ITMO UNIVERSITY

insight

Архитектура программных систем

Занятие #2: Введение в UML

sdCloud

sdLitica

ITMO UNIVERSITY

insight

План лекции

- Понятие информационной системы
- Введение в визуальное моделирование
- История развития UML
- Внутренности UML

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Понятие информационной системы

- **Теория информационных процессов** –

- **Информатика** –

- **Информационные технологии** –

sdCloud

sdLitica

ITMO UNIVERSITY

insight

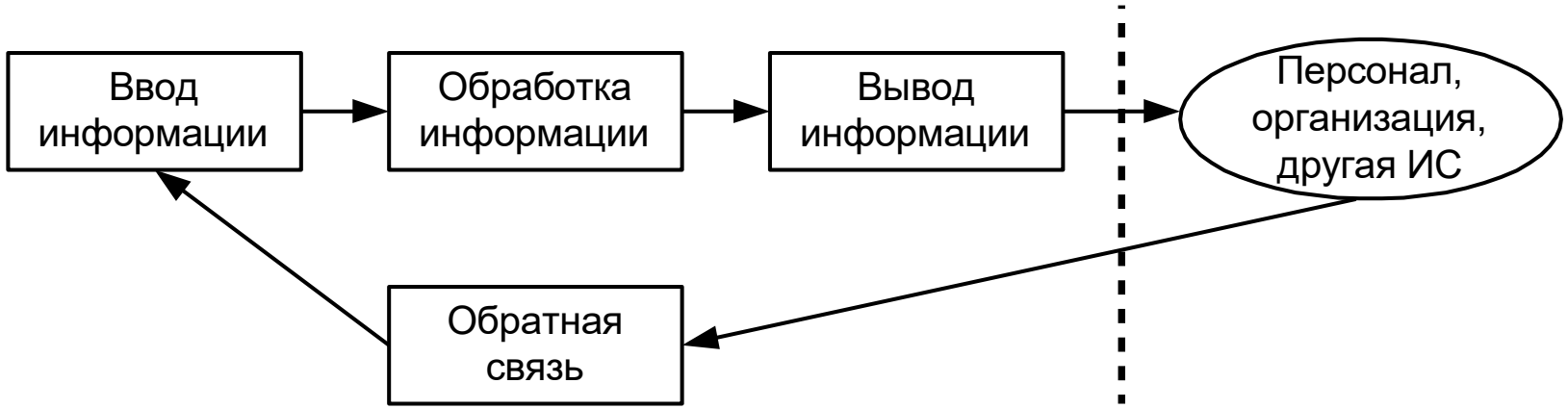
Понятие информационной системы

$$\left[\begin{array}{c}\textit{Информационные}\\ \textit{процессы}\end{array}\right] \; ? \; [\textit{Информатика}] \; ? \; \left[\begin{array}{c}\textit{Информационные}\\ \textit{технологии}\end{array}\right]$$

sdCloud

sdLitica

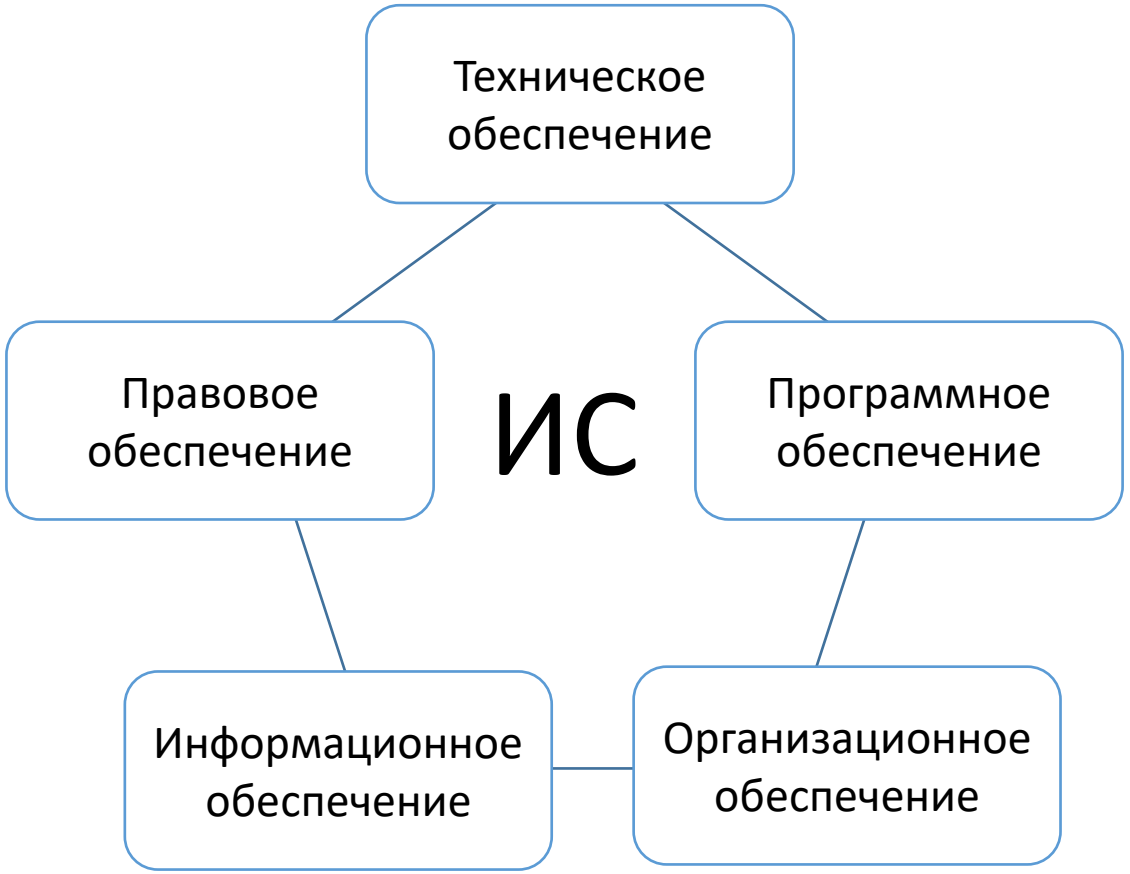
Процессы, протекающие в ИС



ITMO UNIVERSITY

insight

Понятие информационной системы



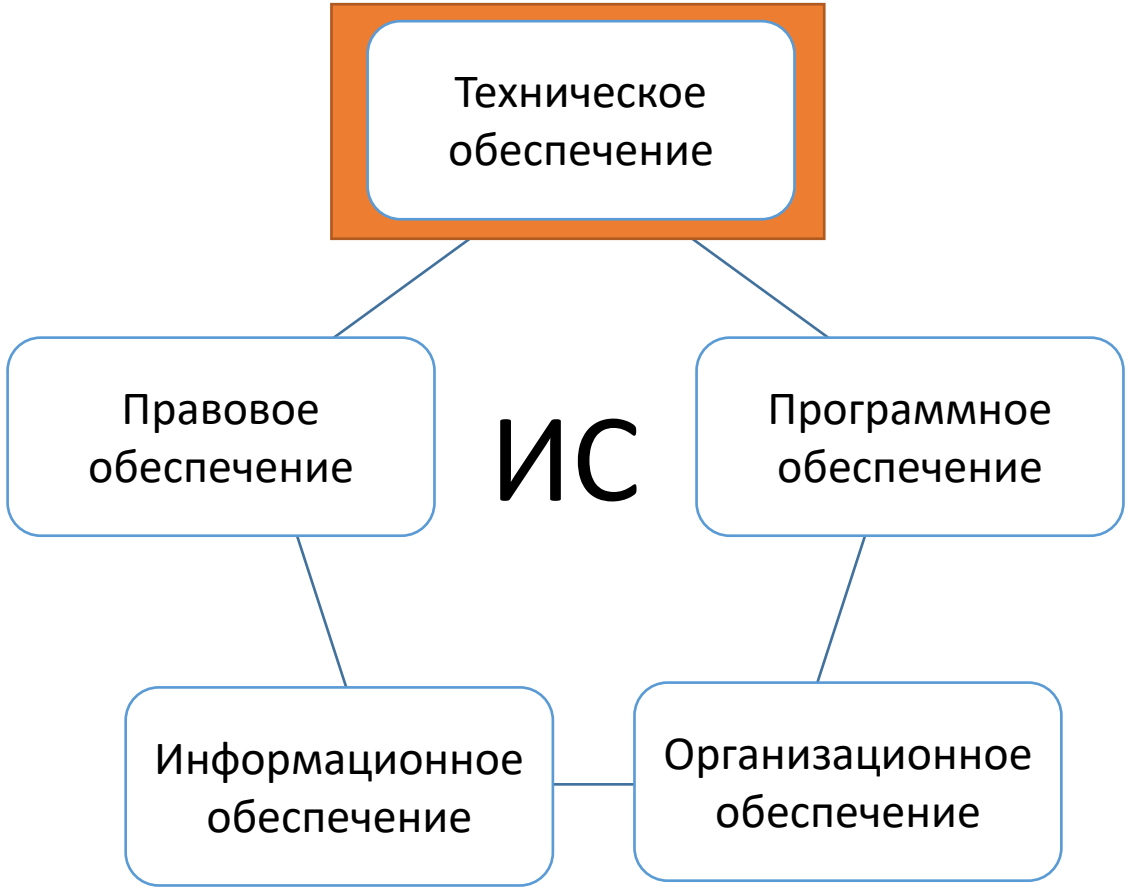
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Понятие информационной системы

