

ІІТМО

Практическая работа

5

1. Используя шаблон курсовой работы и правила оформления работ (ГОСТ 7.32), создать отчет по ПР5.
 - Титульный лист – отчет по практической работе 5
 - Задания – нет
 - Введение – пара фраз о реальном содержании работы.
 - Основная часть должна содержать краткое описание предметной области функционирования и основных пользователей системы.
 - Для каждого из пользователей необходимо представить диаграмму вариантов использования системы на языке UML.
 - Для ключевых прецедентов представить диаграммы активности.
 - Рассмотреть альтернативные потоки событий для основных прецедентов.
 - В качестве инструмента необходимо использовать любой инструмент, поддерживающий построение моделей на языке UML.
 - Заключение – вывод по работе.
 - Список литературы должен содержать ссылки на все источники, использованные в работе.



- UML (Unified Modeling Language) –
Унифицированный Язык Моделирования
- Разработан группой объектного проектирования
OMG (Object Management Group)
- Получил статус отраслевого стандарта



Авторы UML

- Гради Буч (Grady Booch)
- Джеймс Румбах (James Rumbaugh)
- Айвар Якобсон (Ivar Jacobson)



Цели создания UML

- Предоставить пользователям готовый к использованию язык визуального моделирования
- Предоставить механизмы расширения и специализации
- Быть независимым от определенного языка программирования и процесса разработки
- Интегрировать лучший практический опыт разработок



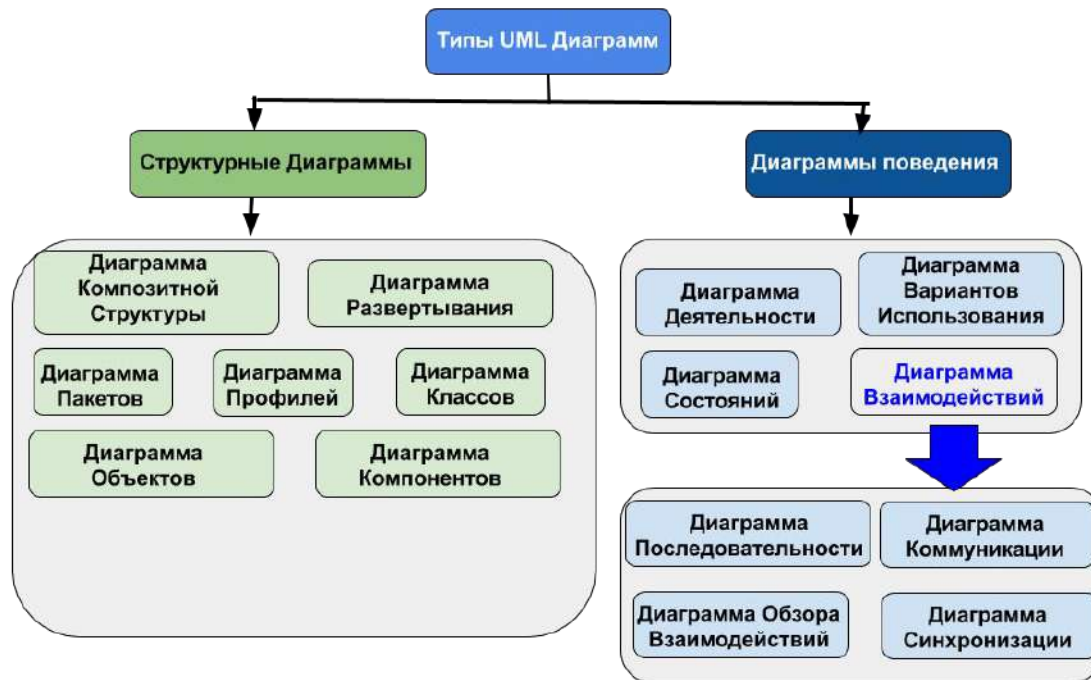
Диаграммы языка UML

Диаграммы языка UML

- вариантов использования или прецедентов (use case diagram)
- классов (class diagram)
- состояния (statechart diagram)
- активности (**activity diagram**)
- последовательности (sequence diagram)
- взаимодействия (collaboration diagram)
- компонентов (component diagram)
- развертывания (deployment diagram)
- композитная структурная диаграмма
- обзорная диаграмма взаимодействия
- временная диаграмма
- диаграмма пакетов



UML диаграммы



Зачем в UML столько диаграмм?

ІТМО

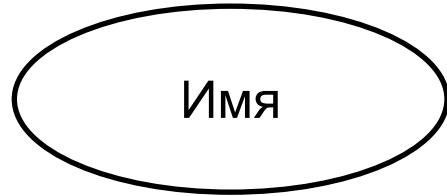
- Причина этого заключается в том, что можно взглянуть на систему с разных точек зрения ведь в разработке программного обеспечения будут участвовать многие заинтересованные стороны, такие как: аналитики, конструкторы, кодеры, тестеры, контроль качества, клиенты, технические авторы.
- Все эти люди заинтересованы в различных аспектах системы, и каждый из них требует разного уровня детализации.
- Например, кодер должен понимать проект системы и уметь преобразовывать проект в код низкого уровня.
- Напротив, технический писатель интересуется поведением системы в целом и должен понимать, как функционирует продукт.
- UML пытается предоставить язык настолько выразительным образом, что все заинтересованные стороны могут извлечь выгоду, как минимум из одной диаграммы UML.

