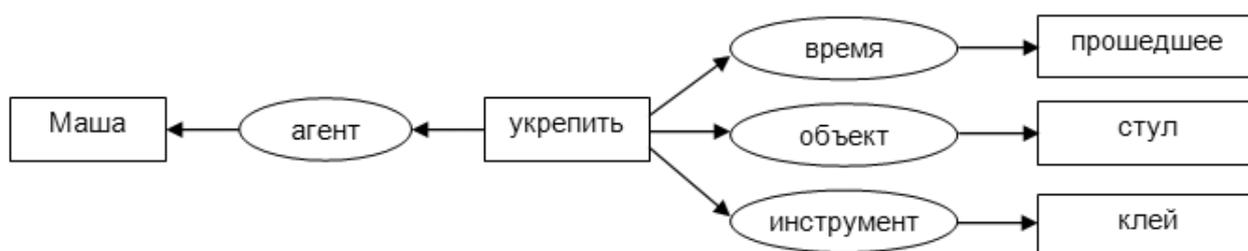
Дальнейшим развитием графовых структур, моделирующих семантику естественного языка, являются концептуальные графы.

**Концептуальный граф** – это двудольный ориентированный граф, состоящий из вершин двух типов: понятий и концептуальных отношений.

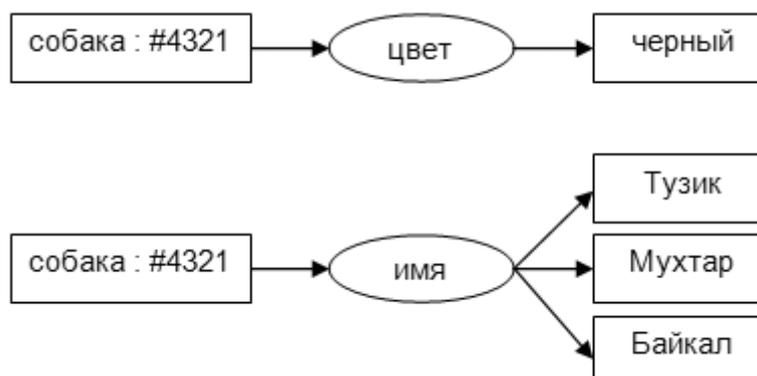
**Двудольный граф** – это граф, множество вершин которого можно разбить на две части таким образом, что каждое ребро графа соединяет какую-то вершину из одной части с какой-то вершиной другой части, то есть не существует ребра, соединяющего две вершины из одной и той же части.



Понятия в концептуальных графах отображаются прямоугольниками, отношения между ними – эллипсами. В отличие от семантических сетей отношение между понятиями отображаются не именованной дугой графа, а вершиной соответствующего типа, которая связывает два понятия дугами без метки. Рассмотренный выше пример "Маша укрепила стул клеем" в виде концептуального графа будет выглядеть следующим образом.

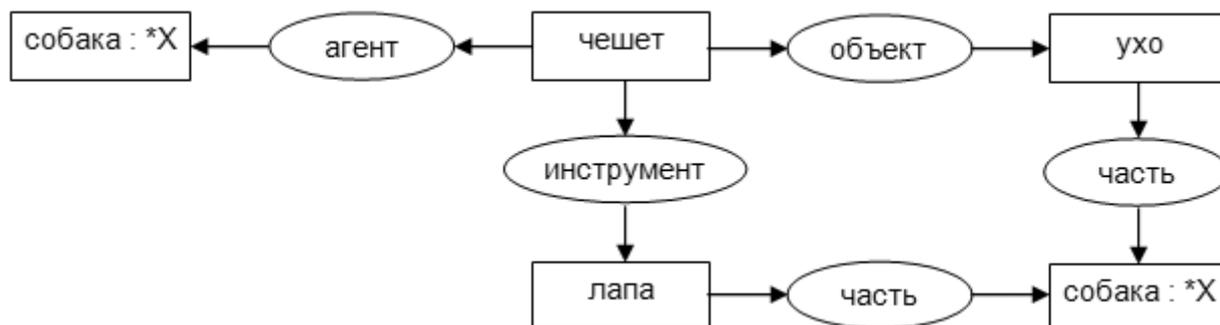


При описании конкретных, но неименованных (анонимных) экземпляров, используется маркер #. Использование анонимных экземпляров позволяет упростить описание и представление предметной области (базы знаний), как набора концептуальных графов. В частности, факт, что "три собаки Тузик, Мухтар и Байкал черного цвета" можно описать вместо одного графа двумя.



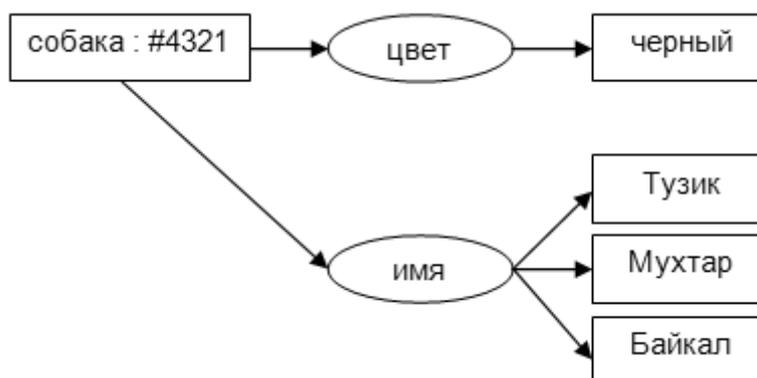
Вместо обращения к экземпляру по имени можно использовать маркер \*, обозначающий любой экземпляр. Дополнительно к обобщающему маркеру \* допускается использование переменных для более упрощенного и наглядного отобра-

ражения графов. Например, факт "Собака чешет лапой ухо" может выглядеть следующим образом.



Теория концептуальных графов предусматривает четыре вида операций, позволяющие создавать новые графы на основе существующих. К ним относятся:

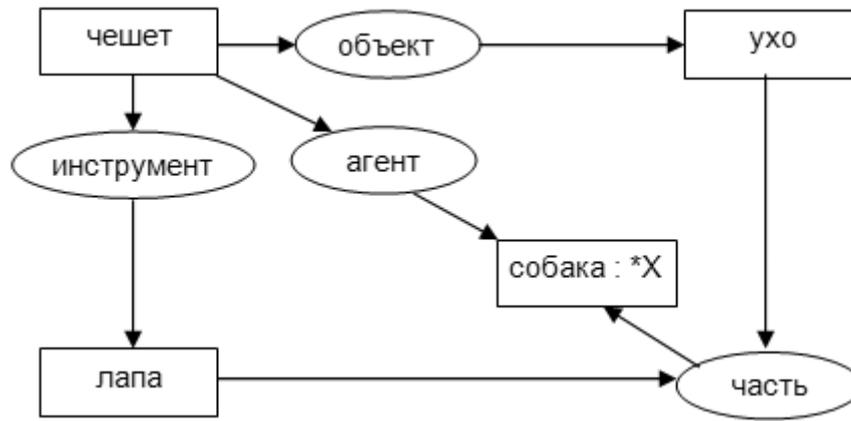
- копирование – создание точной копии какого-либо графа;
- объединение двух графов в один, если они имеют семантически общие вершины-понятия:



- ограничение – замена вершины-понятия графа другой вершиной-понятием, представляющими его специализацию, или замена имени типа на имя экземпляра:

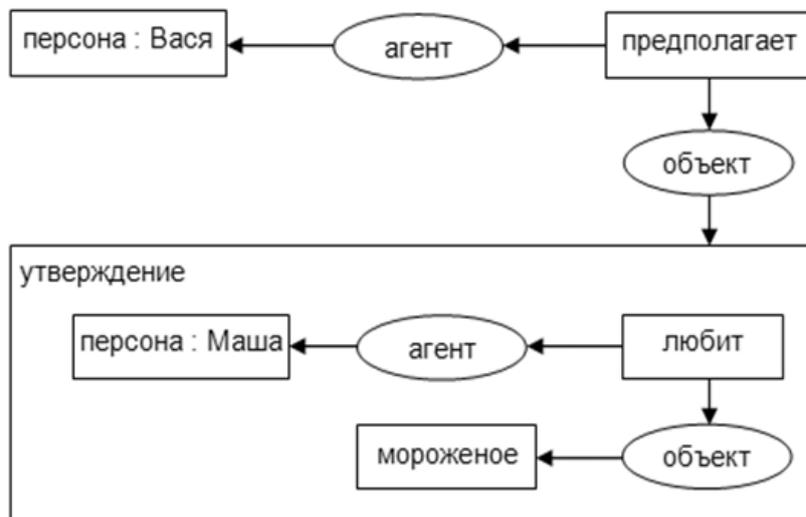


- упрощение – исключение дублирующих понятий или отношений:



С помощью концептуальных графов можно выразить отношения между высказываниями. Например, предложение: "Вася предполагает, что Маша любит мороженное". Здесь "предполагает" является отношением между субъектом "Вася" и высказыванием "Маша любит мороженное". Все предложение целиком является высказыванием о высказывании (метавысказыванием).

В формализме концептуальных графов выделяется особый класс понятий – утверждение. Понятие "утверждение" включает в себя один или несколько концептуальных графов, что и позволяет определять метавысказывания. Визуально "утверждение" выражается в виде прямоугольника, внутри которого располагаются другие концептуальные графы.



Помимо отображения высказываний в виде графов, наиболее наглядной и доступной для понимания форме, они могут быть выражены в т.н. линейной форме. При этом понятия заключаются в квадратные скобки, а отношения – в круглые. Например, граф, отображенный на рисунке, в линейной форме выглядит следующим образом:

```
[персона : Вася] <- (агент) <- [предполагает] -> (объект) ->
[[[персона : Маша] <- (агент) <- [любит] -> (объект) -> [мороженное]].
```

## Лекция 9

### ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И ПРОЕКТИРОВАНИЕ СЛОЖНЫХ СИСТЕМ

Объектно-ориентированный анализ и проектирование (ООАП, Object-Oriented Analysis/Design) – технология разработки программных систем, в основу которых положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов. Методология ООАП тесно связана с концепцией автоматизированной разработки программного обеспечения (Computer Aided Software Engineering, CASE).

Построение моделей сложных систем, отражающих десятки различных типов объектов и связей между ними, привело в конце 80-х годов к появлению большого числа различных графических нотаций, которые в той или иной степени были ориентированы на решение специальных классов задач. Сложилась парадоксальная ситуация, которая получила название "войны методов". Многие подходы, хотя и имели общие истоки, совершенно игнорировали другие альтернативные способы представления семантической информации. Наибольшее распространение в эти годы получил подход к моделированию программных систем, который назвали системным структурным анализом (ССА).

Под структурным системным анализом принято понимать метод исследования системы, который начинается с наиболее общего ее описания с последующей детализацией представления отдельных аспектов ее поведения и функционирования. При этом общая модель системы строится в виде некоторой иерархической структуры, которая отражает различные уровни абстракции с ограниченным числом компонентов на каждом из уровней. Одним из главных принципов структурного системного анализа является выделение на каждом из уровней абстракции только наиболее существенных компонентов или элементов системы.

В рамках данного направления принято рассматривать три графические нотации, получивших названия диаграмм: диаграммы "сущность-связь" (Entity-Relationship Diagrams, ERD), диаграммы функционального моделирования (Structured Analysis and Design Technique, SADT) и диаграммы потоков данных (Data Flow Diagrams, DFD).