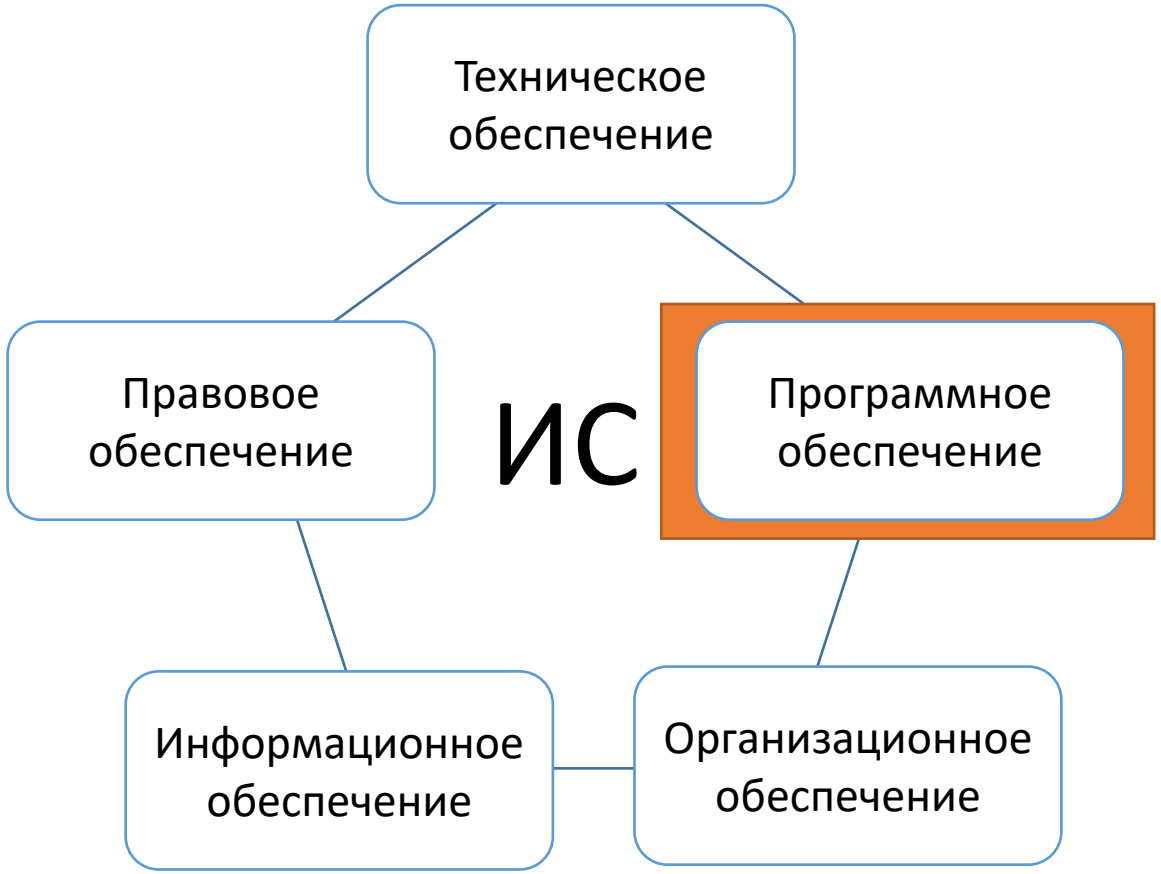


- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы и др.

sdCloud

sdLitica

Понятие информационной системы

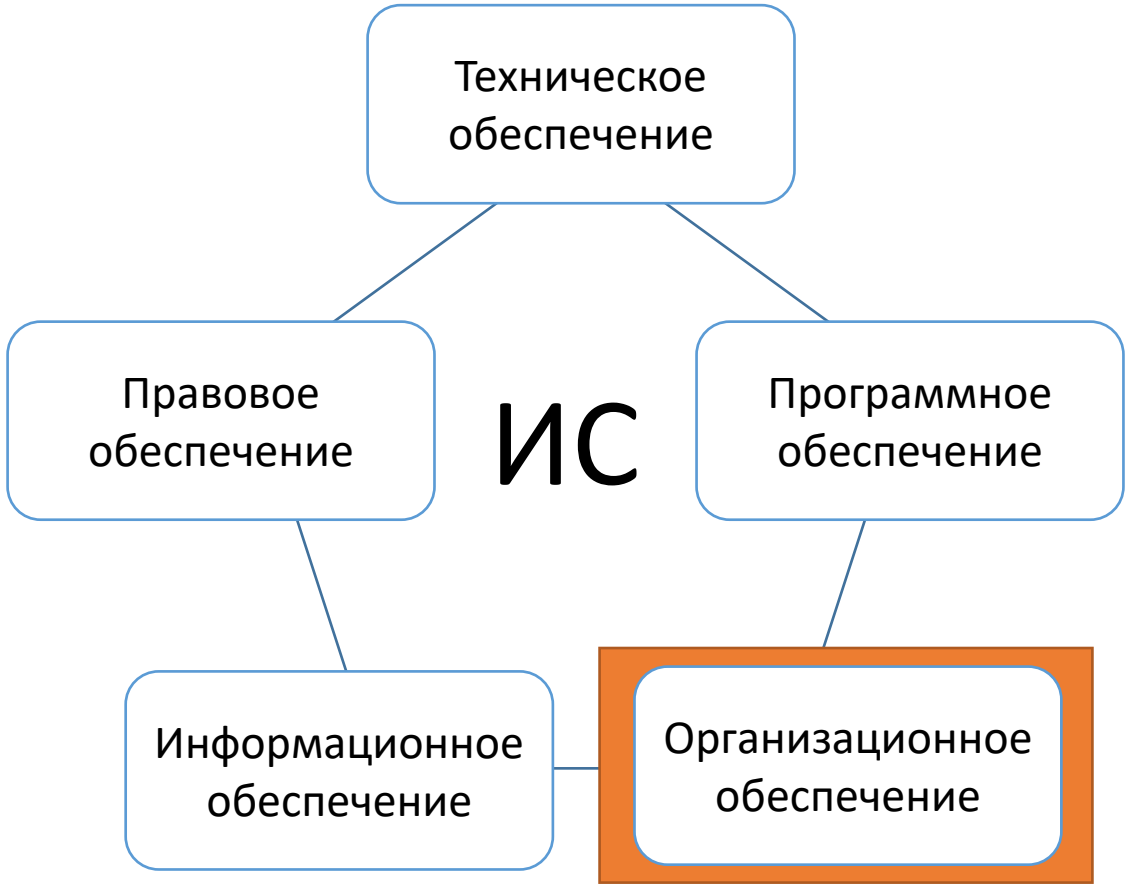


- средства моделирования процессов управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.

ITMO UNIVERSITY

insight

Понятие информационной системы



Функции:

- анализ существующей системы управления организацией и выявление задач, подлежащих автоматизации;
- подготовка задач к решению на компьютере;
- разработка управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

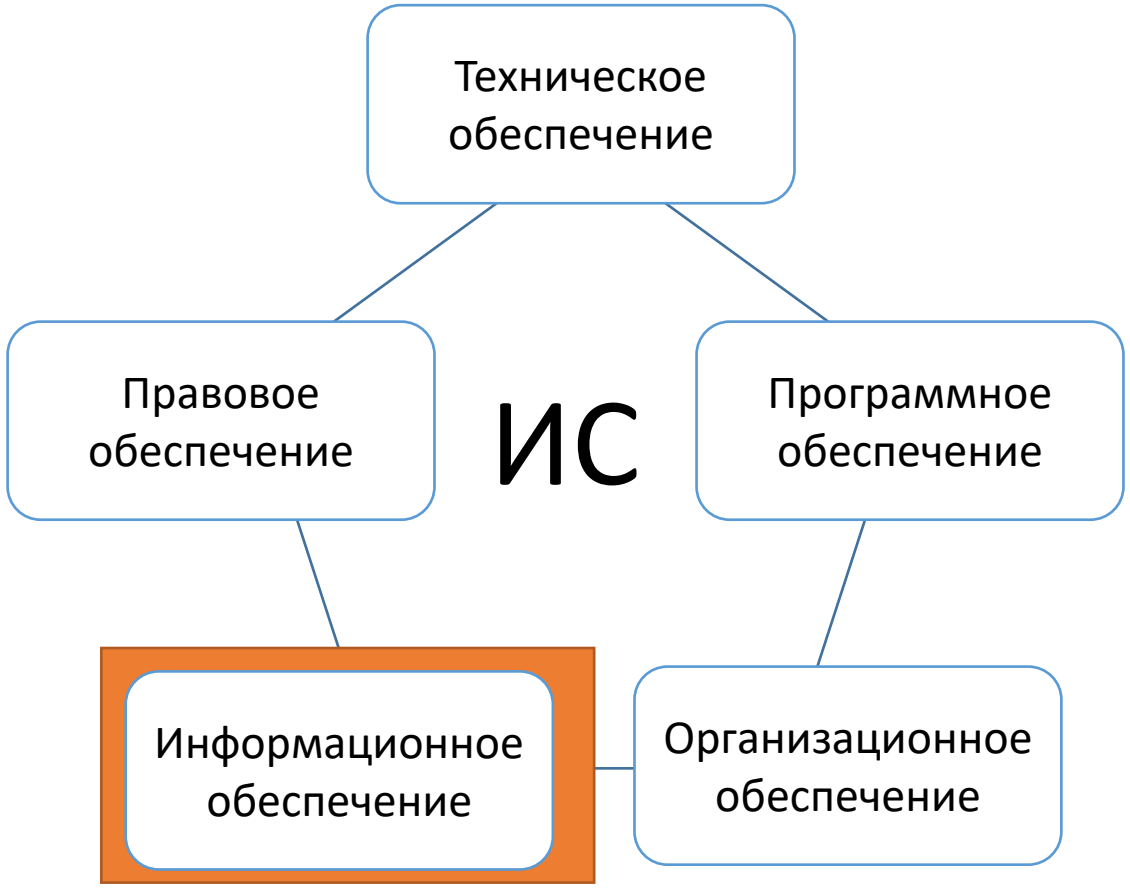
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Понятие информационной системы