Диаграмма прецедентов

- Диаграммы прецедентов описывают функциональное назначение системы (то, что система будет делать в процессе своего функционирования)
- Диаграммы прецедентов являются исходной концептуальной моделью системы в процессе ее проектирования и разработки



Диаграмма прецедентов: элементы



Прецедент



Прецедент – фрагмент поведения ИС без раскрытия его внутренней структуры

Прецедент – сервис, который информационная система предоставляет пользователю (актеру)

Диаграмма прецедентов: Прецедент. Пример

```
graph TD; UC1([Создать карту визита]); UC2([Получить список свободных номеров]); UC3([Проверить наличие клиента в черном списке]);
```

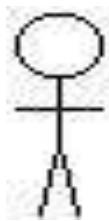
Создать
карту визита

Получить список
свободных
номеров

Проверить наличие
клиента в черном
списке

Диаграмма прецедентов: элементы

Актер



Имя

Актер представляет собой любую внешнюю по отношению к моделируемой ИС сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей

Диаграмма прецедентов: актер

Пример

ІТМО



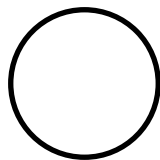
Дежурный
администратор



Менеджер

Диаграмма прецедентов: элементы

Интерфейс



Имя

Интерфейс определяет совокупность операций, которые обеспечивают необходимый набор сервисов для актера

Диаграмма сценариев: элементы ИТМО

Примечание



Примечание предназначено для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта

Диаграмма сценариев: примечание

Пример

ИТМО

Проверить наличие
клиента в черном
списке

Проверка
выполняется
только по
фамилии клиента



Менеджер

менеджер
может только
просматривать
информацию

Диаграмма сценариев: отношения



- отношение ассоциации (association)
- отношение включения (include)
- отношение расширения (extend)
- отношение обобщения (generalization)

Диаграмма сценариев: ассоциация

ІТМО

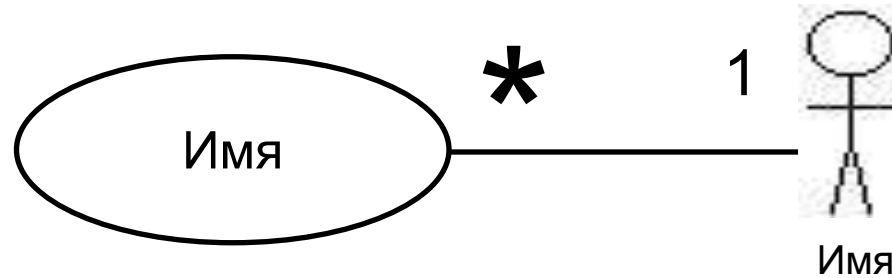


Диаграмма сценариев: ассоциация. Пример

ІТМО

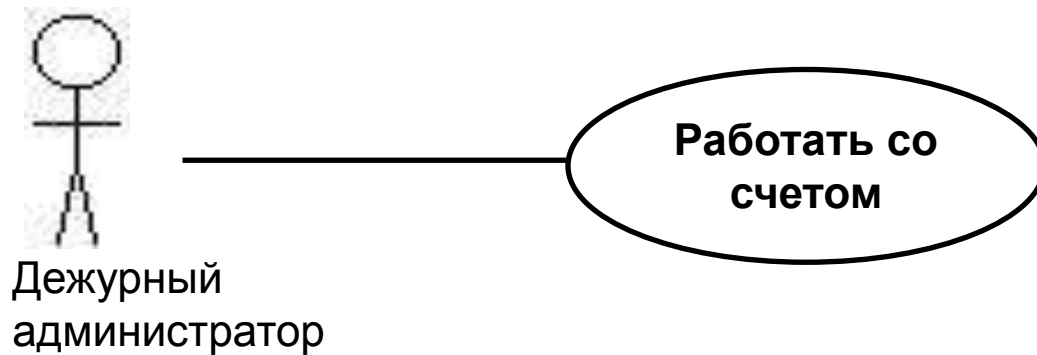
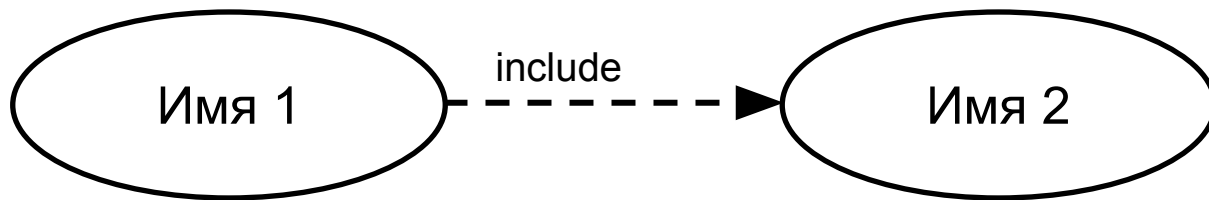


Диаграмма сценариев: включение

ІТМО



Сценарий 1 включает сценарий 2

Диаграмма сценариев: включение. Пример

ІТМО

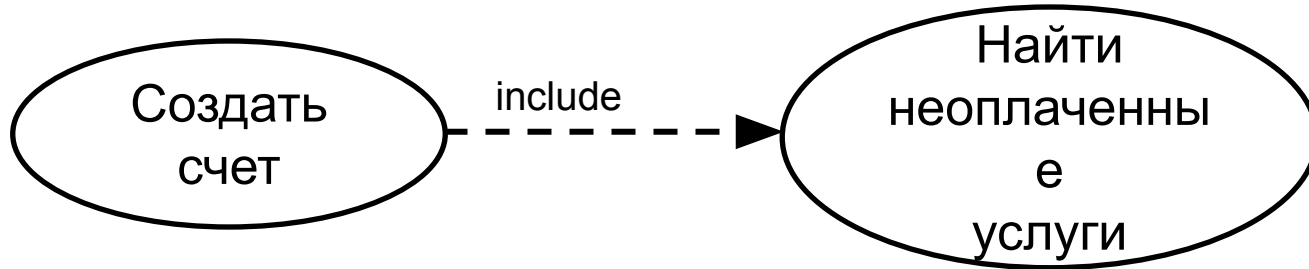
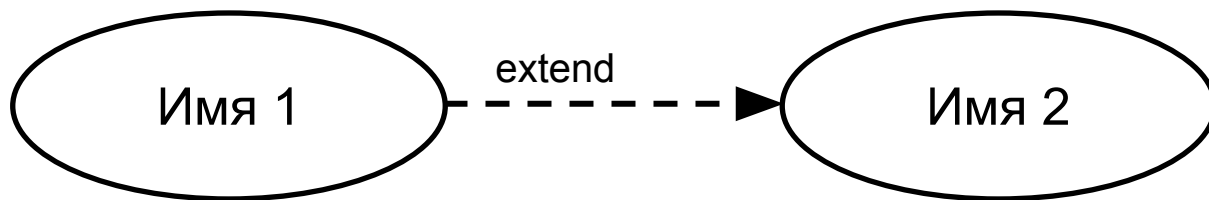