**Графическая нотация** – это набор символов и правил их взаимодействия, используемых для визуального отображения модели процесса.

#### **Диаграммы "сущность-связь"**

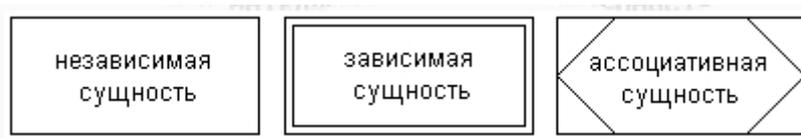
Диаграммы "сущность-связь" (ERD) предназначены для графического представления моделей и предлагают некоторый набор стандартных обозначений для определения объектов системы и отношений между ними. С помощью этого вида диаграмм можно описать отдельные компоненты модели и совокупность взаимосвязей между ними. Данная нотация была предложена Питером Ченом в 1976 году.



*Питер Чэнь Пиньшань – американский ученый в области информатики, предложивший в 1976 году ER-модель данных. Родился в 1947 г. в Китае.*

Основными понятиями данной нотации являются понятия сущности и связи. При этом под сущностью понимается произвольное множество реальных или абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае каждый рассматриваемый объект может являться экземпляром одной и только одной сущности, должен иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности.

Примерами сущностей могут быть: аэропорт, пассажир, рейс, компьютер, терминал, автомобиль, водитель. Каждая из сущностей может рассматриваться с различной степенью детализации и на различном уровне абстракции, что определяется конкретной постановкой задачи. Для графического представления сущностей используются специальные обозначения:



**Независимая сущность** – это сущность, экземпляры которой могут быть уникальным образом идентифицированы без определения ее связи с другой сущностью.

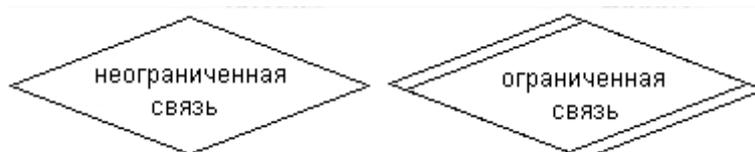
**Зависимая сущность** всегда имеет отношения с другими сущностями.

Например, сотрудник отдела. Сотрудник – зависимая сущность, отдел – независимая.

**Ассоциативными** являются зависимые сущности, у которых множество экземпляров одной сущности связаны с множеством экземпляров другой. Ассоциативные сущности позволяют нам моделировать пересечение экземпляров двух сущностей, обеспечивая уникальность каждого экземпляра ассоциации. Например, взаимоотношения врачей и пациентов.

Связь определяется как отношение или некоторая ассоциация между отдельными сущностями. Наиболее распространены бинарные связи, соединяющие два множества сущностей. Примерами связей могут являться родственные отно-

шения типа "отец-сын" или производственные отношения типа "начальник-подчиненный". Другой тип связей задается отношениями "иметь в собственности" или "обладать свойством". Различные типы связей графически изображаются в форме ромба с соответствующим именем данной связи:



Связь между сущностями, моделирующая отношения, которые всегда существуют, пока существуют соединяемые ими объекты, называется **неограниченной** (обязательной), например, "студент-учебная группа". **Ограниченная** (необязательная) связь моделирует условные отношения между объектами. В условной связи принимают участие не все экземпляры объекта, например, "студент-общежитие".

Графическая модель данных строится таким образом, чтобы связи между отдельными сущностями отражали не только семантический характер соответствующего отношения, но и дополнительные аспекты обязательности связей, а также кратность участвующих в данных отношениях экземпляров сущностей.

Различают связи типа:

- "один к одному": в каждый момент времени каждому представителю (экземпляру) сущности А соответствует 1 или 0 представителей сущности В. Например, "студент-стипендия": студент может не "заработать" стипендию, получить обычную или одну из повышенных стипендий.

- "один ко многим": одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В. Например, "жильцы-квартира": квартира может пустовать, в ней может жить один или несколько жильцов.

- "многие к одному", например: "студент-ВУЗ".

- "многие ко многим", например: "студент-дисциплина".

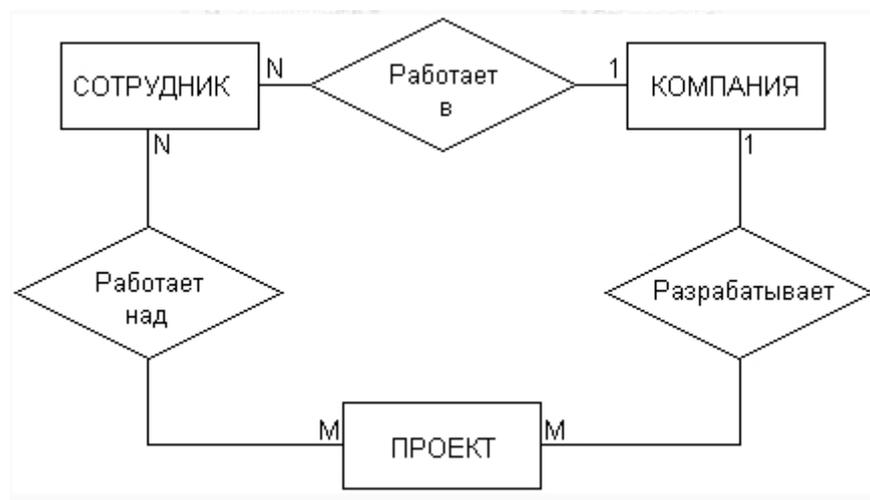
Рассмотрим в качестве простого примера ситуацию, которая описывается двумя сущностями: "Сотрудник" и "Компания". При этом в качестве связи естественно использовать отношение принадлежности сотрудника данной компании. Если учесть соображения о том, что в компании работают несколько сотрудников, и эти сотрудники не могут быть работниками других компаний, то данная информация может быть представлена графически в виде следующей диаграммы "сущность-связь". На данном рисунке буква "N" около связи означает тот факт, что в компании могут работать более одного сотрудника, при этом значение N заранее не фиксируется. Цифра "1" на другом конце связи означает, что сотрудник может работать только в одной конкретной компании, т. е. не допускается прием на работу сотрудников по совместительству из других компаний или учреждений.



Несколько иная ситуация складывается в случае рассмотрения сущностей "сотрудник" и "проект", и связи "участвует в работе над проектом". Поскольку в общем случае один сотрудник может участвовать в разработке нескольких проектов, а в разработке одного проекта могут принимать участие несколько сотрудников, то данная связь является многозначной. Данный факт специально отражается на диаграмме указанием букв "N" и "M" около соответствующих сущностей, при этом выбор конкретных букв не является принципиальным.



Рассмотренные две диаграммы могут быть объединены в одну, на которой будет представлена информация о сотрудниках компании, участвующих в разработке проектов данной компании. При этом может быть введена дополнительная связь, характеризующая проекты данной компании.



Недостатки: в модели отсутствует информация о поведении или функционировании отдельных ее компонентов.

### Диаграммы функционального моделирования

Начало разработки диаграмм функционального моделирования относится к середине 1960-х годов, когда Дуглас Т. Росс предложил специальную технику моделирования, получившую название SADT (Structured Analysis & Design Technique). Военно-воздушные силы США использовали методику функционального моделирования в качестве части своей программы интеграции компьютерных и промышленных технологий (Integrated Computer Aided Manufacturing,

ICAM) и назвали ее IDEFO (Icam DEFinition). Целью программы ICAM было увеличение эффективности компьютерных технологий в сфере проектирования новых средств вооружений и ведения боевых действий. Одним из результатов этих исследований являлся вывод о том, что описательные языки не эффективны для документирования и моделирования процессов функционирования сложных систем. Подобные описания на естественном языке не обеспечивают требуемого уровня непротиворечивости и полноты, имеющих доминирующее значение при решении задач моделирования.

Функциональная модель отображает структуру процессов функционирования системы и ее отдельных подсистем, т. е. выполняемые ими действия и связи между этими действиями. Для этой цели строятся специальные модели, которые позволяют в наглядной форме представить последовательность определенных действий. Исходными строительными блоками любой модели процесса являются деятельность и стрелки.

Деятельность представляет собой некоторое действие или набор действий, которые имеют фиксированную цель и приводят к некоторому конечному результату. Иногда деятельность называют просто процессом. Модели IDEFO отслеживают различные виды деятельности системы, их описание и взаимодействие с другими процессами. На диаграммах деятельность или процесс изображается прямоугольником, который называется блоком. Стрелка служит для обозначения некоторого носителя или воздействия, которые обеспечивают перенос данных или объектов от одной деятельности к другой. Стрелки также необходимы для описания того, что именно производит деятельность и какие ресурсы она потребляет. При этом различают стрелки четырех видов:

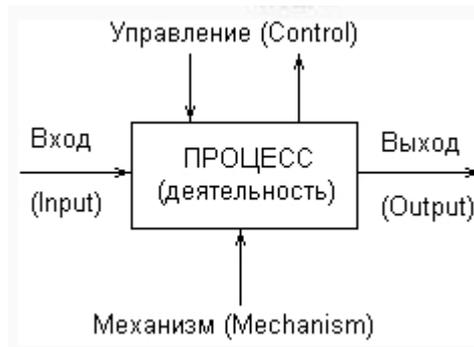
I (Input) – вход, т. е. все, что поступает в процесс или потребляется процессом.

C (Control) – управление или ограничения на выполнение операций процесса.

O (Output) – выход или результат процесса.

M (Mechanism) – механизм, который используется для выполнения процесса.

Методология функционального моделирования однозначно определяет, каким образом изображаются на диаграммах стрелки каждого вида. Стрелка Вход (Input) выходит из левой стороны рамки рабочего поля и входит слева в прямоугольник процесса. Стрелка Управление (Control) входит и выходит сверху. Стрелка Выход (Output) выходит из правой стороны процесса и входит в правую сторону рамки. Стрелка Механизм (Mechanism) входит в прямоугольник процесса снизу. Таким образом, базовое представление процесса на диаграммах функционального моделирования имеет следующий вид:

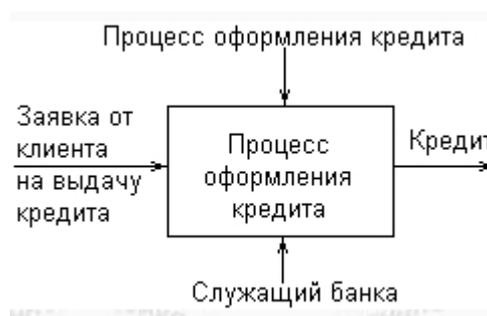


Техника построения диаграмм представляет собой главную особенность методологии функционального моделирования. Управляющая информация входит в блок сверху, в то время как информация, которая подвергается обработке, изображается с левой стороны блока. Результаты процесса представляются как выходы процесса и показываются с правой стороны блока. В качестве механизма может выступать человек или автоматизированная система, которые реализуют данную операцию. Соответствующий механизм на диаграмме представляется стрелкой, которая входит в блок процесса снизу.

Одной из наиболее важных особенностей методологии функционального моделирования является постепенное введение все более детальных представлений модели системы по мере разработки отдельных диаграмм. Построение модели начинается с представления всей системы в виде простейшей диаграммы, состоящей из одного блока процесса и стрелок, служащих для изображения основных видов взаимодействия с объектами вне системы. Поскольку исходный процесс представляет всю систему как единое целое, данное представление является наиболее общим и подлежит дальнейшей декомпозиции.