Информационное обеспечение — совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных

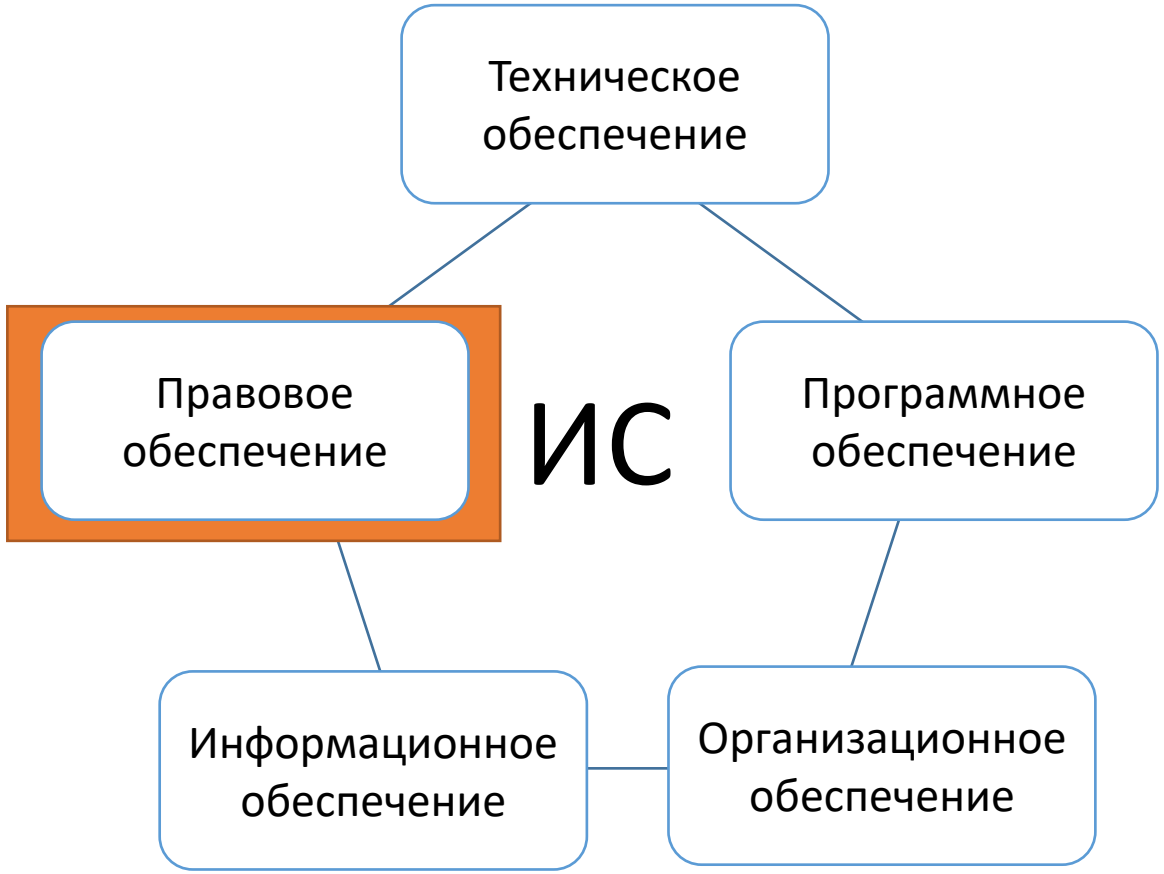
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Понятие информационной системы



Включает:

- статус информационной системы;
- права, обязанности и ответственность персонала;
- порядок создания и использования информации и др.

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Классификация ИС

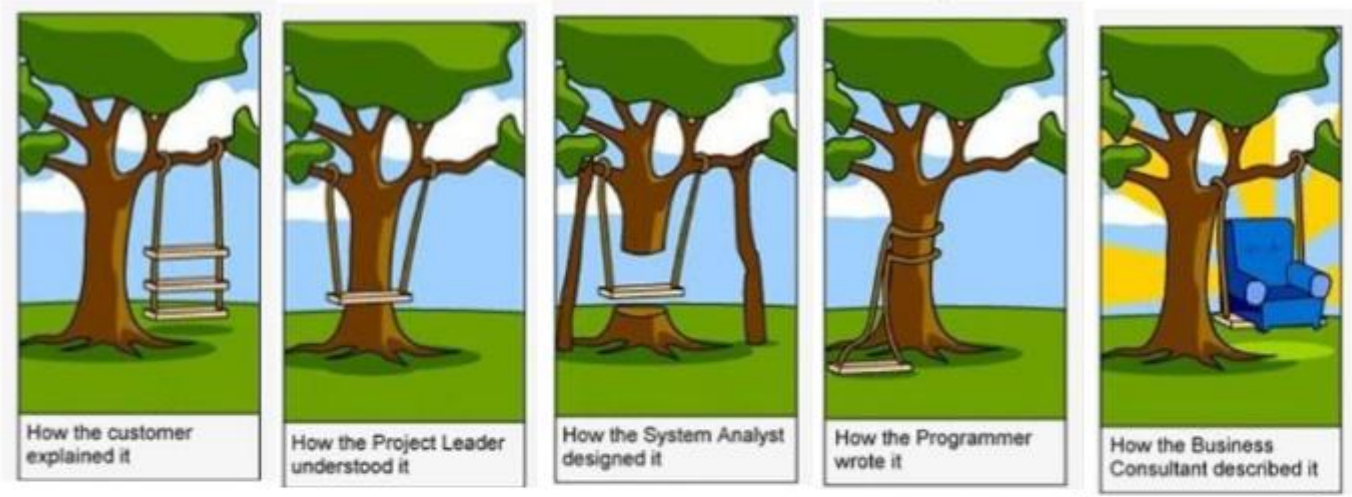
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Вспоминаем про разработку ПО



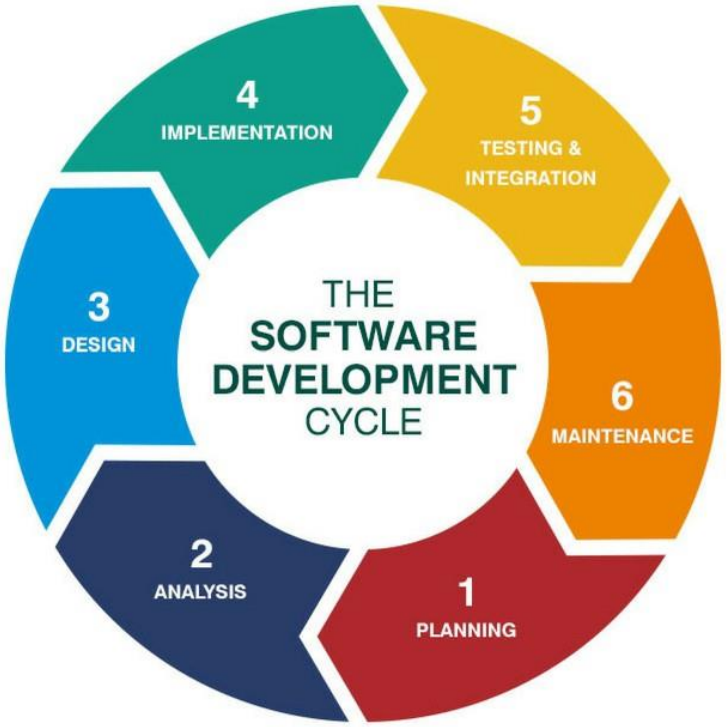
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Не забываем про то что у нас есть SDLC