Диаграмма сценариев: расширение

ІТМО



Сценарий 1 расширяет сценарий 2

Диаграмма сценариев: расширение. Пример

ІТМО

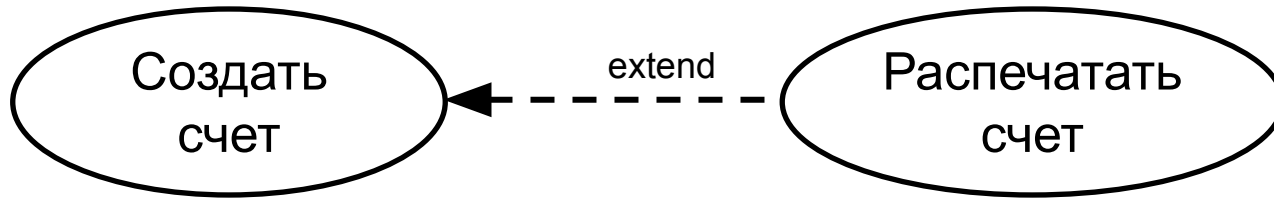
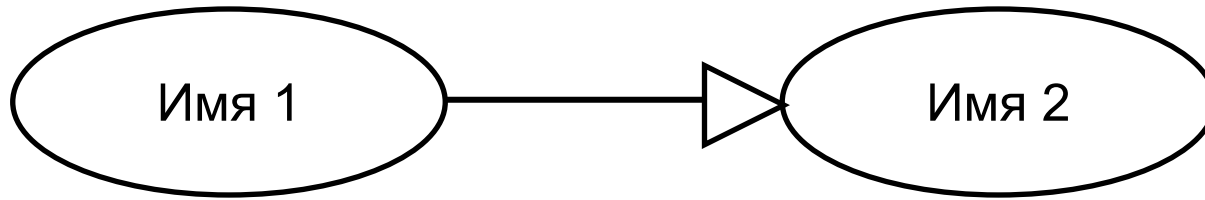