Для иллюстрации основных идей методологии функционального моделирования рассмотрим следующий простой пример. В качестве процесса будем представлять деятельность по оформлению кредита в банке. Входом данного процесса является заявка от клиента на получение кредита, а выходом – соответствующий результат, т. е. непосредственно кредит. При этом управляющими факторами являются правила оформления кредита, которые регламентируют условия получения соответствующих финансовых средств в кредит. Механизмом данного процесса является служащий банка, который уполномочен выполнить все операции по оформлению кредита.

Пример исходной диаграммы функционального моделирования для процесса оформления кредита в банке:



В конечном итоге модель представляет собой серию иерархически взаимосвязанных диаграмм с сопроводительной документацией, которая разбивает исходное представление сложной системы на отдельные составные части. Детали каждого из основных процессов представляются в виде более детальных процессов на других диаграммах. В этом случае каждая диаграмма нижнего уровня является декомпозицией некоторого процесса из более общей диаграммы. Поэтому на каждом шаге декомпозиции более общая диаграмма конкретизируется на ряд более детальных диаграмм.

В настоящее время диаграммы функционального моделирования используются для построения и детального анализа бизнес-процессов. Основным недостатком данной методологии связан с отсутствием явных средств для объектно-ориентированного представления моделей сложных систем, что существенно сужают диапазон решаемых с ее помощью задач.

### **Диаграммы потоков данных**

Модель системы в контексте диаграммы потоков данных представляется в виде некоторой информационной модели, основными компонентами которой являются различные потоки данных, которые переносят информацию от одной подсистемы к другой. Каждая из подсистем выполняет определенные преобразования входного потока данных и передает результаты обработки информации в виде потоков данных для других подсистем.

Основными компонентами диаграмм потоков данных являются:

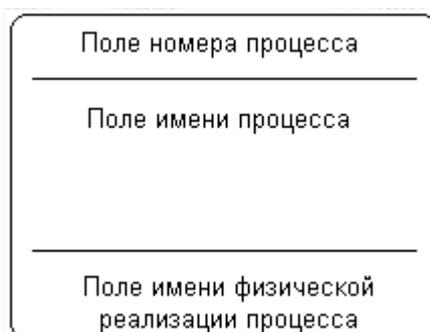
- внешние сущности;
- накопители данных или хранилища;
- процессы;
- потоки данных;
- системы/подсистемы.

Внешняя сущность представляет собой материальный объект или физическое лицо, которые могут выступать в качестве источника или приемника информации. Определение некоторого объекта или системы в качестве внешней сущности не является строго фиксированным. Хотя внешняя сущность находится за пределами границ рассматриваемой системы, в процессе дальнейшего анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы модели системы. С другой стороны, отдельные процессы могут быть вынесены за пределы диаграммы и представлены как внешние сущности. Примерами внешних сущностей могут служить: клиенты организации, заказчики, персонал, поставщики.

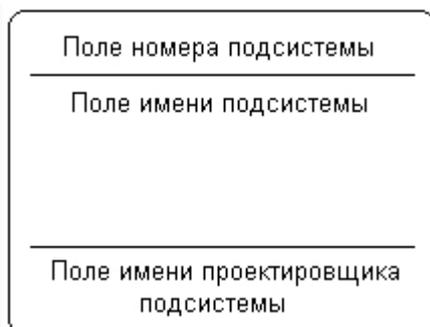
Внешняя сущность обозначается прямоугольником с тенью, внутри которого указывается ее имя. При этом в качестве имени рекомендуется использовать существительное в именительном падеже. Иногда внешнюю сущность называют также терминатором.



Процесс представляет собой совокупность операций по преобразованию входных потоков данных в выходные в соответствии с определенным алгоритмом или правилом. Хотя физически процесс может быть реализован различными способами, наиболее часто подразумевается программная реализация процесса. Процесс на диаграмме потоков данных изображается прямоугольником с закругленными вершинами, разделенным на три секции или поля горизонтальными линиями. Поле номера процесса служит для идентификации последнего. В среднем поле указывается имя процесса. В качестве имени рекомендовано использовать глагол в неопределенной форме с необходимыми дополнениями. Нижнее поле содержит указание на способ физической реализации процесса.



Информационная модель системы строится как некоторая иерархическая схема в виде так называемой контекстной диаграммы, на которой исходная модель последовательно представляется в виде модели подсистем соответствующих процессов преобразования данных. При этом подсистема или система на контекстной диаграмме потоков данных изображается так же, как и процесс – прямоугольником с закругленными вершинами.



Накопитель данных или хранилище представляет собой абстрактное устройство или способ хранения информации, перемещаемой между процессами. Предполагается, что данные можно в любой момент поместить в накопитель и через некоторое время извлечь, причем физические способы помещения и извлечения данных могут быть произвольными. Накопитель данных может быть физи-

чески реализован различными способами, но наиболее часто предполагается его реализация в электронном виде на магнитных носителях. Накопитель данных на диаграмме потоков данных изображается прямоугольником с двумя полями. Первое поле служит для указания номера или идентификатора накопителя, который начинается с буквы "D". Второе поле служит для указания имени. При этом в качестве имени накопителя рекомендуется использовать существительное, которое характеризует способ хранения соответствующей информации.



Поток данных определяет качественный характер информации, передаваемой через некоторое соединение от источника к приемнику. Реальный поток данных может передаваться по сети между двумя компьютерами или любым другим способом, допускающим извлечение данных и их восстановление в требуемом формате. Поток данных на диаграмме потоков данных изображается линией со стрелкой на одном из ее концов, при этом стрелка показывает направление потока данных. Каждый поток данных имеет свое собственное имя, отражающее его содержание.

Таким образом, информационная модель системы строится в виде диаграмм потоков данных, которые графически представляются с использованием соответствующей системы обозначений. В качестве примера рассмотрим упрощенную модель процесса получения некоторой суммы наличными по кредитной карточке клиентом банка. Внешними сущностями данного примера являются клиент банка и, возможно, служащий банка, который контролирует процесс обслуживания клиентов. Накопителем данных может быть база данных о состоянии счетов отдельных клиентов банка. Отдельные потоки данных отражают характер передаваемой информации, необходимой для обслуживания клиента банка. Соответствующая модель для данного примера может быть представлена в виде диаграммы потоков данных:



## Лекция 10

### ВВЕДЕНИЕ В ЯЗЫК ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ UML

#### Назначение UML

**Язык UML** – это графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых при разработке программных систем.

История развития языка UML берет начало с октября 1994 года, когда Гради Буч и Джеймс Рамбо из Rational Software Corporation (США) начали работу по унификации методов объектно-ориентированного моделирования. В 1995 г. к ним присоединился Ивар Якобсон из компании Objectory AB (Швеция). В январе 1997 года был опубликован документ с описанием языка UML 1.0. Эта версия языка моделирования была достаточно хорошо определена, обеспечивала требуемую выразительность и мощь и предполагала решение широкого класса задач.



Гради Буч



Джеймс Рамбо



Ивар Якобсон

**Спецификация** программного обеспечения – это законченное описание поведения программы, которую требуется разработать, т.е. описывает все варианты взаимодействия между пользователями и программным обеспечением.

В процессе разработки приложений участвуют, по меньшей мере, два действующих лица: заказчик и разработчик. Из-за того, что действующих лиц двое, очень многое зависит от степени их взаимопонимания. Заказчик может не осознавать своих объективных потребностей или неверно их интерпретировать. Разработчик может не разбираться в предметной области заказчика и интерпретировать формулировки спецификаций совершенно превратным образом. Основное назначение UML – предоставить формальное, удобное и универсальное средство, позволяющее снизить риск расхождений в толковании спецификаций.

**Визуализация** – общее название приёмов представления информации в виде, удобном для зрительного наблюдения и анализа.

Особенности человеческого восприятия таковы, что текст с картинками воспринимается легче, чем голый текст. Модели UML допускают представление в форме картинок, причем эти картинки наглядны, интуитивно понятны, практически однозначно интерпретируются и легко составляются. Таким образом, второе

по важности назначение UML состоит в том, чтобы служить адекватным средством коммуникации между людьми.

**Проектирование** программного обеспечения – это процесс создания проекта программного обеспечения.

UML позволяет создавать модели, для которых возможна автоматическая генерация программного кода приложений. Более того, природа моделей UML такова, что возможен и обратный процесс: автоматическое построение модели по коду готового приложения.

**Документирование** – это процесс создания документации. Программный документ – это документ, содержащий данные, необходимые для разработки, производства, эксплуатации или сопровождения программного средства.

Модели UML – это документы, которые можно хранить и использовать разными способами, начиная с печати картинок и заканчивая автоматической генерацией человекочитаемых текстовых описаний. Стандарт требует, чтобы во внутреннем представлении модели для каждого элемента моделирования было отведено место, где можно хранить неформальное текстовое описание этого элемента. Инструменты UML умеют из этих текстовых описаний собирать цельные, вполне осмысленные и хорошо отформатированные текстовые документы, которые можно использовать именно как привычные текстовые описания моделируемой системы.

### Особенности языка UML

1. Предназначен для разработки и документирования моделей сложных систем самого различного целевого назначения.

2. Предоставляет возможность расширения и специализации базовых понятий языка для более точного представления моделей систем в конкретной предметной области.

3. Описание языка UML поддерживает такую спецификацию моделей, которая не зависит от конкретных языков программирования и инструментальных средств проектирования программных систем.

4. Описание языка UML включает в себя семантический базис для понимания общих особенностей объектно-ориентированного анализа и проектирования.

Использование языка UML основывается на следующих общих принципах моделирования:

**абстрагирование** – в модель следует включать только те элементы проектируемой системы, которые имеют непосредственное отношение к выполнению системой своих функций или своего целевого предназначения. Другие элементы опускаются, чтобы не усложнять процесс анализа и исследования модели;

**многомодельность** – никакая единственная модель не может с достаточной степенью точности описать различные аспекты сложной системы. Допускается описывать систему некоторым числом взаимосвязанных представлений, каждое из которых отражает определенный аспект её поведения или структуры;

**иерархическое построение** – при описании системы используются различные уровни абстрагирования и детализации в рамках фиксированных представле-

ний. При этом первое представление системы описывает её в наиболее общих чертах и является представлением концептуального уровня, а последующие уровни раскрывают различные аспекты системы с возрастающей степенью детализации вплоть до физического уровня. Модель физического уровня в языке UML отражает компонентный состав проектируемой системы с точки зрения ее реализации на аппаратурной и программной платформах конкретных производителей.

### Семантика и нотация UML

С самой общей точки зрения описание языка UML состоит из двух взаимодействующих частей, таких как:

- **Семантика языка UML.** Представляет собой некоторую метамодель, которая определяет абстрактный синтаксис и семантику понятий объектного моделирования на языке UML.

- **Нотация языка UML.** Представляет собой графическую нотацию для визуального представления семантики языка UML.

Чтобы подчеркнуть, что UML язык графический, правила записи (рисования) моделей называют не синтаксисом, а нотацией.

При разработке UML были предложены и приняты разумные рекомендации по выбору нотации. Авторы исходили из того, что UML будет использоваться по-разному: начиная от не очень аккуратного рисования от руки на листке бумаги, печати черно-белых изображений в книгах и заканчивая созданием сложных диаграмм с помощью компьютера. Поэтому в качестве основных графических элементов были выбраны такие, которые было бы легко использовать во всех случаях. Типов элементов нотации пять:

- фигура (shape);
- линия (line);
- значок (icon);
- текст (text);
- рамка (frame).