sdCloud

sdLitica

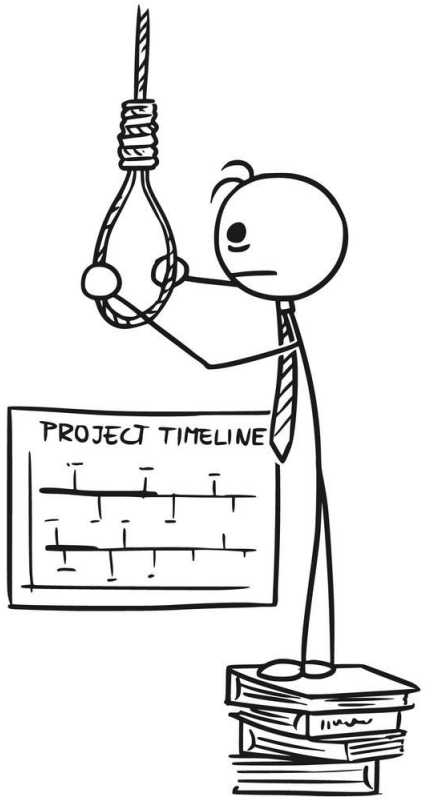
ITMO UNIVERSITY

insight

Занимательный факт

Процесс разработки RUP описывает **31** роль

Эти роли выполняют **136** активностей



sdCloud

sdLitica

ITMO UNIVERSITY

insight

Им всем надо общаться

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Им всем надо общаться

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Визуальное моделирование

- Функциональное моделирование, связано с использованием стандартов IDEF0, IDEF1X, IDEF3 ,DFD, STD и др. с использованием CASE инструментария All Function Process Modeler (BPWin)
- Объектное моделирование , связано с использованием графических схем языка UML с использованием инструментариев Rational Rose и MS Visio

sdCloud

sdLitica

ITMO UNIVERSITY

insight

И тут придумали… UML

- **U**nified **M**odeling **L**anguage – Унифицированный язык моделирования
- Открытый **стандарт** для создания абстрактной модели системы (UML-модели). Использует **графические объекты** и **текст**. Является **объектно-ориентированным**.
- **Основное назначение:** моделирование ПО, Другие: бизнес-процессы, организационные структуры, проектирование в промышленности и других отраслях.

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Как это было

- Началось всё с октября 1994 года, когда Гради Буч и Джеймс Румбах из Rational Software Corporation начали работу по разработке языка объединяющего ООП с методиками IDEF
- сформулировали следующие требования к языку моделирования:
  - Позволять моделировать не только программное обеспечение, но и более широкие классы систем и бизнес-приложений, с использованием объектно-ориентированных понятий.
  - Явным образом обеспечивать взаимосвязь между базовыми понятиями для моделей концептуального и физического уровней.
  - Обеспечивать масштабируемость моделей, что является важной особенностью сложных многоцелевых систем.
  - Быть понятен аналитикам и программистам, а также должен поддерживаться специальными инструментальными средствами, реализованными на различных компьютерных платформах

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Виды диаграмм UML

Structure Diagrams:

- Class diagram
- Component diagram
- Composite structure diagram
  - Collaboration (UML2.0)
- Deployment diagram
- Object diagram
- Package diagram
- Profile diagram (UML2.2)

Behavior Diagrams:

- Activity diagram
- State Machine diagram
- Use case diagram
- **Interaction Diagrams:**
  - Communication diagram (UML2.0) / Collaboration (UML1.x)
  - Interaction overview diagram (UML2.0)
  - Sequence diagram
  - Timing diagram (UML2.0)

Структурные диаграммы:

- Диаграмма классов
- Диаграмма компонентов
- Композитной/составной структуры
  - Диаграмма кооперации (UML2.0)
- Диаграмма развёртывания
- Диаграмма объектов
- Диаграмма пакетов
- Диаграмма профилей (UML2.2)

Диаграммы поведения:

- Диаграмма деятельности
- Диаграмма состояний
- Диаграмма прецедентов
- **Диаграммы взаимодействия:**
  - Диаграмма коммуникации (UML2.0) / Диаграмма кооперации (UML1.x)
  - Диаграмма обзора взаимодействия (UML2.0)
  - Диаграмма последовательности
  - Диаграмма синхронизации (UML2.0)

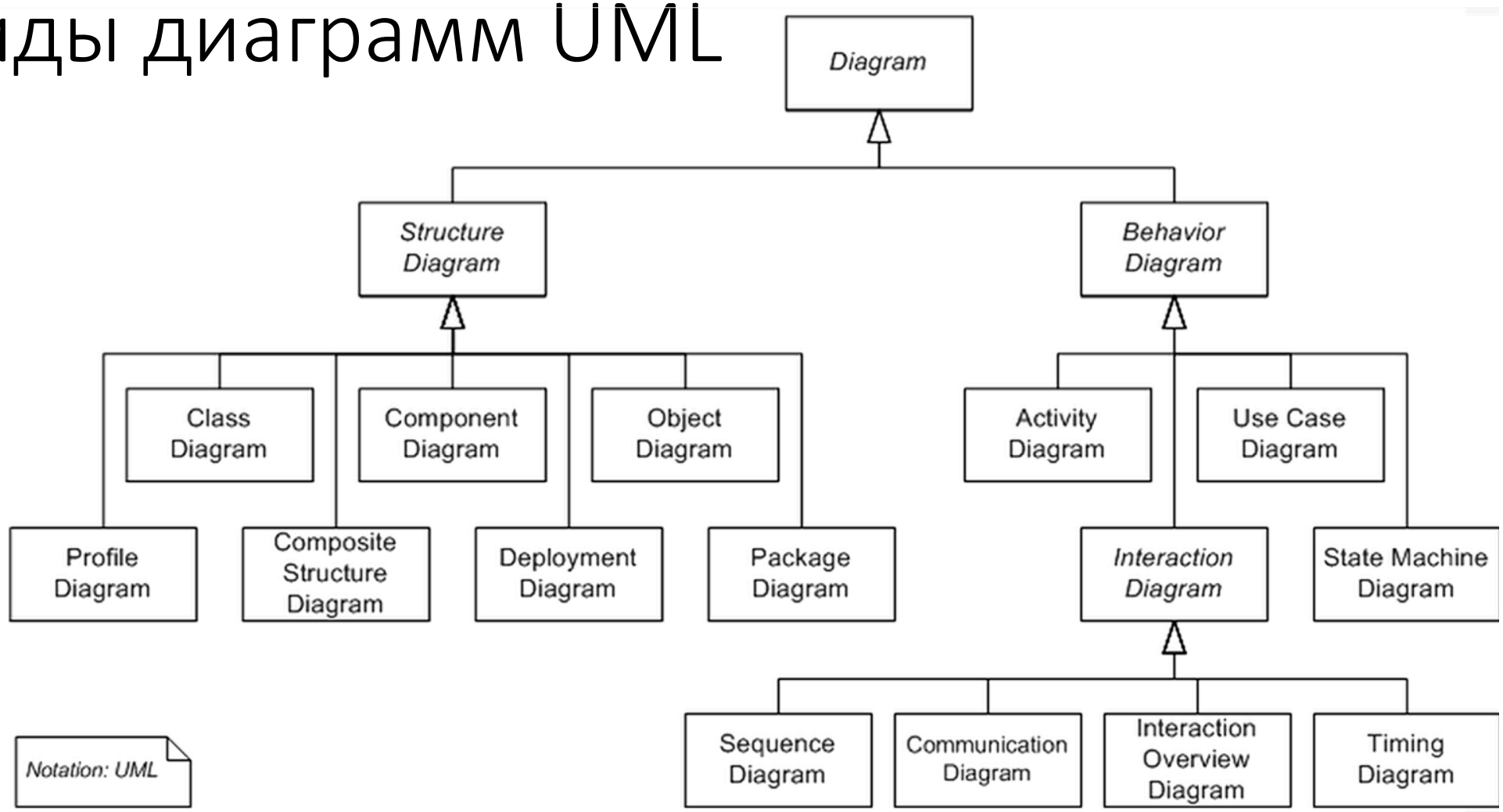
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Виды диаграмм UML