Диаграмма сценариев: обобщение

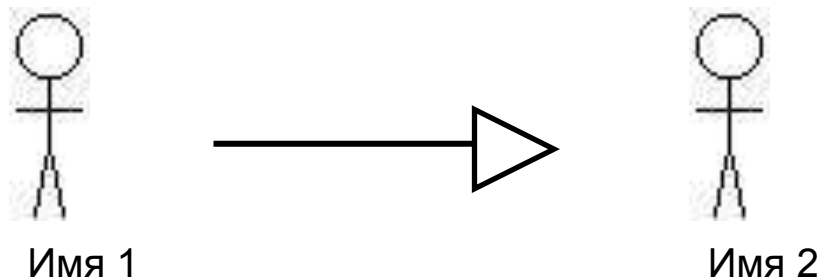
ІТМО



Сценарий 2 обобщает сценарий 1

Диаграмма сценариев: обобщение. Пример

ІТМО



Актер 2 обобщает Актера 1

Диаграмма сценариев: интерфейс

ІІТМО

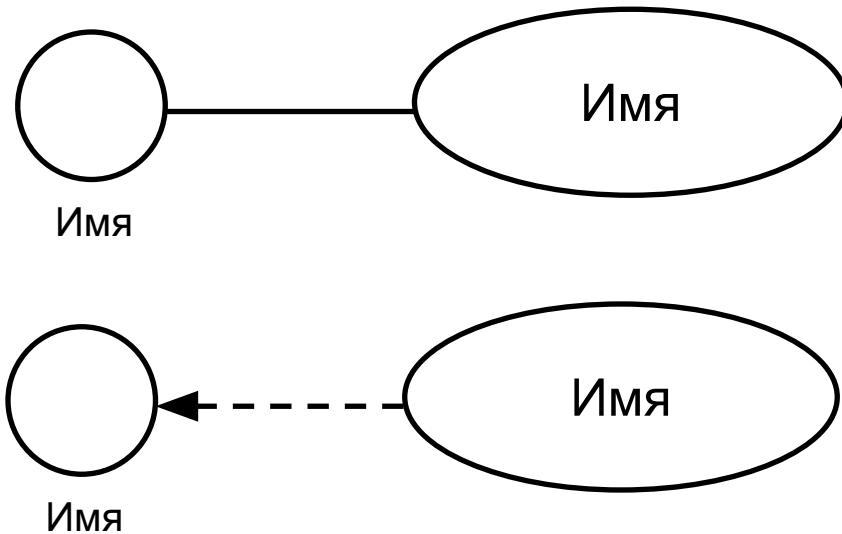


Диаграмма сценариев: интерфейс. Пример

ІІТМО

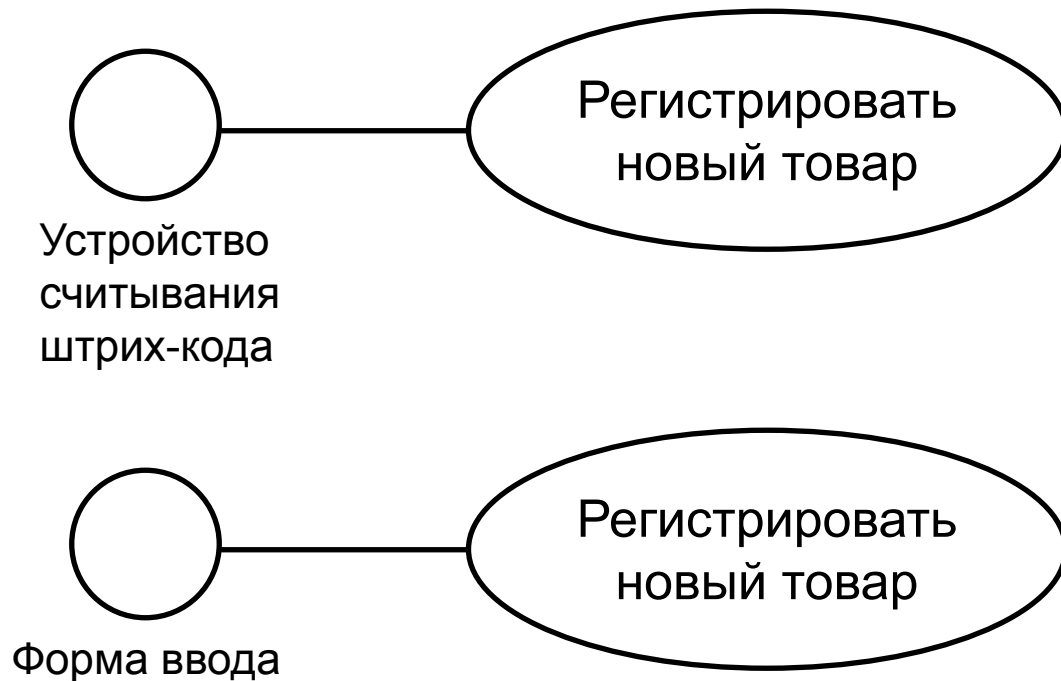


Диаграмма сценариев

Пример

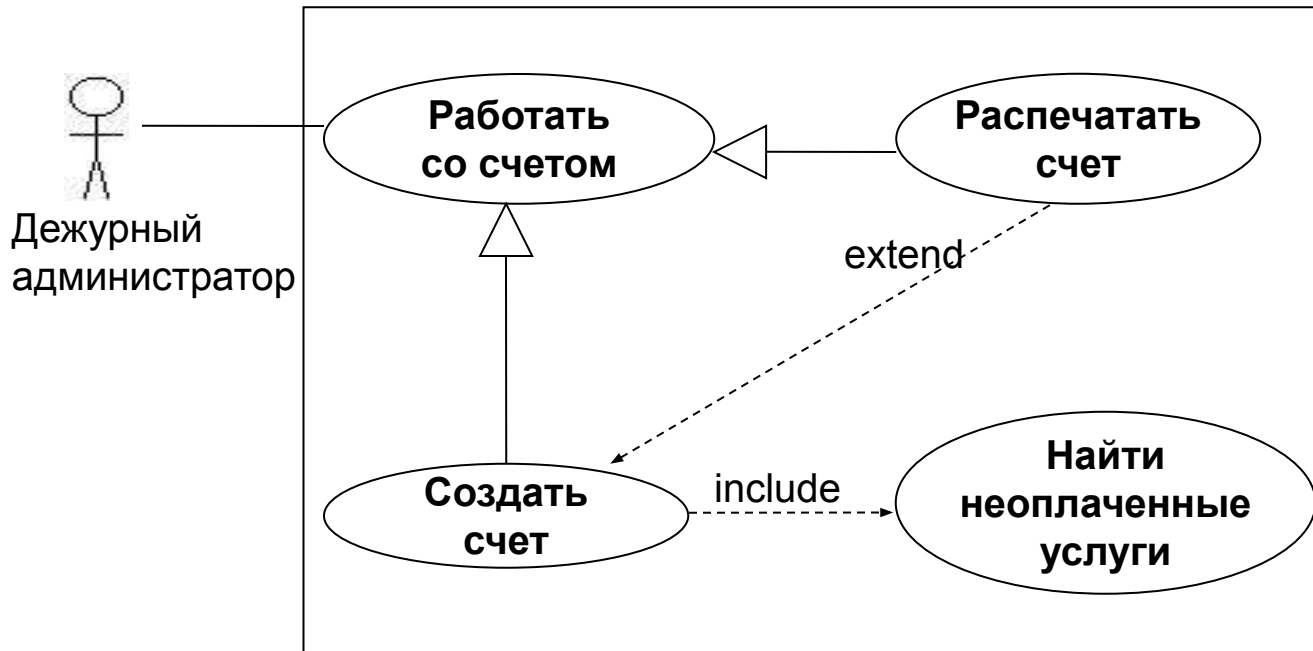


Диаграмма прецедентов

Диаграмма Прецедентов

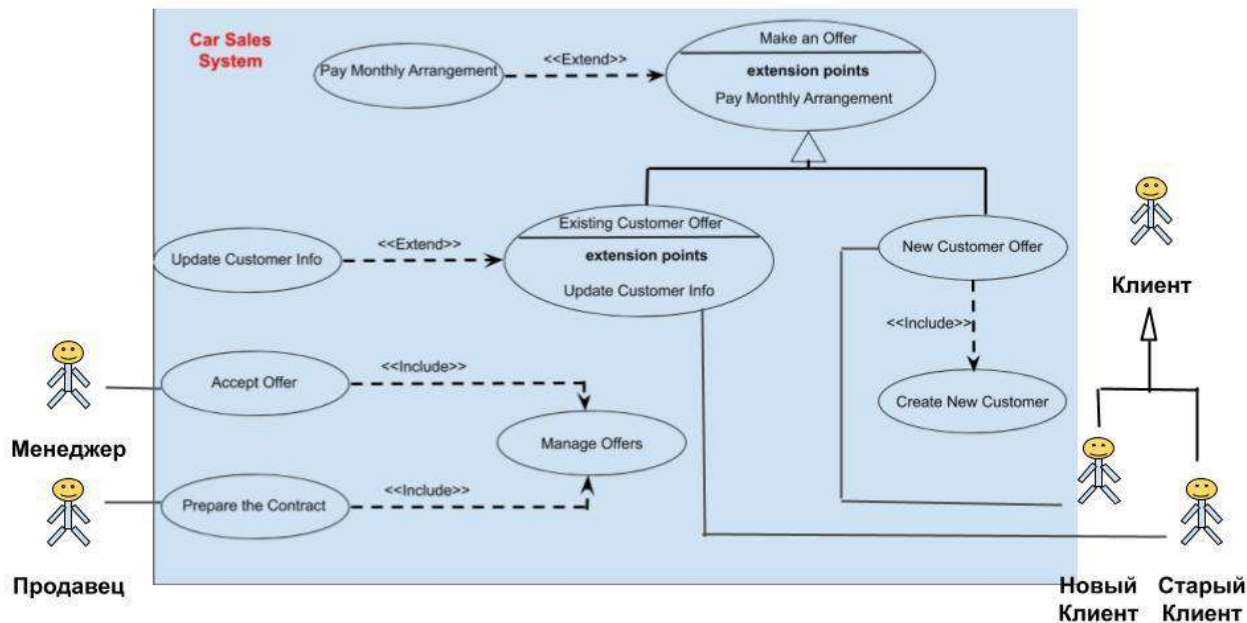


Диаграмма деятельности: определение



- Диаграмма деятельности описывает процесс выполнения действий, т.е. логику или последовательность перехода от одного действия к другому
- Диаграмма деятельности используется для моделирования бизнес-процессов

Диаграмма деятельности: элементы

ІТМО

Действие



Имя

Действие – операция,
выражение, вычисления и т.д.