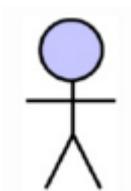
Фигуры (shape) в UML используются двумерные (т.е. их можно нарисовать на плоскости) и замкнутые (т.е. есть внутренняя и внешняя части). Фигуры могут менять свои размеры и форму, сохраняя при этом свои интуитивные отличительные признаки. Например, среди фигур UML есть прямоугольники и эллипсы. Они могут быть изображены многими способами: разного размера, с разным соотношением длин сторон (или, соответственно, полуосей), по-разному ориентированы относительно границ страницы и т.д., но во всех случаях прямоугольник отличен от эллипса и не может быть с ним спутан. Внутри фигур могут помещаться другие элементы нотации: тексты, линии, значки и даже другие фигуры. Единственное требование: должно быть однозначно понятно, что элемент нотации находится внутри фигуры, в частности, его изображение не должно пересекать границу фигуры. Например:



Линии (line) в UML, естественно, одномерные. Линии всегда присоединяются своими концами к фигурам или значкам, они не могут быть нарисованы сами по себе. Линия должна быть нарисована так, чтобы любому нормальному человеку было ясно, присоединяется данная линия к данной фигуре, или нет. Форма линий произвольна: это могут быть прямые, ломаные, плавные кривые – значения это не имеет. Толщина линий также произвольна. А вот стиль линии имеет значение. В UML используется только два стиля линий: сплошные и пунктирные линии. К линиям могут быть пририсованы различные дополнительные элементы: стрелки на концах, тексты и т.д. Единственное требование: должно быть ясно, что дополнительный элемент относится именно к данной линии. Линии могут пересекаться, и это ничего не значит, но рекомендуется избегать таких случаев, поскольку это затрудняет восприятие. Например:

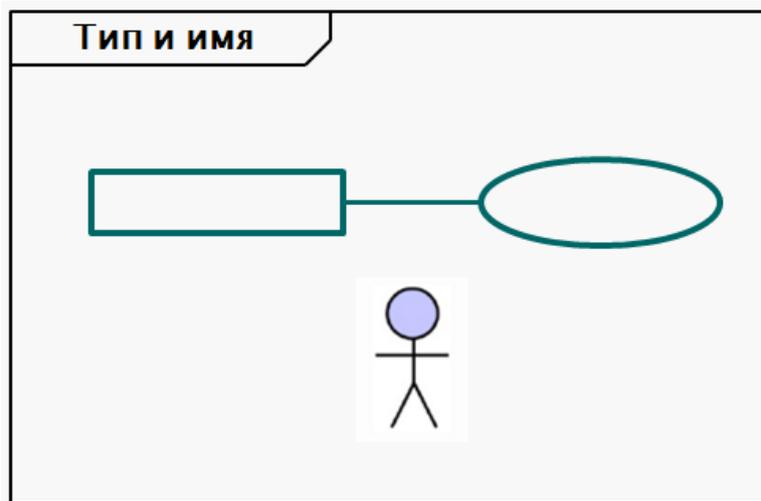


Значки (icon) в UML похожи на фигуры тем, что они двумерные, а отличаются тем, что не имеют внутренности, в которую можно что-то поместить, и, как правило, не меняют свою форму и размеры. Значки в UML используются очень умеренно, а потому сохраняют свою основную функцию однозначно воспринимаемого иероглифа. Например:



Тексты (text) в UML – это, как обычно, последовательности различных символов некоторого алфавита. Алфавит не фиксирован – он только должен быть понятен читателю модели. Гарнитура, размер и цвет шрифта не имеют значения, а вот начертание шрифта имеет: в UML различаются прямые, курсивные и подчеркнутые тексты. Предполагается, что читатель сумеет их различить.

Рамки (frame) – это частный случай фигуры, которая используется исключительно как контейнер для других фигур, линий, значков и текстов. Пустые рамки не применяются. Рамка имеет прямоугольную форму и, как правило, ярлычок в левом верхнем углу, в котором указывается тип и имя рамки. Например:



В общем, нотация UML довольно свободная: рисовать можно как угодно, лишь бы не возникало недоразумений. При этом следует придерживаться канонической нотации. Согласно ей любая модель может быть описана монохромными рисунками с текстовыми пояснениями. При этом рисунки должны оставаться узнаваемыми после печати на черно-белом принтере. Допускается использование цветов для заливки фигур и раскрашивания линий, тени у значков и фигур, разные шрифты в текстах, наконец, анимация изображений для повышения наглядности картинок.

В дополнение к этому, нотация UML описывает соответствие графической нотации базовым понятиям семантики. При этом семантика языка UML описывается на основе некоторой метамодели, имеющей три отдельных представления: абстрактный синтаксис, правила корректного построения выражений и семантику. Рассмотрение семантики языка UML предполагает некоторый "полуформальный" стиль изложения, который объединяет естественный и формальный языки для представления базовых понятий и правил их расширения. Семантика определяется для двух видов объектных моделей: структурных моделей и моделей поведения. Структурные модели, известные также как статические модели, описывают структуру сущностей или компонентов некоторой системы, включая их классы, интерфейсы, атрибуты и отношения. Модели поведения, называемые иногда динамическими моделями, описывают поведение или функционирование объектов системы, включая их методы, взаимодействие и сотрудничество между ними, а также процесс изменения состояний отдельных компонентов и системы в целом.

Нотация языка UML включает в себя описание отдельных семантических элементов, которые могут применяться при построении диаграмм. Формальное описание самого языка UML основывается на некоторой общей иерархической структуре модельных представлений, состоящей из четырех уровней:

- Мета-метамодель
- Метамодель
- Модель
- Объекты пользователя

Уровень мета-метамоделей образует исходную основу для всех метамодельных представлений. Мета-мета-модель определяет модель языка UML на самом высоком уровне абстракции и является наиболее компактным ее описанием. Примерами понятий этого уровня служат метакласс, метаатрибут, метаоперация.

Мета-модель является экземпляром или конкретизацией мета-мета-модели. Все основные понятия языка UML – это понятия уровня мета-модели. Примеры таких понятий – класс, атрибут, операция, компонент, ассоциация и многие другие.

Модель в контексте языка UML является экземпляром мета-модели в том смысле, что любая конкретная модель системы должна использовать только понятия мета-модели, конкретизировав их применительно к данной ситуации. Это уровень для описания информации о конкретной предметной области. Примерами понятий уровня модели могут служить, например, имена полей проектируемой базы данных, такие как имя и фамилия сотрудника, возраст, должность, адрес, телефон. При этом данные понятия используются лишь как имена соответствующих информационных атрибутов.

Конкретизация понятий модели происходит на уровне объектов. В настоящем контексте объект является экземпляром модели, поскольку содержит конкретную информацию относительно того, чему в действительности соответствуют те или иные понятия модели. Примером объекта может служить следующая запись в проектируемой базе данных: "Илья Петров, 30 лет, иллюзионист, ул. Невидимая, 10-20, 100-0000".

Описание семантики языка UML предполагает рассмотрение базовых понятий только уровня мета-модели. Мета-модель языка UML имеет довольно сложную структуру, которая включает в себя порядка 90 метаклассов, более 100 метаассоциаций и почти 50 стереотипов, число которых возрастает с появлением новых версий языка. Чтобы справиться с этой сложностью языка UML, все его элементы организованы в логические пакеты, основными из которых являются: основные элементы, элементы поведения и общие механизмы.

### **Пакеты в языке UML**

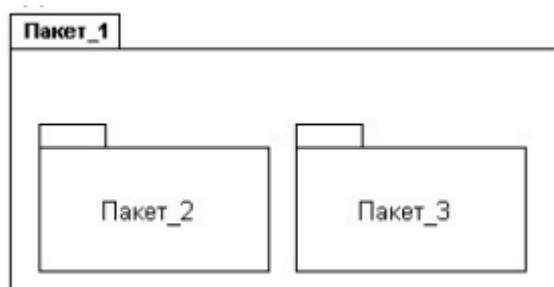
Пакет – основной способ организации элементов модели в языке UML. Каждый пакет владеет всеми своими элементами, т. е. теми элементами, которые включены в него. Про соответствующие элементы пакета говорят, что они принадлежат пакету или входят в него. При этом каждый элемент может принадлежать только одному пакету. В свою очередь, одни пакеты могут быть вложены в другие пакеты. В этом случае первые называются подпакетами, поскольку все элементы подпакета будут принадлежать более общему пакету. Тем самым для элементов модели задается отношение вложенности пакетов, которое представляет собой иерархию.

Для графического изображения пакетов на диаграммах применяется специальный графический символ – большой прямоугольник с небольшим прямоугольником, присоединенным к левой части верхней стороны первого. Можно сказать, что визуально символ пакета напоминает пиктограмму папки в популярном гра-

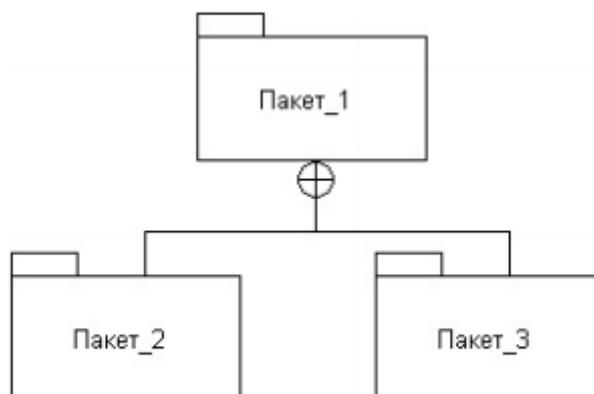
фическом интерфейсе. Внутри большого прямоугольника может записываться информация, относящаяся к данному пакету. Если такой информации нет, то внутри большого прямоугольника записывается имя пакета, которое должно быть уникальным в пределах рассматриваемой модели. Если же такая информация имеется, то имя пакета записывается в верхнем маленьком прямоугольнике.



Пакеты содержат лишь информацию о входящих в их состав элементах модели. Важно представить графически отношения, которые могут иметь место между отдельными пакетами. Например, отношение вложенности или включения пакетов друг в друга. В языке UML это отношение может быть изображено простым размещением одного пакета-прямоугольника внутри другого пакета-прямоугольника. Так, в данном случае пакет с именем Пакет\_1 содержит в себе два подпакета: Пакет\_2 и Пакет\_3:



Это же отношение может быть изображено с помощью отрезков линий аналогично графическому представлению дерева. В этом случае наиболее общий пакет (метапакет или контейнер) изображается в верхней части рисунка, а его подпакеты – уровнем ниже. Символ  $\oplus$  означает, что подпакеты являются "собственностью" или частью контейнера, и, кроме этих подпакетов, контейнер не содержит никаких других подпакетов.



Существуют и другие типы отношений между пакетами.

## Диаграммы UML

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций, получивших название диаграмм. Наиболее часто используются следующие диаграммы:

- Диаграмма вариантов использования (use case diagram)
- Диаграмма классов (class diagram)
- Диаграмма состояний (statechart diagram)
- Диаграмма деятельности (activity diagram)
- Диаграмма последовательности (sequence diagram)
- Диаграмма кооперации (collaboration diagram)
- Диаграмма компонентов (component diagram)
- Диаграмма развертывания (deployment diagram)

Каждая из этих диаграмм детализирует и конкретизирует различные представления о модели сложной системы в терминах языка UML.