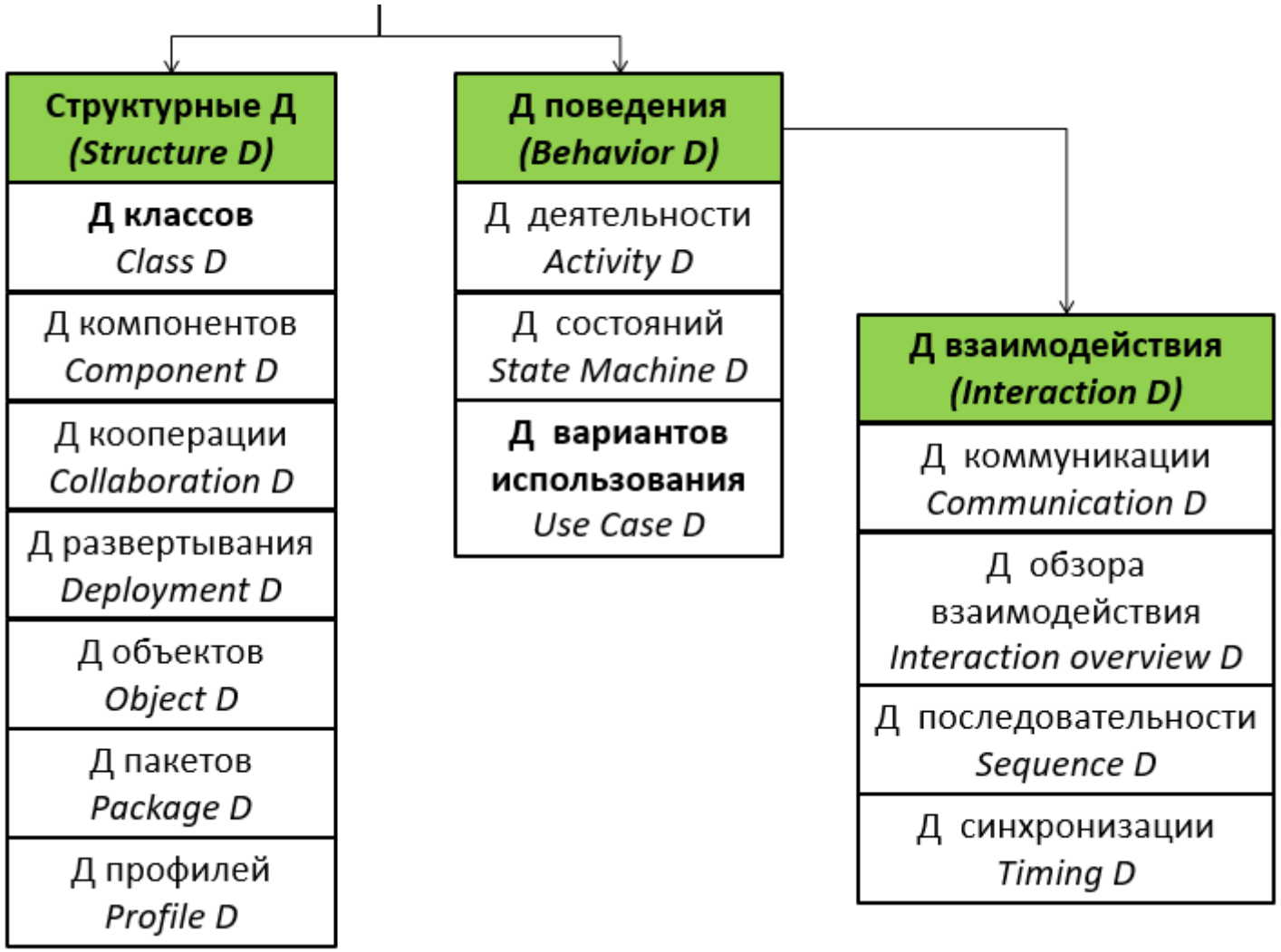
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Виды диаграмм UML



sdCloud

sdLitica

ITMO UNIVERSITY

insight

Диаграмма вариантов использования

- **Use Case, диаграмма прецедентов -** Позволяет описывать функциональные требования.
- Состоит из
  - Действующие лица (актеры, актанты, actors)
  - Варианты использования (прецеденты, Use Case)
  - Отношения (связи)
  - Граница системы

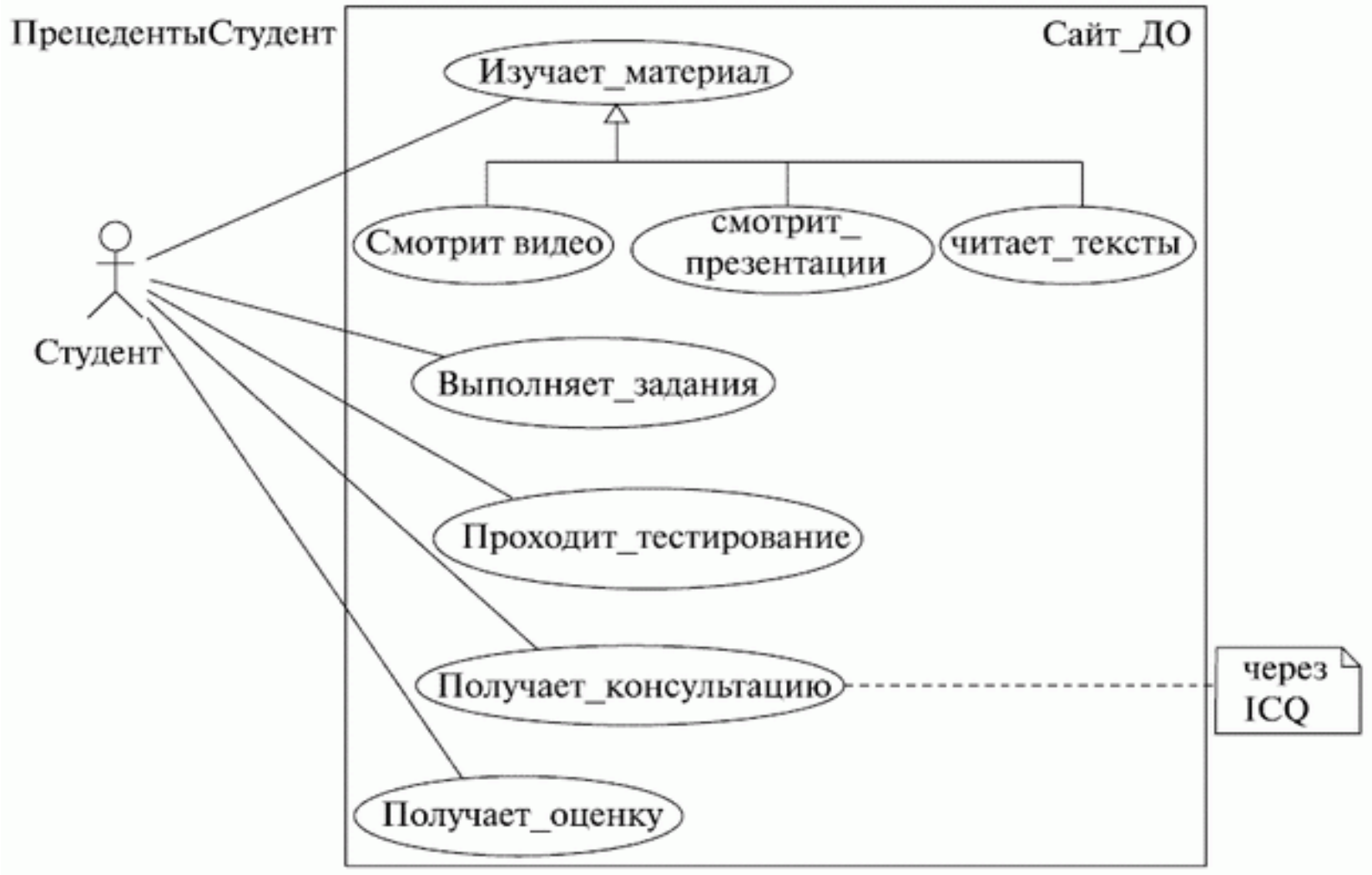
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Диаграмма вариантов использования



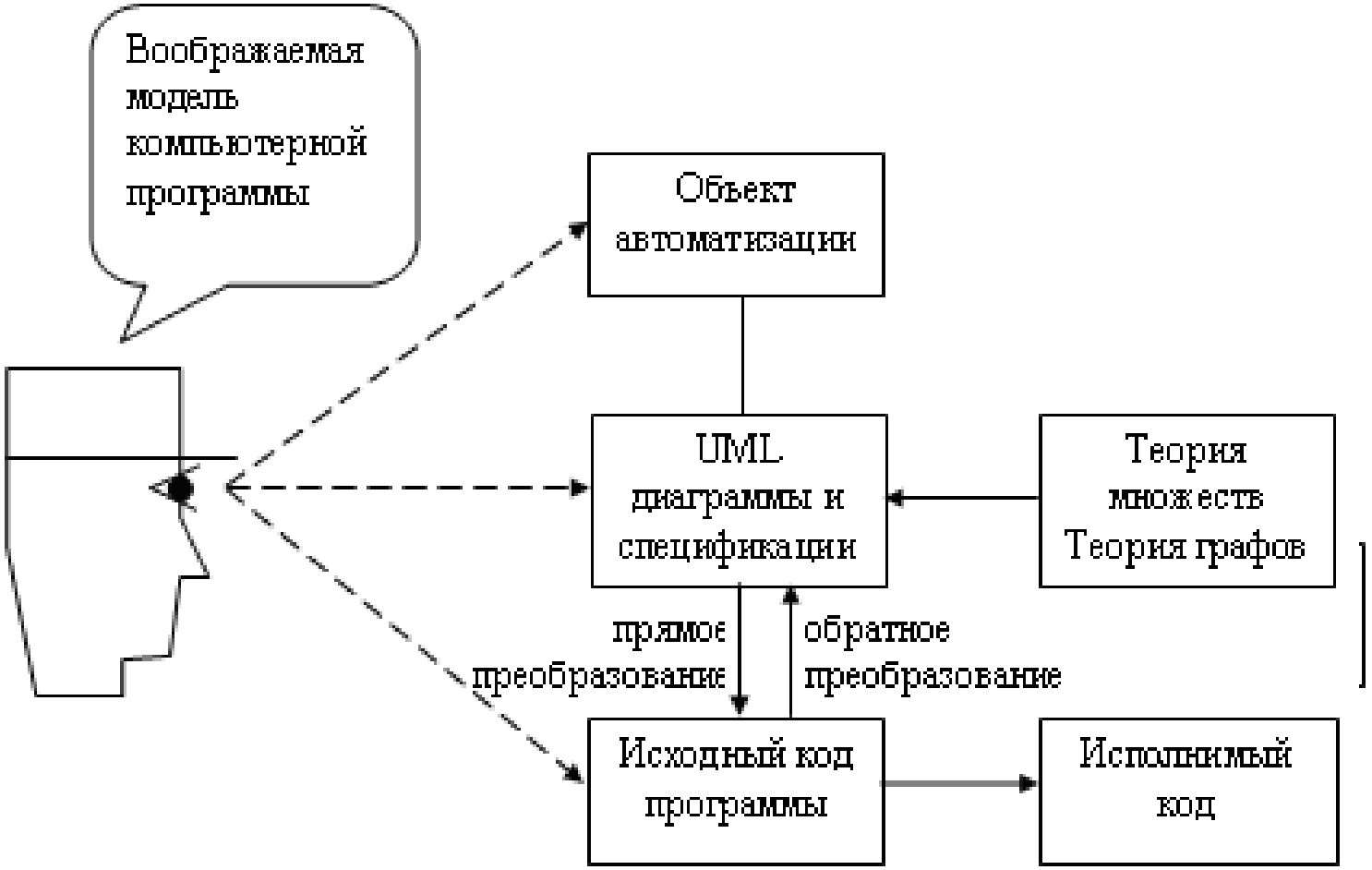
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Разработка с использованием UML