Диаграмма деятельности: действие. Пример

ІТМО



Выполнить запрос

$i = i + 1$

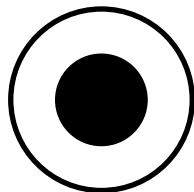
Решить систему
уравнений

Диаграмма деятельности: элементы

ІТМО



Начало алгоритма



Конец алгоритма

Диаграмма деятельности: элементы

ІТМО

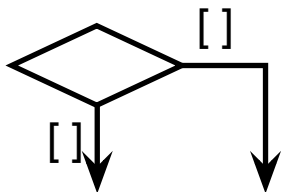
Переход



Переход срабатывает сразу
после завершения действия



Диаграмма деятельности: элементы



Ветвление

Ветвление – разделение на альтернативные ветви.

Диаграмма деятельности

Пример

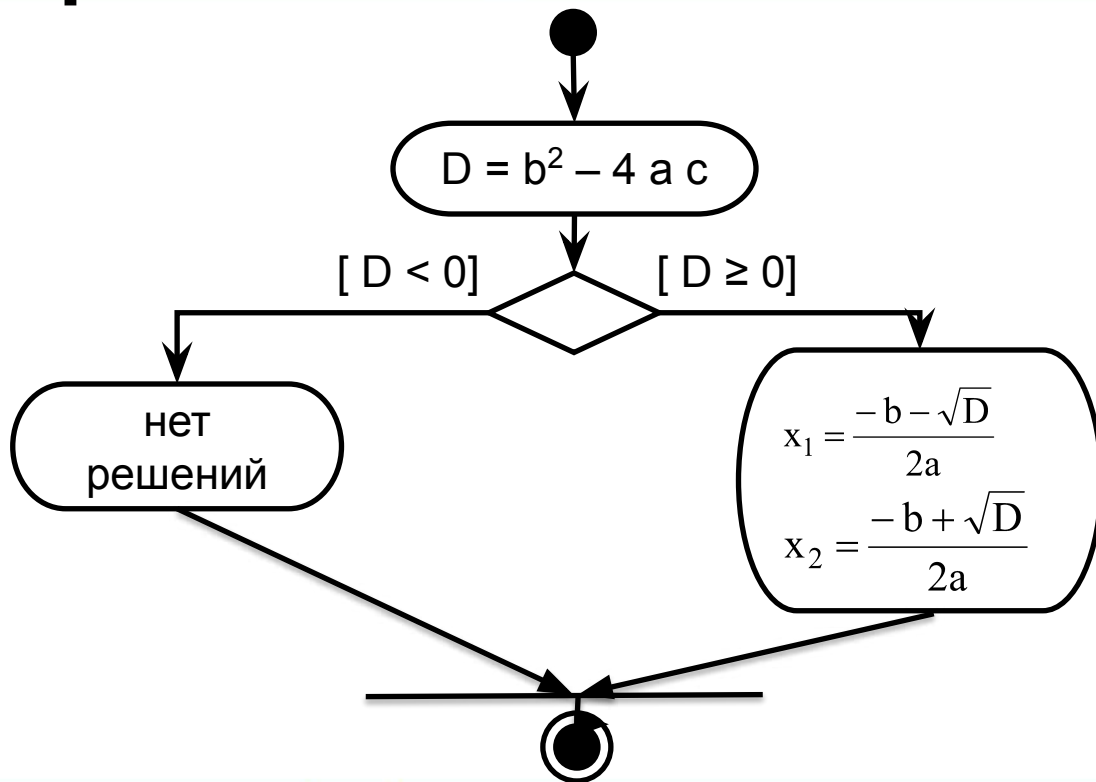
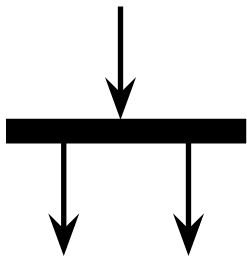
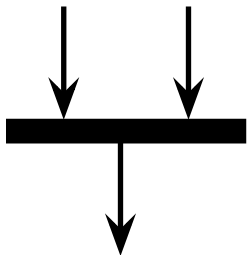


Диаграмма деятельности: элементы



Разделение

Разделение –
распараллеливание действий



Согласование

Согласование – переход к
следующему действию после
окончания всех согласуемых
действий



Диаграмма деятельности

Пример (с ошибками)