Большинство перечисленных выше диаграмм являются в своей основе графами специального вида, состоящими из вершин в форме геометрических фигур, которые связаны между собой ребрами или дугами.

При графическом изображении диаграмм следует придерживаться следующих основных рекомендаций:

- Каждая диаграмма должна служить законченным представлением соответствующего фрагмента моделируемой предметной области. Речь идет о том, что в процессе разработки диаграммы необходимо учесть все сущности, важные с точки зрения контекста данной модели и диаграммы. Отсутствие тех или иных элементов на диаграмме служит признаком неполноты модели и может потребовать ее последующей доработки.

- Все сущности на диаграмме модели должны быть одного концептуального уровня. Здесь имеется в виду согласованность не только имен одинаковых элементов, но и возможность вложения отдельных диаграмм друг в друга для достижения полноты представлений. В случае достаточно сложных моделей систем желательно придерживаться стратегии последовательного уточнения или детализации отдельных диаграмм.

- Вся информация о сущностях должна быть явно представлена на диаграммах. Речь идет о том, что, хотя в языке UML при отсутствии некоторых символов на диаграмме могут быть использованы их значения по умолчанию (например, в случае неявного указания видимости атрибутов и операций классов), необходимо стремиться к явному указанию свойств всех элементов диаграмм.

- Диаграммы не должны содержать противоречивой информации. Противоречивость модели может служить причиной серьезных проблем при ее реализации и последующем использовании на практике. Например, наличие замкнутых путей при изображении отношений агрегирования или композиции приводит к ошибкам в программном коде, который будет реализовывать соответствующие классы. Наличие элементов с одинаковыми именами и различными атрибутами

свойств в одном пространстве имен также приводит к неоднозначной интерпретации и может служить источником проблем.

- Диаграммы не следует перегружать текстовой информацией. Принято считать, что визуализация модели является наиболее эффективной, если она содержит минимум пояснительного текста. Как правило, наличие больших фрагментов развернутого текста служит признаком недостаточной проработанности модели или ее неоднородности, когда в рамках одной модели представляется различная по характеру информация. Поскольку общая декомпозиция модели на отдельные типы диаграмм способна удовлетворить самые детальные представления разработчиков о системе, важно уметь правильно отображать те или иные сущности и аспекты моделирования в соответствующие элементы канонических диаграмм.

- Каждая диаграмма должна быть самодостаточной для правильной интерпретации всех ее элементов и понимания семантики всех используемых графических символов. Любые пояснительные тексты, которые не являются собственными элементами диаграммы (например, комментариями), не должны приниматься во внимание разработчиками. В то же время отдельные достаточно общие фрагменты диаграмм могут уточняться или детализироваться на других диаграммах этого же типа, образуя вложенные или подчиненные диаграммы. Таким образом, модель системы на языке UML представляет собой пакет иерархически вложенных диаграмм, детализация которых должна быть достаточной для последующей генерации программного кода, реализующего проект соответствующей системы.

- Количество типов диаграмм для конкретной модели приложения не является строго фиксированным. Речь идет о том, что для простых приложений нет необходимости строить все без исключения типы диаграмм. Некоторые из них могут просто отсутствовать в проекте системы, и этот факт не будет считаться ошибкой разработчика. Например, модель системы может не содержать диаграмму развертывания для приложения, выполняемого локально на компьютере пользователя. Важно понимать, что перечень диаграмм зависит от специфики конкретного проекта системы.

Любая из моделей системы должна содержать только те элементы, которые определены в нотации языка UML. Имеется в виду требование начинать разработку проекта, используя только те конструкции, которые уже определены в метамоделе UML. Как показывает практика, этих конструкций вполне достаточно для представления большинства типовых проектов программных систем. И только в случае отсутствия необходимых базовых элементов языка UML следует использовать механизмы их расширения для адекватного представления конкретной модели системы. При этом не допускается какое бы то ни было переопределение семантики тех элементов, которые отнесены к базовой нотации метамодела языка UML.

## Лекция 11

### ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Визуальное моделирование в UML можно представить как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы. Для достижения этих целей вначале строится модель в форме так называемой диаграммы вариантов использования (use case diagram), которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

В самом общем случае, диаграмма вариантов использования представляет собой граф специального вида, который является графической нотацией для представления конкретных вариантов использования, актеров, интерфейсов, и отношений между этими элементами.

#### **Элемент "Вариант использования"**

Элемент языка UML вариант использования применяется для спецификации общих особенностей поведения системы или любой другой сущности предметной области без рассмотрения внутренней структуры этой сущности. Каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером. Диаграмма вариантов может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов. Такой пояснительный текст получил название примечания или сценария.

Отдельный вариант использования обозначается на диаграмме эллипсом, внутри которого содержится его краткое название или имя в форме глагола с пояснительными словами:

Проверить наличие  
запчастей на складе

Примерами вариантов использования могут являться следующие действия: проверка наличия запасных частей на складе, оформление заказа на ремонт транспортного средства, оформление требования на закупку необходимых запчастей и другие действия.

### Элемент "Актер"

Актер представляет собой любую внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач. Стандартным графическим обозначением актера на диаграммах является фигурка "человечка", под которой записывается конкретное имя актера:



Примерами актеров могут быть: водитель транспортного средства, рабочий по ремонту транспортных средств, система автоматизированного учета запасных частей и другие сущности, имеющие отношение к концептуальной модели соответствующей предметной области.

В качестве актеров могут выступать другие системы или подсистемы проектируемой системы.

Так как в общем случае актер всегда находится вне системы, его внутренняя структура никак не определяется. Для актера имеет значение только его внешнее представление, т. е. то, как он воспринимается со стороны системы. Актеры взаимодействуют с системой посредством передачи и приема сообщений от вариантов использования. Сообщение представляет собой запрос актером сервиса от системы и получение этого сервиса. Это взаимодействие может быть выражено посредством ассоциаций между отдельными актерами и вариантами использования. Кроме этого, с актерами могут быть связаны интерфейсы, которые определяют, каким образом другие элементы модели взаимодействуют с этими актерами.

Два и более актера могут иметь общие свойства, т. е. взаимодействовать с одним и тем же множеством вариантов использования одинаковым образом.

### Интерфейсы

Интерфейс (interface) – именованное множество операций, определяющее набор услуг, которые могут быть запрошены потребителем и предоставлены поставщиком услуг. Применительно к диаграммам вариантов использования, интерфейсы определяют совокупность операций, которые обеспечивают необходи-

мый набор сервисов или функциональности для актеров. Интерфейсы содержат только операции без указания особенностей их реализации. Формально интерфейс эквивалентен абстрактному классу без атрибутов и методов с наличием только абстрактных операций.

На диаграмме вариантов использования интерфейс изображается в виде маленького круга, рядом с которым записывается его имя. В качестве имени может быть существительное, которое характеризует соответствующую информацию или сервис (например, "датчик", "сирена", "видеокамера"), но чаще строка текста (например, "запрос к базе данных", "форма ввода", "устройство подачи звукового сигнала").



Графический символ отдельного интерфейса может соединяться на диаграмме сплошной линией с тем вариантом использования, который его поддерживает. Сплошная линия в этом случае указывает на тот факт, что связанный с интерфейсом вариант использования должен реализовывать все операции, необходимые для данного интерфейса.



Кроме этого, интерфейсы могут соединяться с вариантами использования пунктирной линией со стрелкой, означающей, что вариант использования предназначен для спецификации только того сервиса, который необходим для реализации данного интерфейса.

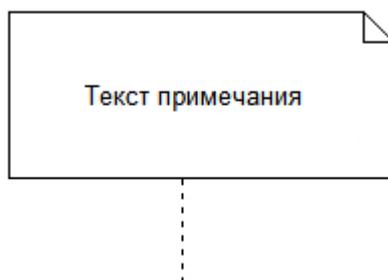


### Примечания

Примечания (notes) в языке UML предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта. Применительно к диаграммам вариантов использования примечание может носить самую общую информацию, относящуюся к общему контексту системы.

Графически примечания обозначаются прямоугольником с "загнутым" верхним правым уголком. Внутри прямоугольника содержится текст примечания.

Примечание может относиться к любому элементу диаграммы, в этом случае их соединяет пунктирная линия. Если примечание относится к нескольким элементам, то от него проводятся, соответственно, несколько линий.



### Отношения на диаграмме вариантов использования

Между компонентами диаграммы вариантов использования могут существовать различные отношения, которые описывают взаимодействие экземпляров одних актеров и вариантов использования с экземплярами других актеров и вариантов. Один актер может взаимодействовать с несколькими вариантами использования. В этом случае этот актер обращается к нескольким сервисам данной системы. В свою очередь один вариант использования может взаимодействовать с несколькими актерами, предоставляя для всех них свой сервис. В языке UML имеется несколько стандартных видов отношений между актерами и вариантами использования:

- Отношение ассоциации (association relationship)
- Отношение расширения (extend relationship)
- Отношение обобщения (generalization relationship)
- Отношение включения (include relationship)

### Отношение ассоциации

Отношение ассоциации является одним из фундаментальных понятий в языке UML и в той или иной степени используется при построении всех графических моделей систем в форме канонических диаграмм.

Применительно к диаграммам вариантов использования оно служит для обозначения специфической роли актера в отдельном варианте использования. Таким образом, это отношение устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования. На диаграмме вариантов использования отношение ассоциации обозначается сплошной линией между актером и вариантом использования. Эта линия может иметь дополнительные условные обозначения, такие, например, как имя и кратность.



Кратность (multiplicity) ассоциации указывается рядом с обозначением компонента диаграммы, который является участником данной ассоциации. Кратность характеризует общее количество конкретных экземпляров данного компонента, которые могут выступать в качестве элементов данной ассоциации. Для диаграмм вариантов использования наиболее распространенными являются четыре основные формы записи кратности отношения ассоциации:

- Целое неотрицательное число (включая цифру 0). Предназначено для указания кратности, которая является строго фиксированной для элемента соответствующей ассоциации. В этом случае количество экземпляров актеров или вариантов использования, которые могут выступать в качестве элементов отношения ассоциации, в точности равно указанному числу.

Примером этой формы записи кратности ассоциации является указание кратности "1" для актера "Водитель". Эта запись означает, что каждый экземпляр варианта использования "Оформить заказ на ремонт транспортного средства" может иметь в качестве своего элемента единственный экземпляр актера "Водитель".

- Два целых неотрицательных числа, разделенные двумя точками и записанные в виде: "первое число .. второе число". Эту запись следует понимать как множество целых неотрицательных чисел, следующих в последовательно возрастающем порядке:

[первое\_число, первое\_число+1, первое\_число+2, ..., второе\_число].

Очевидно, что первое число должно быть строго меньше второго числа в арифметическом смысле, при этом первое число может быть равно 0.

Пример такой формы записи кратности ассоциации: "1..5". Эта запись означает, что количество отдельных экземпляров данного компонента, которые могут выступать в качестве элементов данной ассоциации, равно некоторому заранее неизвестному числу из множества целых чисел {1, 2, 3, 4, 5}.

- Два символа, разделенные двумя точками. При этом первый из них является целым неотрицательным числом или 0, а второй – специальным символом "\*". Здесь символ "\*" обозначает произвольное конечное целое неотрицательное число, значение которого неизвестно на момент задания соответствующего отношения ассоциации.