sdCloud

sdLitica

ITMO UNIVERSITY

insight

Ключевые элементы UML

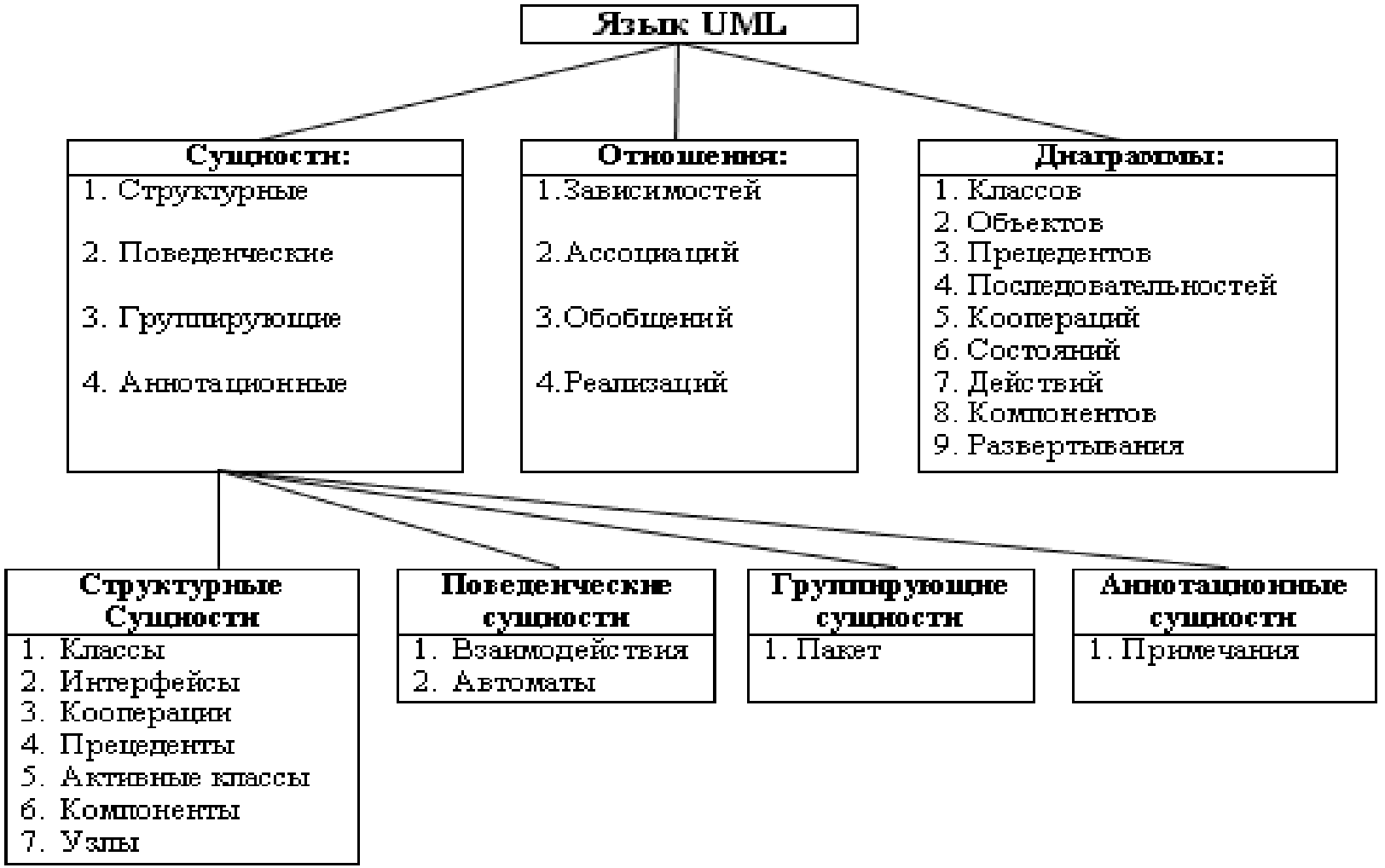
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Словарь языка UML



sdCloud

sdLitica

ITMO UNIVERSITY

insight

Действующие лица (актёры)

- Это **роли**, исполняемые сущностями, непосредственно взаимодействующими с системой. Одну роль могут исполнять разные реальные сущности.
  - Пример: роль «Разработчик» исполняется разными людьми.
- Кто такие актеры:
  - Кто или что использует систему?
  - Кто устанавливает систему?
  - Кто или что запускает и выключает систему?
  - Кто обслуживает систему?
  - Какие системы взаимодействуют с данной системой?
  - Кто или что получает и предоставляет информацию системе?
  - Происходит ли что-нибудь в точно установленное время?

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Вариант использования (прецедент)

- Это **описание последовательности действий**, включая альтернативные последовательности (ошибки и исключения), которые система, подсистема или класс могут осуществлять, взаимодействуя с внешними актерами.
- Что **должна делать система** для конкретного актера, «вариант использования» системы актером.
- Обозначается **глаголом** или отглагольным существительным (единообразно на одной диаграмме).
  - Как найти прецеденты:
  - Зачем нужна система каждому актеру?
  - Что вообще должна делать система?

  При необходимости добавляются новые актеры.

sdCloud

sdLitica

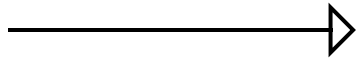
ITMO UNIVERSITY

insight

Отношения

Ассоциация (ненаправленная связь, между актером и вариантом использования)

Направленная ассоциация (если надо показать направление взаимодействия)



Обобщение (от частного к общему). Пример: администратор и гость являются частными случаями пользователя системы

Зависимость (от зависимого элемента к целевому, изменение целевого влияет на зависимый)

Включение (от целого к части). Пример: авторизация пользователя включает ввод пароля

Расширение (от дополнительного действия к основному). Пример: вариант использования «Запомнить пользователя» расширяет вариант использования «Авторизация пользователя»

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Нужна ли мне диаграмма прецедентов?

- **ДА, если:**
  - преобладают функциональные требования;
  - много типов пользователей (много актеров);
  - в системе много интерфейсов (много актеров).
- **НЕТ, если:**
  - преобладают нефункциональные требования;
  - в системе мало пользователей;
  - в системе мало интерфейсов.

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Диаграмма классов

- Одна из наиболее часто используемых. Назначение диаграммы может быть разным.
- Точки зрения на построение диаграммы в зависимости от цели применения:
- **Концептуальная** т.зр. – описывает модель предметной области, в ней присутствуют только классы прикладных объектов;
- Т. зр. **спецификации** – применяется при проектировании ИС и БД, может использоваться вместо или вместе с ER-моделью;
- Т. зр. **реализации** – диаграмма классов содержит классы, используемые непосредственно в программном коде (в ООП).
- Содержит **классы**, их **атрибуты** и **методы**, **взаимосвязи** классов, **объекты** и др.
- Классы должны отражать сформулированные требования к ПО.

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Диаграмма классов - термины

- **Класс** – множество однотипных объектов, обладающих общим набором свойств (атрибутов) и методов поведения. Описание класса включает имя, набор атрибутов и набор методов.
- **Объект (экземпляр класса)** – представитель класса с конкретными значениями атрибутов. Обычно объект имеет собственное имя.
- **Атрибут (свойство)** – именованное свойство, присущее каждому объекту класса.
- **Метод (поведение)** – действие, которое может выполнять любой объект данного класса.

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Что такое хороший класс?

- имя класса о…но перечислять атрибуты и ме…ласс)
- класс имеет о…ность) в системе, очень редко 2…
- имеет сильны…методы не могут существовать,
- имеет слабые…ы других классов могут быть пр…

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Класс[ные] рекомендации

- 3-5 методов в 1 классе
- не должен быть изолированным (иметь связи с другими классами)
- избегать глубокой иерархии наследования

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Аттрибуты

- [<видимость>] <имя> [: <тип> [[<размерность>]] [= <значение>]]

Видимость

+	Public
-	Private
#	Protected

Основные типы:
- Integer
- Real
- String
- Boolean
- ...

Размерность = множественность = кратность
[*n*] – ровно *n* штук
[*a*..*b*] – от *a* до *b* штук
[*a*..*] – от *a* до ∞ штук

Значение – по умолчанию, начальное, общее для всего класса.