ІТМО

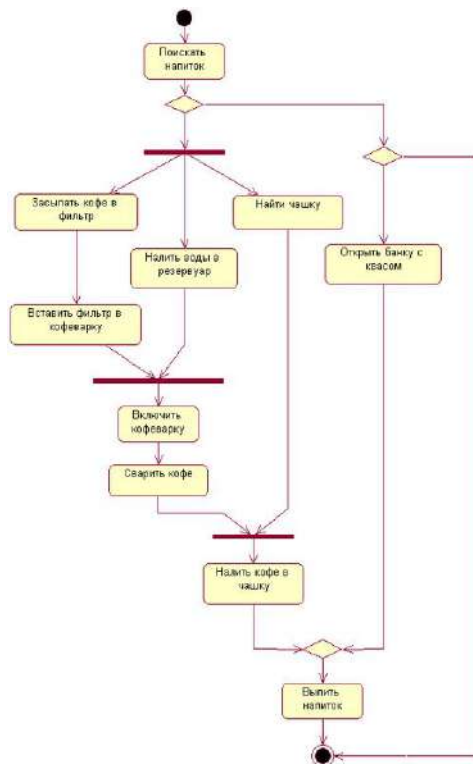


Диаграмма деятельности

Диаграмма Деятельности

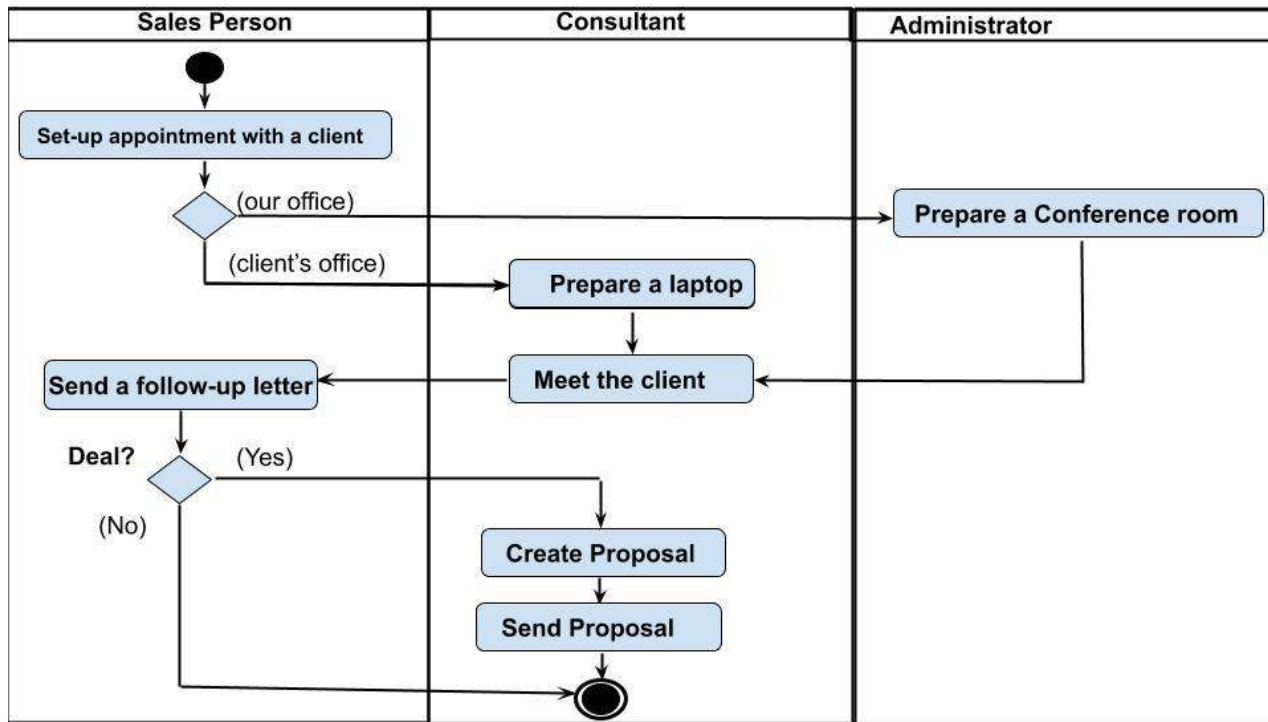
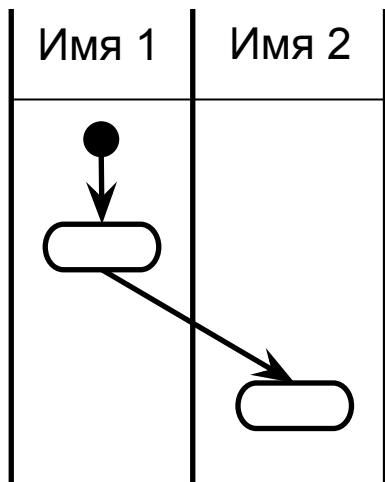


Диаграмма деятельности: элементы

ІТМО

Дорожка

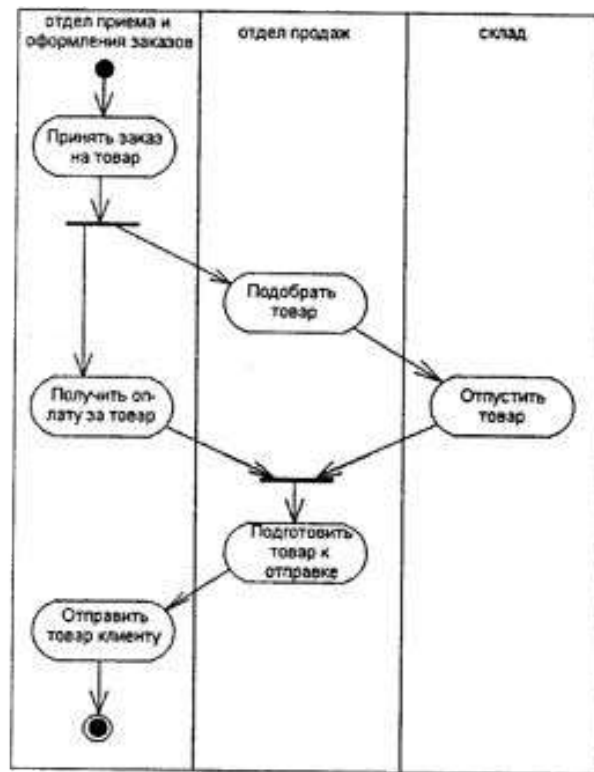


Дорожка обозначает
исполнителя действий

Диаграмма деятельности

Пример

ІІТМО





Последовательность построения диаграмм

Что внутри УС (прецедента)?