Пример такой формы записи кратности ассоциации: "2..\*". Запись означает, что количество отдельных экземпляров данного компонента, которые могут выступать в качестве элементов данной ассоциации, равно некоторому заранее неизвестному числу из подмножества натуральных чисел: {2, 3, 4}.

- Единственный символ "\*", который является сокращением записи интервала "0..\*". В этом случае количество отдельных экземпляров данного компонента отношения ассоциации может быть любым целым неотрицательным числом. При этом 0 означает, что для некоторых экземпляров соответствующего компонента данное отношение ассоциации может вовсе не иметь места.

Если кратность отношения ассоциации не указана, то по умолчанию принимается ее значение, равное 1.

### Отношение расширения

Отношение расширения определяет взаимосвязь экземпляров отдельного варианта использования с более общим вариантом, свойства которого определяются на основе способа совместного объединения данных экземпляров. Так, если имеет место отношение расширения от варианта использования А к варианту использования В, то это означает, что свойства экземпляра варианта использования В могут быть дополнены благодаря наличию свойств у расширенного варианта использования А.

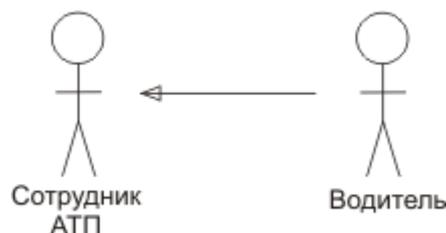
Отношение расширения между вариантами использования обозначается пунктирной линией со стрелкой, направленной от того варианта использования, который является расширением для исходного варианта использования. Данная линия со стрелкой помечается ключевым словом "extend" ("расширяет").



### Отношение обобщения

Отношение обобщения служит для указания того факта, что некоторый вариант использования А может быть обобщен до варианта использования В. В этом случае вариант А будет являться специализацией варианта В. При этом В называется предком или родителем по отношению А, а вариант А – потомком по отношению к варианту использования В. Следует подчеркнуть, что потомок наследует все свойства и поведение своего родителя, а также может быть дополнен новыми свойствами и особенностями поведения. Графически данное отношение обозначается сплошной линией со стрелкой в форме незакрашенного треугольника, которая указывает на родительский вариант использования.

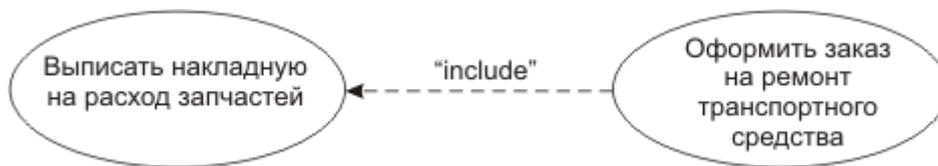
Между отдельными актерами также может существовать отношение обобщения. Данное отношение является направленным и указывает на факт специализации одних актеров относительно других. Например, отношение обобщения от актера А к актеру В отмечает тот факт, что каждый экземпляр актера А является одновременно экземпляром актера В и обладает всеми его свойствами. В этом случае актер В является родителем по отношению к актеру А, а актер А, соответственно, потомком актера В. При этом актер А обладает способностью играть такое же множество ролей, что и актер В.



## Отношение включения

Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования.

Отношение включения, направленное от варианта использования А к варианту использования В, указывает, что каждый экземпляр варианта А включает в себя функциональные свойства, заданные для варианта В. Эти свойства специализируют поведение соответствующего варианта А на данной диаграмме. Графически данное отношение обозначается пунктирной линией со стрелкой, направленной от базового варианта использования к включаемому. При этом данная линия со стрелкой помечается ключевым словом "include" ("включает").



## Лекция 12

### ДИАГРАММЫ UML

#### Диаграмма классов

Центральное место в ООАП занимает разработка логической модели системы в виде диаграммы классов. Диаграмма классов (class diagram) служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

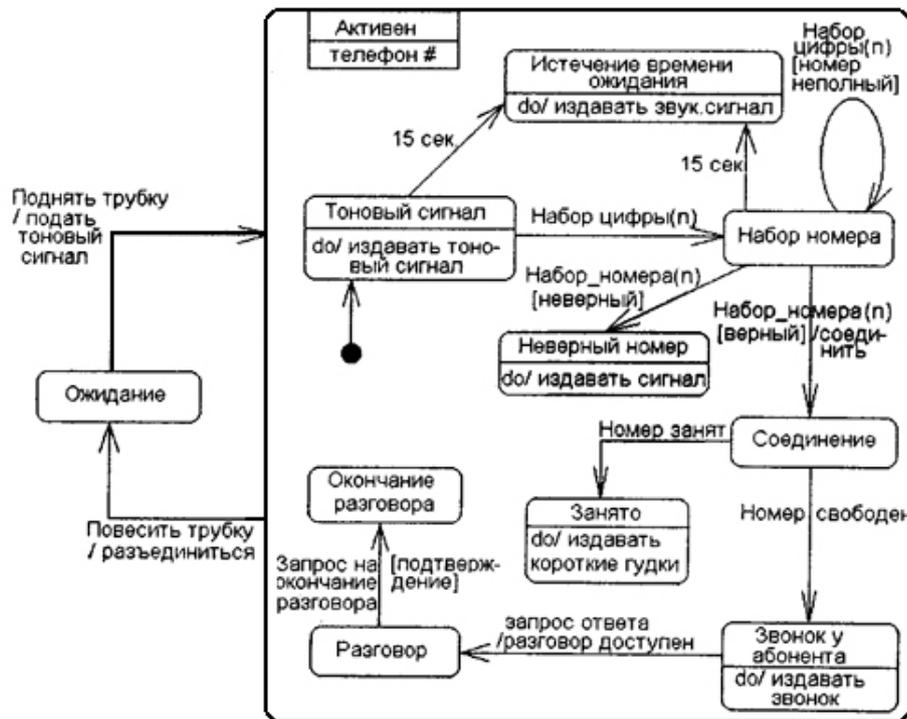


### Диаграмма состояний

Рассмотренная выше диаграмма классов представляет собой логическую модель статического представления моделируемой системы. Речь идет о том, что на данной диаграмме изображаются только взаимосвязи структурного характера, не зависящие от времени или реакции системы на внешние события. Однако для большинства физических систем, кроме самых простых и тривиальных, статических представлений совершенно недостаточно для моделирования процессов функционирования подобных систем как в целом, так и их отдельных подсистем и элементов.

Рассмотрим простой пример. Любое техническое устройство, такое как телевизор, компьютер, автомобиль, телефонный аппарат в самом общем случае может характеризоваться такими своими состояниями, как "исправен" и "неисправен". Интуитивно ясно, какой смысл вкладывается в каждое из этих понятий. Более того, использование по назначению данного устройства возможно только тогда, когда оно находится в исправном состоянии. В противном случае необходимо предпринять совершенно конкретные действия по его ремонту и восстановлению работоспособности.

Главное предназначение диаграммы состояний - описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий.



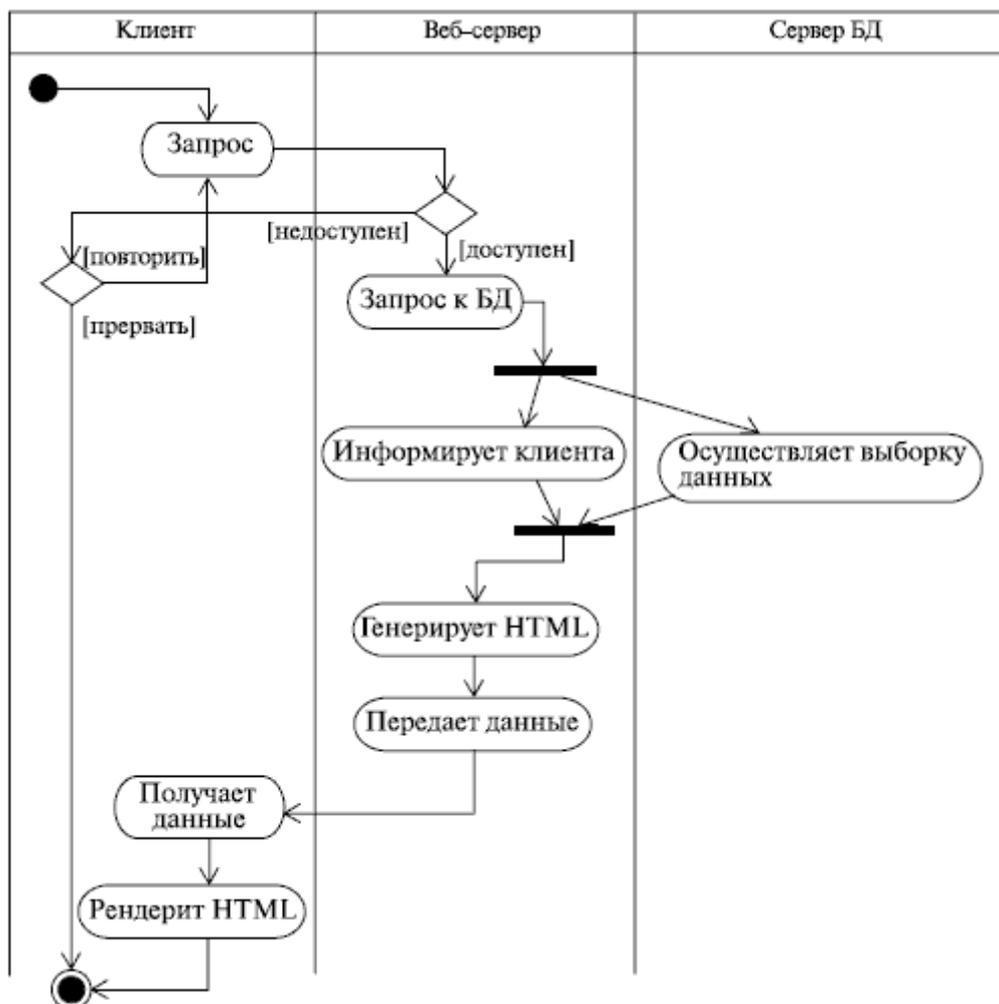
### Диаграмма деятельности

При моделировании поведения проектируемой или анализируемой системы возникает необходимость не только представить процесс изменения ее состояний, но и детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Традиционно для этой цели использовались блок-схемы или структурные схемы алгоритмов. Каждая такая схема акцентирует внимание на последовательности выполнения определенных действий или элементарных операций, которые в совокупности приводят к получению желаемого результата.

Алгоритмические и логические операции, требующие выполнения в определенной последовательности, окружают нас постоянно. Конечно, мы не всегда задумываемся о том, что подобные операции относятся к столь научным категориям. Например, чтобы позвонить по телефону, нам предварительно нужно снять трубку или включить его. Для приготовления кофе или заваривания чая необходимо вначале вскипятить воду. Чтобы выполнить ремонт двигателя автомобиля, требуется осуществить целый ряд нетривиальных операций, таких как разборка силового агрегата, снятие генератора и некоторых других.

Важно подчеркнуть то обстоятельство, что с увеличением сложности системы строгое соблюдение последовательности выполняемых операций приобретает все более важное значение. Если попытаться заварить кофе холодной водой, то мы можем только испортить одну порцию напитка. Нарушение последовательности операций при ремонте двигателя может привести к его поломке или выходу из строя. Еще более катастрофические последствия могут произойти в случае отклонения от установленной последовательности действий при взлете или посадке авиалайнера, запуске ракеты, регламентных работах на АЭС.