Госномер: символ [9]

RegNum: Literal [9]

Номер телефона: строка [1..*]

Phone: String[1..*]

Кол-во детей: целое = 0

Children: Integer = 0

Наличие задолженности: логический = нет

HasDebt: Boolean = false

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Методы

- [<видимость>] <имя> **(**{<параметр>,}**)** [**:** <тип>]

<параметр> ::=
[<направление>] <имя> [**:** <тип> [**=** <значение>]]

Направление

in	Входной
out	Выходной
inout	Входной и выходной
return	Возвращаемое значение

Открыть (ИмяФайла: Строка): Логический

Open (FileName: String): Boolean

Переместить (Из, В: Адрес): Логический

Move (From, To: TAdress): Boolean

Езда(in Скорость: Вещественный, inout ПунктНазначения: Строка)

Drive (in Speed: Real, inout Destination: String)

sdCloud

sdLitica

ITMO UNIVERSITY

insight

Отношения между классами

- ассоциация
- обобщение
- агрегация
- композиция
- зависимость

- **Кратность** отношений:
- 0..1
- 1
- n
- 1..n
- n..*
- *

sdCloud

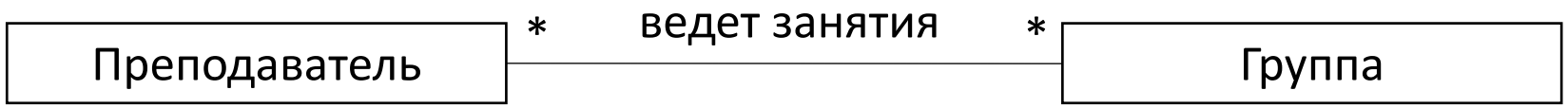
sdLitica

ITMO UNIVERSITY

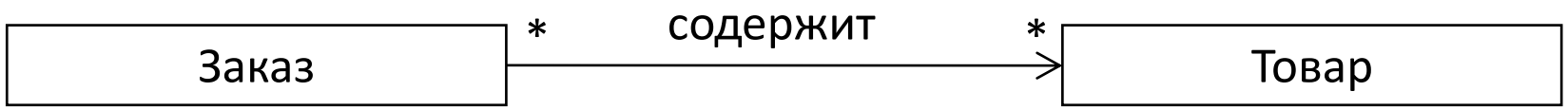
insight

Ассоциация

«Обычная» связь между равноправными независимыми классами.



Возможность навигации – класс, от которого идет стрелка «знает» про другой класс, но не наоборот.



sdCloud

sdLitica

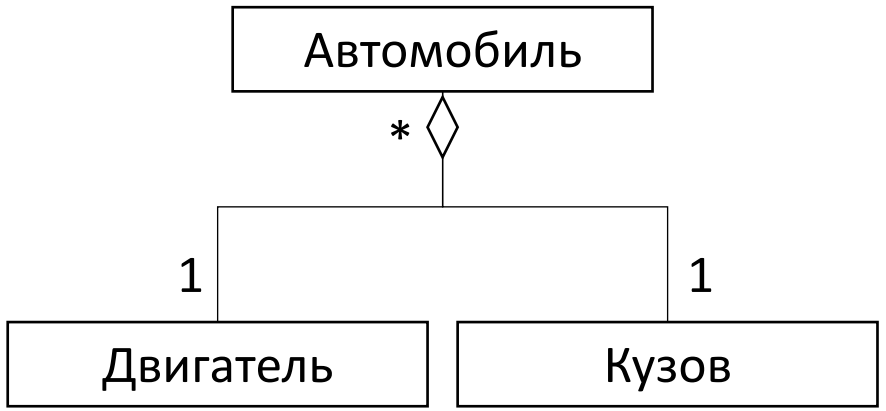
ITMO UNIVERSITY

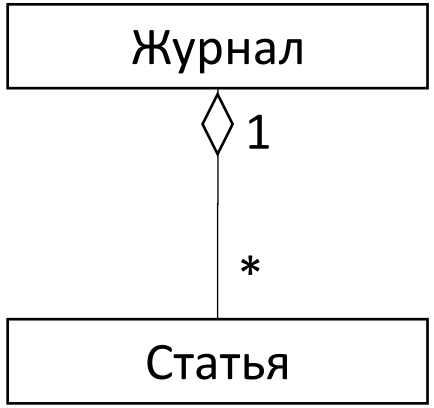
insight

Агрегация

A Part Of = APO

Класс-контейнер содержит в себе другие классы-части.





sdCloud

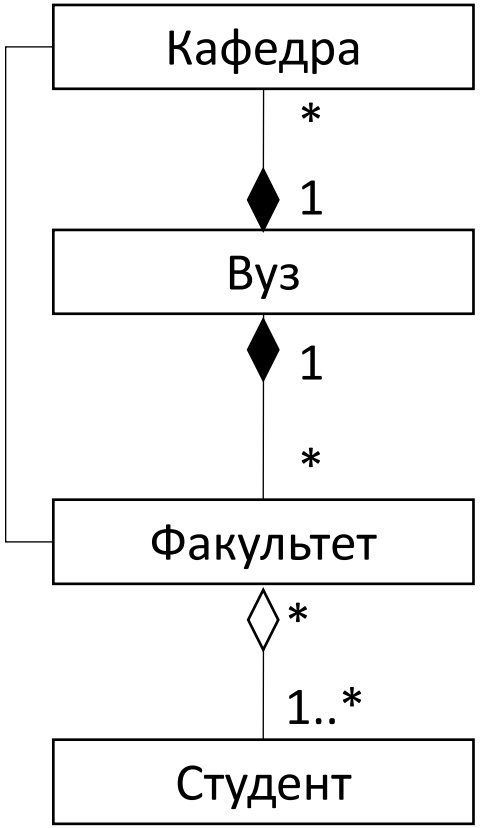
sdLitica

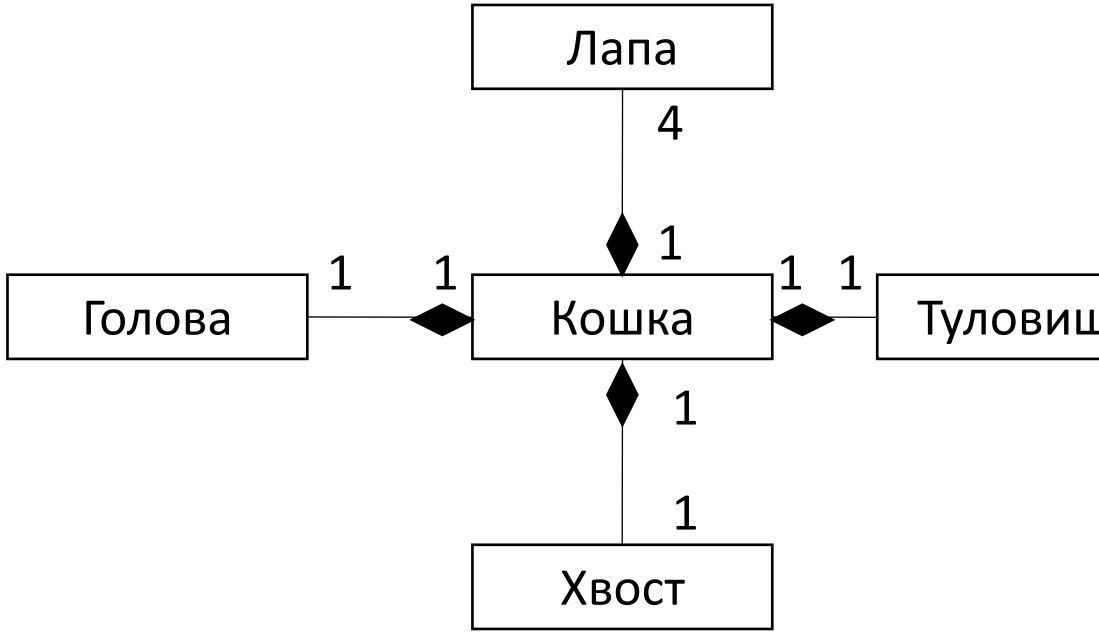
ITMO UNIVERSITY

insight

Композиция

«Строгая» агрегация: части не могут существовать без контейнера





Со стороны целого не может быть кратность 0!