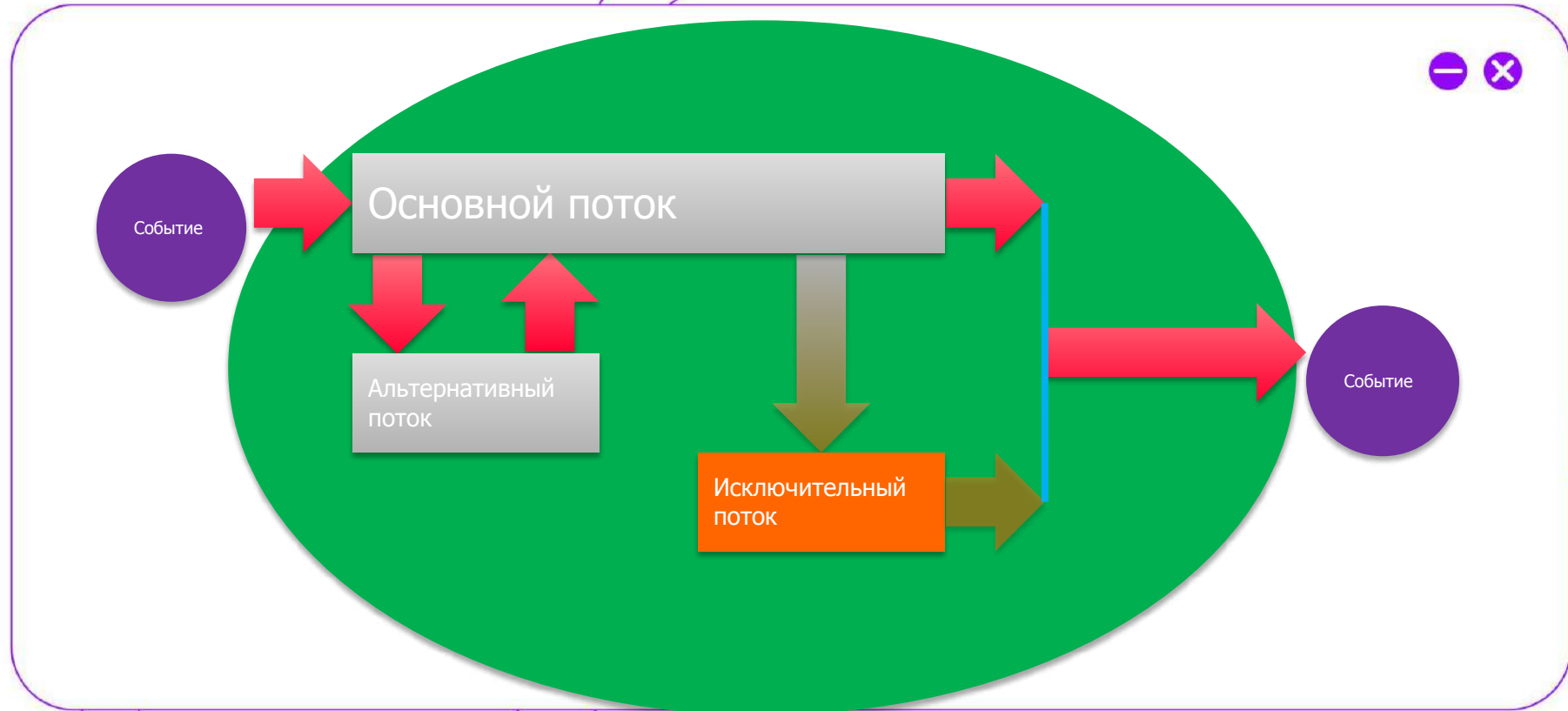


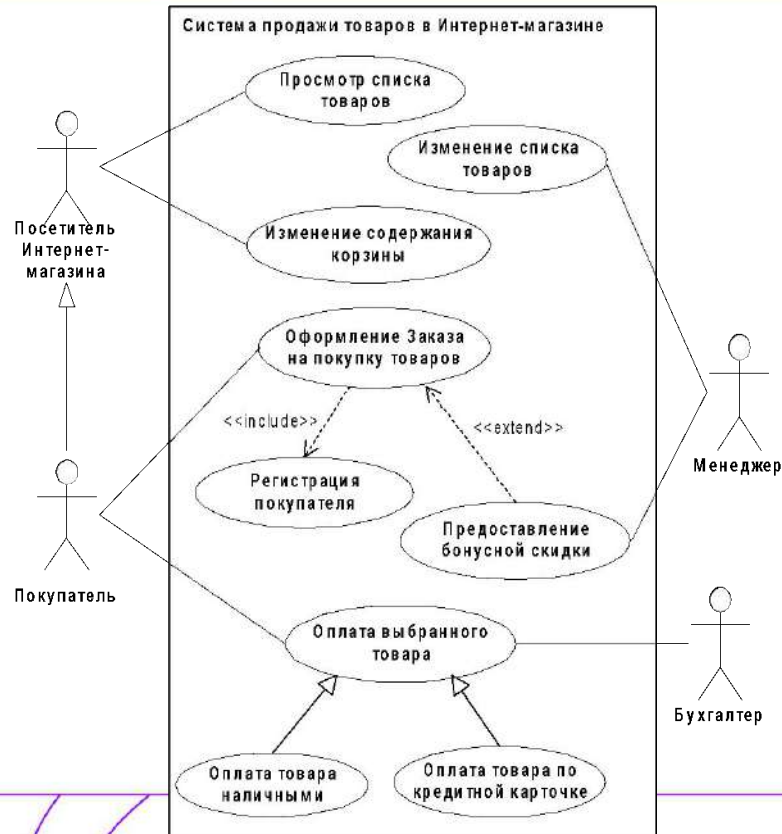
Диаграмма вариантов использования для модели банкомата

ІІТМО



Пример диаграммы прецедентов для Интернет-магазина

ИТМО



Типичные ошибки при разработке диаграмм прецедентов

- Превращение диаграммы вариантов использования в диаграмму деятельности за счет желания отразить все функциональные действия
- Инициатором действий является разрабатываемая система
- Задание слишком кратких имен вариантам использования
- Описание вариантов использования в терминологии, непонятной пользователям системы или заказчику
- Отсутствие описаний альтернативных последовательностей действий
- Тратится слишком много времени на решение вопросов о том, какие стереотипы и ассоциации использовать на диаграмме