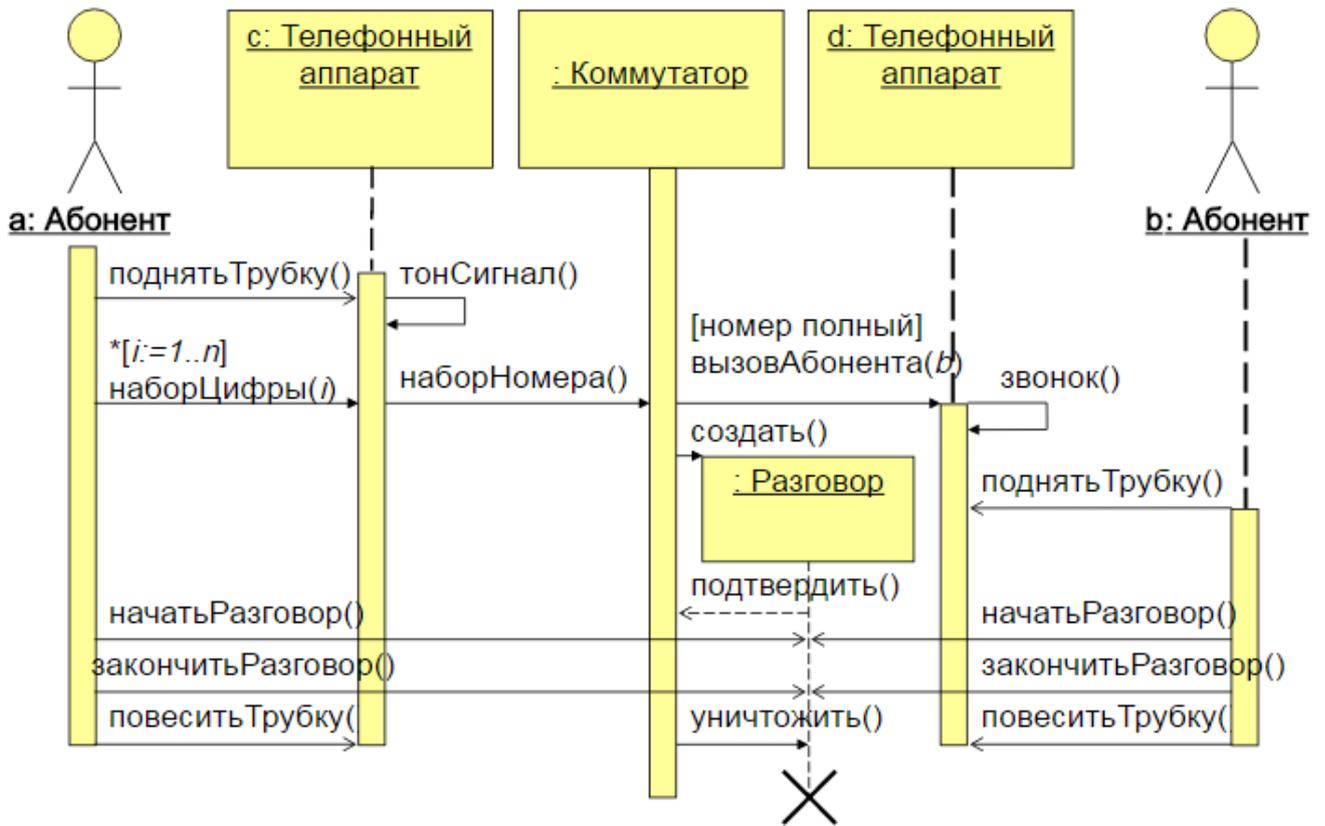
Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой, операции в предыдущем состоянии.



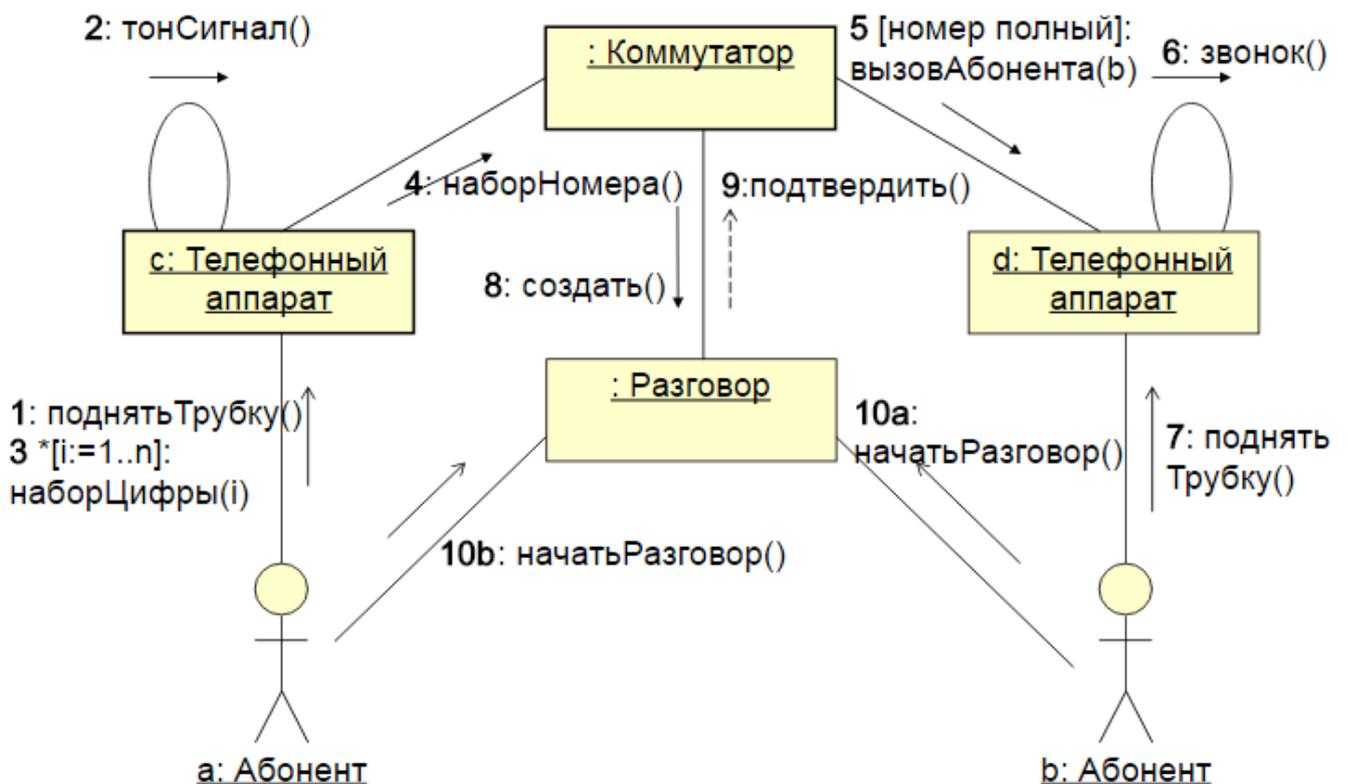
### Диаграммы последовательности и кооперации

Одной из характерных особенностей систем различной природы и назначения является взаимодействие между собой отдельных элементов, из которых образованы эти системы. Речь идет о том, что различные составные элементы систем не существуют изолированно, а оказывают определенное влияние друг на друга, что и отличает систему как целостное образование от простой совокупности элементов.

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. Говоря об этих диаграммах, имеют в виду два аспекта взаимодействия. Во-первых, взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.



Во-вторых, можно рассматривать структурные особенности взаимодействия объектов. Для представления структурных особенностей передачи и приема сообщений между объектами используется диаграмма кооперации.



## Диаграммы компонентов и развертывания

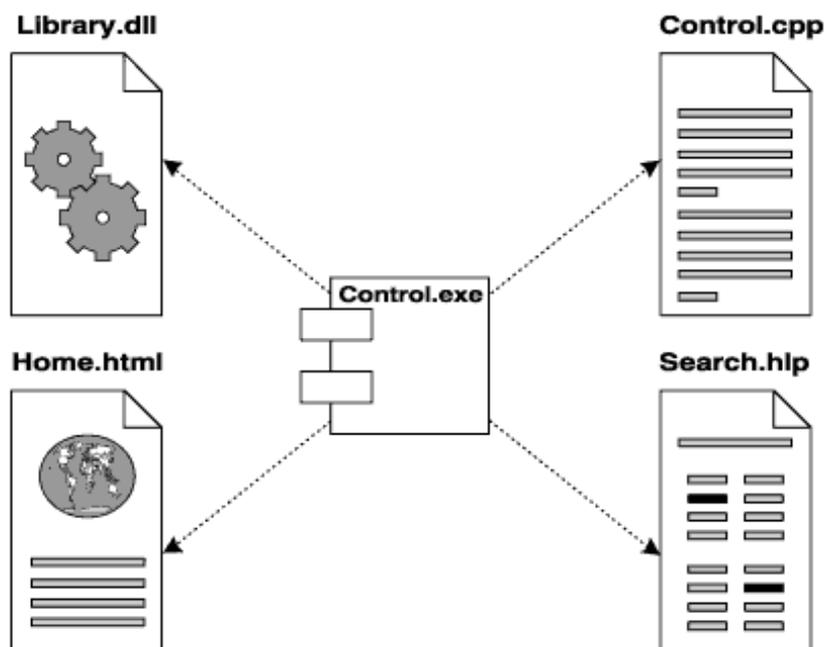
Все рассмотренные ранее диаграммы отражали концептуальные аспекты построения модели системы и относились к логическому уровню представления. Особенность логического представления заключается в том, что оно оперирует понятиями, которые не имеют самостоятельного материального воплощения. Другими словами, различные элементы логического представления не существуют материально или физически. Они лишь отражают наше понимание структуры физической системы или аспекты ее поведения. Основное назначение логического представления состоит в анализе структурных и функциональных отношений между элементами модели системы. Однако для создания конкретной физической системы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели.

Чтобы пояснить отличие логического и физического представлений, рассмотрим в общих чертах процесс разработки некоторой программной системы. Ее исходным логическим представлением могут служить структурные схемы алгоритмов и процедур, описания интерфейсов и концептуальные схемы баз данных. Однако для реализации этой системы необходимо разработать исходный текст программы на некотором языке программирования (C++, Pascal, Basic/VBA, Java). При этом уже в тексте программы предполагается такая организация программного кода, которая предполагает его разбиение на отдельные модули.

Тем не менее, исходные тексты программы еще не являются окончательной реализацией проекта, хотя и служат фрагментом его физического представления. Очевидно, программная система может считаться реализованной в том случае, когда она будет способна выполнять функции своего целевого предназначения. А это возможно, только если программный код системы будет реализован в форме исполняемых модулей, библиотек классов и процедур, стандартных графических интерфейсов, файлах баз данных. Именно эти компоненты являются необходимыми элементами физического представления системы.

Таким образом, полный проект программной системы представляет собой совокупность моделей логического и физического представлений, которые должны быть согласованы между собой. В языке UML для физического представления моделей систем используются так называемые диаграммы реализации (implementation diagrams), которые включают в себя две отдельные канонические диаграммы: диаграмму компонентов и диаграмму развертывания.

Диаграмма компонентов описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код.