sdCloud

sdLitica

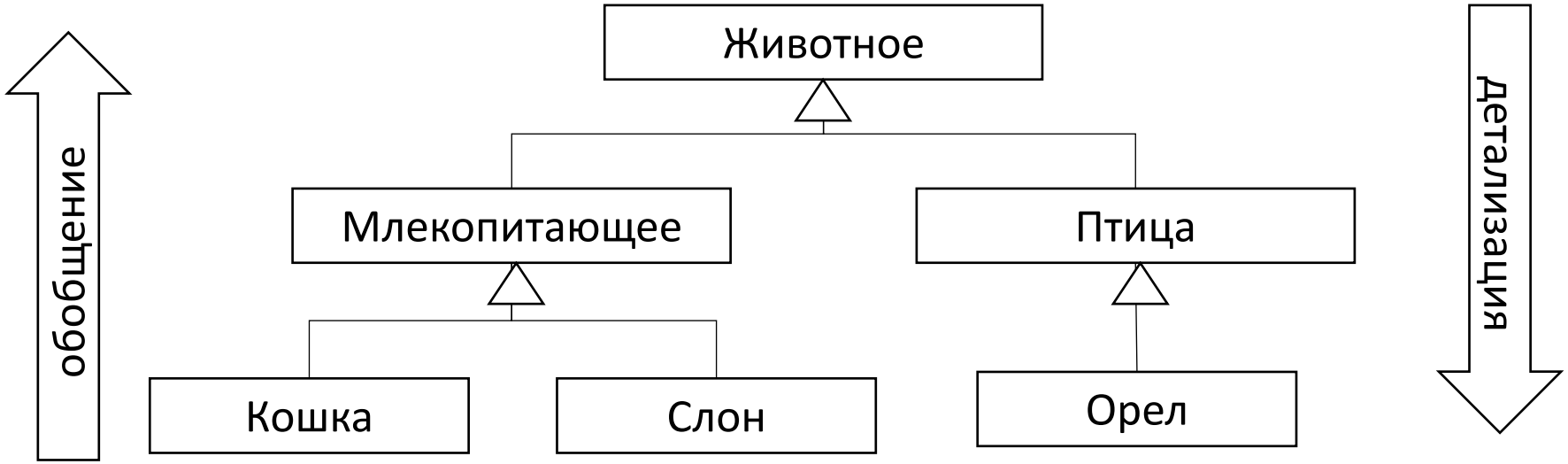
ITMO UNIVERSITY

insight

Обобщение (наследование)

A Kind Of = AKO

Класс-наследник является частным случаем класса-предка.



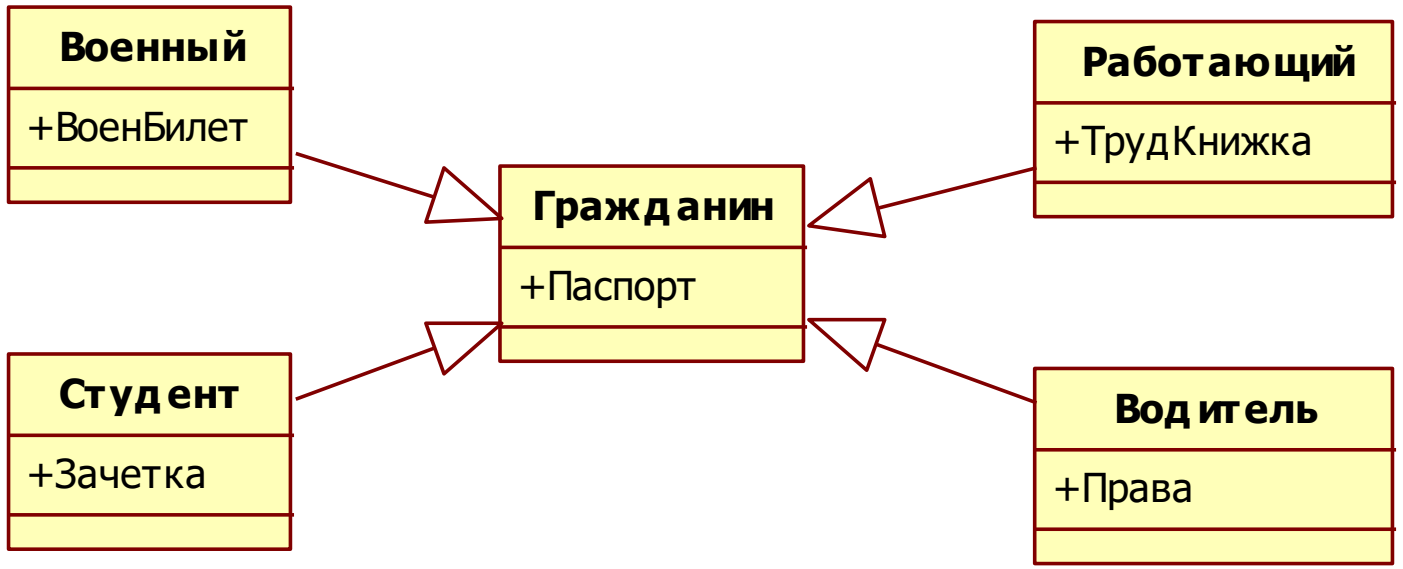
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Обобщение (наследование)



sdCloud

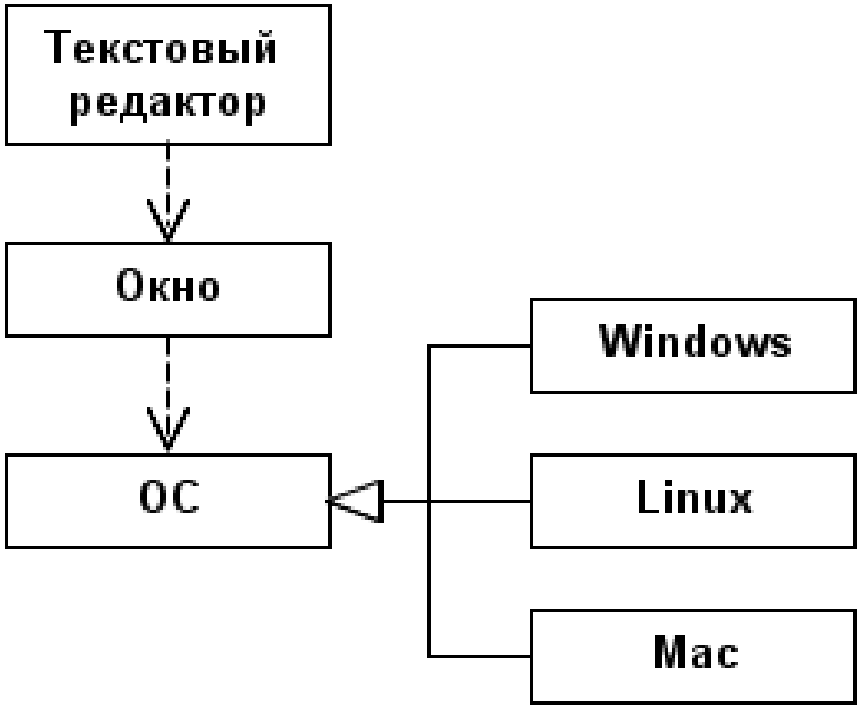
sdLitica

ITMO UNIVERSITY

insight

Зависимости

Изменение основного класса приводит к изменению зависимого, хотя он не является частью или наследником.



sdCloud

sdLitica

ITMO UNIVERSITY

insight

Стереотипы классов

Стереотип указывается в «» или <<>> перед именем класса или специальным символом.

Существует множество стереотипов. При разработке ИС рекомендуется использовать:

«boundary»	Граничный класс	Осуществляет взаимодействие с пользователем или другими системами (форма, сокет)
«entity»	Сущностный класс	Соответствует сущностям ER-модели. Обычно не имеет методов, только содержит данные
«control»	Управляющий класс	Выполняет активные действия в системе, реализует ее функции (СУБД, клиентское приложение, сервер)

Читатель

=

«entity»
Читатель

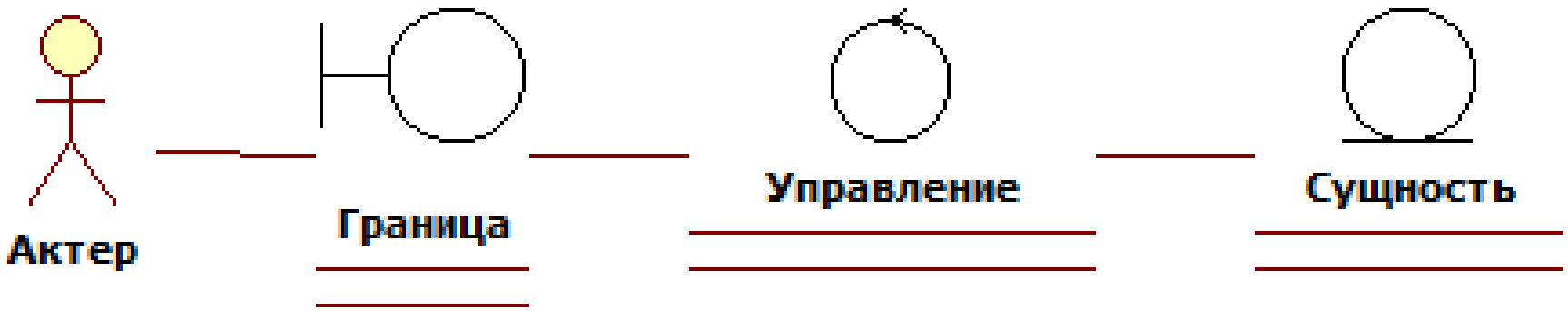
sdCloud

sdLitica

ITMO UNIVERSITY

insight

Взаимодействие



sdCloud

sdLitica

ITMO UNIVERSITY

insight

Спасибо!
Вопросы?

sdCloud

sdLitica

ITMO UNIVERSITY

insight

References/Acknowledgments

https://pptcloud3.ams3.digitaloceanspaces.com/presentations/ppts/000/228/438/original/mHcwgtod4Q0DXYbda_FbOA.pptx?1497281602

https://kornast.ucoz.ru/KSUvSAPiB/UML.pptx

http://portal.tpu.ru:7777/SHARED/t/TSIV/uchebn/TIPS/2.%20UML%20%D0%B8%20%D0%B2%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5.pptx

sdCloud

sdLitica