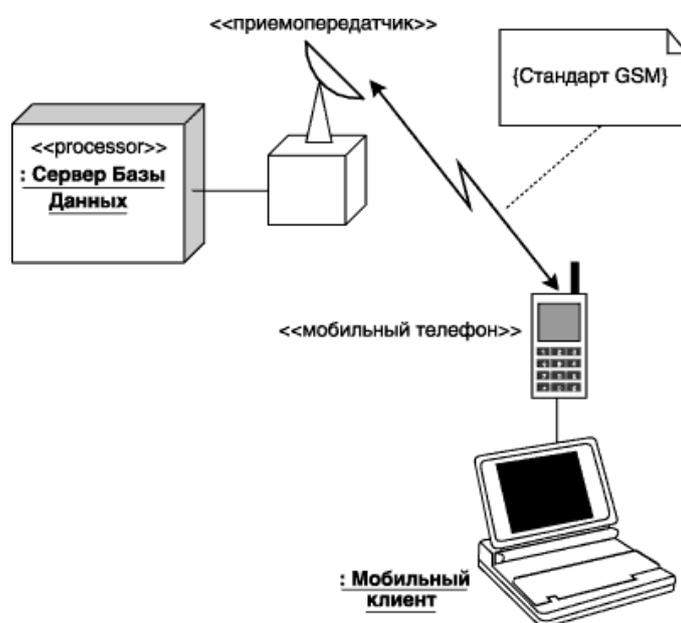
Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Конечно, если разрабатывается простая программа, которая может выполняться локально на компьютере пользователя, не задействуя никаких периферийных устройств и ресурсов, то в этом случае нет необходимости в разработке дополнительных диаграмм. Однако при разработке корпоративных приложений ситуация представляется совсем по-другому.

Во-первых, сложные программные системы могут реализовываться в сетевом варианте на различных вычислительных платформах и технологиях доступа к распределенным базам данных. Наличие локальной корпоративной сети требует решения целого комплекса дополнительных задач по рациональному размещению компонентов по узлам этой сети, что определяет общую производительность программной системы.

Во-вторых, интеграция программной системы с Интернетом определяет необходимость решения дополнительных вопросов при проектировании системы, таких как обеспечение безопасности, криптозащищенности и устойчивости доступа к информации для корпоративных клиентов. Эти аспекты в немалой степени зависят от реализации проекта в форме физически существующих узлов системы, таких как серверы, рабочие станции, брандмауэры, каналы связи и хранилища данных.

Наконец, технологии доступа и манипулирования данными в рамках общей схемы "клиент-сервер" также требуют размещения больших баз данных в различных сегментах корпоративной сети, их резервного копирования, архивирования, кэширования для обеспечения необходимой производительности системы в целом. Эти аспекты также требуют визуального представления с целью спецификации программных и технологических особенностей реализации распределенных архитектур.

Второй формой физического представления программной системы является диаграмма развертывания (синоним – диаграмма размещения). Она применяется для представления общей конфигурации и топологии распределенной программной системы и содержит распределение компонентов по отдельным узлам системы. Кроме того, диаграмма развертывания показывает наличие физических соединений – маршрутов передачи информации между аппаратными устройствами, задействованными в реализации системы.



## Лекция 13

### ОСОБЕННОСТИ РЕАЛИЗАЦИИ ЯЗЫКА UML В CASE-ИНСТРУМЕНТАРИИ RATIONAL ROSE

#### Общая характеристика CASE-средства Rational Rose

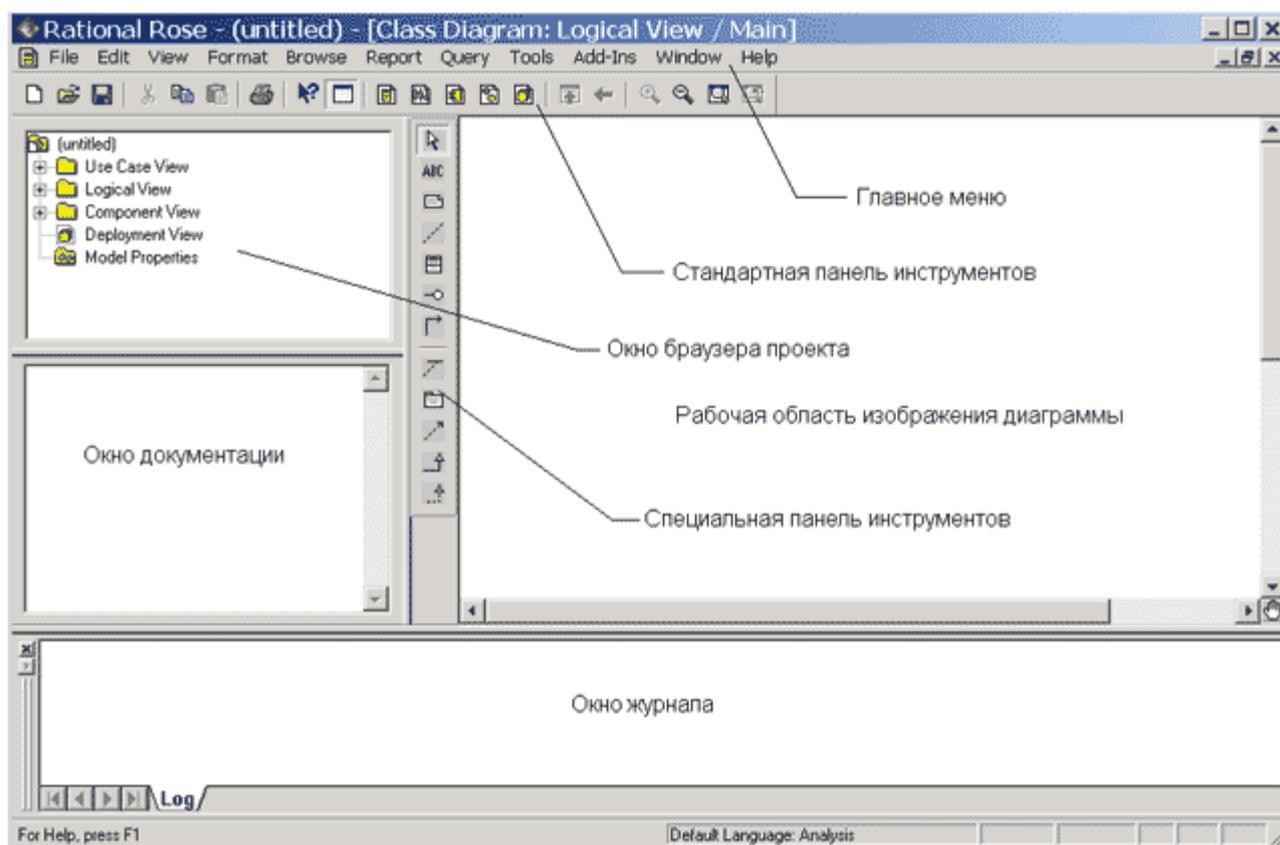
Появление на рынке программных продуктов первых CASE-средств (Computer Aided Software Engineering) ознаменовало новый этап развития программной инженерии, характерными особенностями которого являются существенное сокращение сроков разработки программных проектов, реализация проектов группой программистов и ориентация на визуальные средства спецификации компонентов программного обеспечения. Среди всех фирм-производителей CASE-средств именно компания Rational Software Corp. одна из первых осознала стратегическую перспективность развития объектно-ориентированных технологий анализа и проектирования программных систем. Эта компания выступила инициатором унификации языка визуального моделирования, что, в конечном итоге, привело к появлению первых версий языка UML. И эта же компания первой разработала инструментальное объектно-ориентированное CASE-средство, в котором был реализован язык UML как базовая нотация визуального моделирования. CASE-средство Rational Rose со времени

своего появления претерпело серьезную эволюцию и превратилось в современное и мощное средство анализа, моделирования и разработки программных систем

### Интерфейс Rational Rose

Рабочий интерфейс Rational Rose состоит из различных элементов, основными из которых являются:

- Главное меню программы
- Окно диаграммы
- Стандартная панель инструментов
- Окно документации
- Окно браузера
- Окно журнала
- Специальная панель инструментов



Общий вид рабочего интерфейса CASE-средства Rational Rose 2003

Отдельные пункты **главного меню**, назначение которых понятно из их названий, объединяют сходные операции, относящиеся ко всему проекту в целом. Некоторые из пунктов меню содержат хорошо знакомые функции (открытие проекта, вывод печать диаграмм, копирование в буфер и вставка из буфера различных элементов диаграмм). Другие настолько специфичны, что могут потребовать дополнительных усилий на изучение (опции генерации программного кода, проверка согласованности моделей, подключение дополнительных модулей).

**Стандартная панель инструментов** обеспечивает быстрый доступ к тем командам меню, которые выполняются разработчиками наиболее часто. Пользователь может настроить внешний вид этой панели по своему усмотрению.

**Окно браузера** предназначено для представления модели в виде иерархической структуры, которая упрощает навигацию и позволяет отыскать любой элемент модели в проекте. При этом любой элемент, который разработчик добавляет в модель, сразу отображается в окне браузера. Соответственно, выбрав элемент в окне браузера, мы можем его визуализировать в окне диаграммы или изменить его спецификацию. Браузер позволяет также организовывать элементы модели в пакеты и перемещать элементы между различными представлениями модели.

**Специальная панель инструментов** по умолчанию предназначена для построения диаграммы классов модели. Состав панели можно настраивать под конкретный вид диаграммы.

**Окно диаграммы** является основной рабочей областью ее интерфейса, в которой визуализируются различные представления модели проекта. При разработке нового проекта окно диаграммы представляет собой чистую область, не содержащую никаких элементов модели. Одновременно в окне диаграммы могут присутствовать несколько диаграмм, однако активной может быть только одна из них.

**Окно документации** предназначено для документирования элементов представления модели. В него можно записывать самую различную информацию, в том числе и на русском языке. Эта информация в последующем преобразуется в комментарии и никак не влияет на логику выполнения программного кода. В окне документации отображается только та информация, которая относится к отдельному выделенному элементу диаграммы.

**Окно журнала** предназначено для автоматической записи различной служебной информации, образующейся в ходе работы с программой. В журнале фиксируется время и характер выполняемых разработчиком действий, таких как обновление модели, настройка меню и панелей инструментов, а также сообщения об ошибках, возникающих при генерации программного кода.

## ЛИТЕРАТУРА

1. Сурмин, Ю.П. Теория систем и системный анализ: учеб. пособ. / Ю.П. Сурмин – К.: МАУП, 2008. – 369 с.
2. Антонов, А.В. Системный анализ. Учеб. для ВУЗов / А.В. Антонов – М.: Высш. шк., 2006. – 452 с.
3. Спицнадель, В.Н. Основы системного анализа: учеб. пособ. / В.Н. Спицнадель – М., Бизнес-пресса, 2009. – 512 с.
4. Ніконов, О.Я. Основи системного аналізу: навч. посіб. / О.Я. Ніконов, А.І. Кудін, М.В. Костікова, І.В. Скрипіна, В.О. Шевченко – Х.: ХНАДУ, 2013. – 160 с.
5. Системный анализ и примеры решений. Словарь-справочник: учеб. пособ. для ВУЗов / Под ред. В.Н. Волковой, В.Н. Козлова – М.: Высш. шк., 2004. – 616 с.
6. <http://dl.khadi.kharkiv.edu/course/view.php?id=331> – Курс "Теорія систем та системний аналіз" для студентів потоку 4РК: матеріали для самостійної роботи, лекції по курсу, методичні вказівки до підготовки та виконання лабораторних робіт, методичні вказівки та приклад оформлення курсової роботи, питання для підготовки до контролю знань.