Текстовые сценарии в UML

- Центральное место занимают функциональные требования, специфицирующие особенности реализации отдельных бизнес-процессов моделируемой системы. Они служат исходной информацией для построения диаграмм ВИ. Однако графических средств языка UML на практике оказывается недостаточно для спецификации функциональных требований.
- Одним из требований языка UML является самодостаточность диаграмм для представления информации о моделях проектируемых систем. Однако большинство разработчиков и экспертов утверждают, что изобразительных средств языка UML не хватает для того, чтобы учесть на диаграммах вариантов использования особенности функционального поведения сложной системы. С этой целью рекомендуется дополнять этот тип диаграмм текстовыми сценариями, которые уточняют или детализируют последовательность действий, совершаемых системой при выполнении ее вариантов использования.



Спецификация ВИ с помощью текстовых сценариев

- *Сценарий (scenario)* – специально написанный текст, который описывает поведение моделируемой системы в форме последовательности выполняемых действий актеров и самой системы.
- В контексте языка UML сценарий используется для дополнительной иллюстрации взаимодействия актеров и вариантов использования. Существуют различные способы представления или написания подобных сценариев.



Шаблон для написания сценария отдельного варианта использования

ИТМО

Главный раздел	Раздел "Типичный ход событий"	Раздел "Исключения"	Раздел "Примечания"
Имя варианта использования	Типичный ход событий, приводящий к успешному выполнению варианта использования	Исключение № 1	Примечания № 1
Актеры		Исключение № 2	Примечания № 2
Цель		...	...
Краткое описание			
Тип			
Ссылки на другие варианты использования		Исключение № N	Примечания № N



Шаблон для написания сценария отдельного варианта использования

ІТМО



Актер	Цель № 1	Успех	Исключение № 1	Примечания
			Исключение № 2	
			Исключение № 3	
	Цель № 2	Успех	Исключение № 1	
			Исключение № 2	
			Исключение № 3	

ГЛАВНЫЙ РАЗДЕЛ сценария выполнения варианта использования

ИТМО

"Снятие наличных по кредитной карточке"

Вариант использования	Снятие наличных по кредитной карточке
Актеры	Клиент, Банк
Цель	Получение требуемой суммы наличными
Краткое описание	Клиент запрашивает требуемую сумму. Банкомат обеспечивает доступ к счету клиента. Банкомат выдает клиенту наличные.
Тип	Базовый
Ссылки на другие варианты использования	Включает в себя ВИ: • Проверка ПИН-кода кредитной карточки • Идентифицировать кредитную карточку



Раздел ТИПИЧНЫЙ ход событий сценария выполнения варианта использования

"Снятие наличных по кредитной карточке"

ИТМО

Действия актеров	Отклик системы
1. Клиент вставляет кредитную карточку в устройство чтения банкомата Исключение №1: Кредитная карточка недействительна	2. Банкомат проверяет кредитную карточку 3. Банкомат предлагает ввести ПИН-код
4. Клиент вводит персональный PIN-код Исключение №2: Клиент вводит неверный ПИН-код	5. Банкомат проверяет ПИН-код 6. Банкомат отображает опции меню
7. Клиент выбирает снятие наличных со своего счета	8. Система делает запрос в Банк и выясняет текущее состояние счета клиента 9. Банкомат предлагает ввести требуемую сумму
10. Клиент вводит требуемую сумму 11. Банк проверяет введенную сумму Исключение №3: Требуемая сумма превышает сумму на счете клиента	12. Банкомат изменяет состояние счета клиента, выдает наличные и чек
13. Клиент получает наличные и чек	14. Банкомат предлагает клиенту забрать кредитную карточку
15. Клиент получает свою кредитную карточку	16. Банкомат отображает сообщение о готовности к работе



Раздел ИСКЛЮЧЕНИЯ сценария выполнения варианта использования



"Снятие наличных по кредитной карточке"

Исключение №1. Кредитная карточка недействительна или неверно вставлена

Действия актера	Отклик системы
	3. Банкомат отображает информацию о неверно вставленной кредитной карточке
	14. Банкомат возвращает клиенту его кредитную карточку
15. Клиент получает свою кредитную карточку	

Исключение №2. Клиент вводит неверный ПИН-код

	6. Банкомат отображает информацию о неверном <u>ПИН-коде</u>
4. Клиент вводит новый ПИН-код	

Исключение №3. Требуемая сумма превышает сумму на счете клиента

	12. Банкомат отображает информацию о превышении кредита
10. Клиент вводит новую требуемую сумму	



Последовательность разработки вариантов использования

- Определить главных (первичных) актеров и определить их цели по отношению к системе
- Специфицировать все базовые (основные) варианты использования (они соответствуют высокоуровневым требованиям, предъявляемым к системе)
- Выделить цели базовых ВИ, интересы актеров в контексте этих ВИ, предусловия и постусловия ВИ
- Написать успешный сценарий выполнения базовых ВИ
- Определить исключения (неуспех) в сценариях ВИ и написать сценарии для всех исключений (может быть изображено на отдельном UC)
- Выделить ВИ исключений и изобразить их со стереотипом «extend» (может быть изображено на отдельном UC)
- Выделить общие фрагменты функциональности ВИ (для нескольких UC) и изобразить их отдельными ВИ со стереотипом «include»



Показатели качества для модели прецедентов

- Все ли функциональные требования описываются вариантами использования?
- Не содержит ли модель вариантов использования ненужное поведение, которое отсутствует в требованиях?
- Действительно ли в модели необходимы все выявленные связи включения, расширения и обобщения?
- Правильно ли произведено деление модели на пакеты вариантов использования?
- Стала ли модель в результате деления на пакеты проще и удобнее для восприятия и сопровождения?
- Можно ли на основе модели вариантов использования составить четкое представление о функционировании системы в контексте ее пользователей?



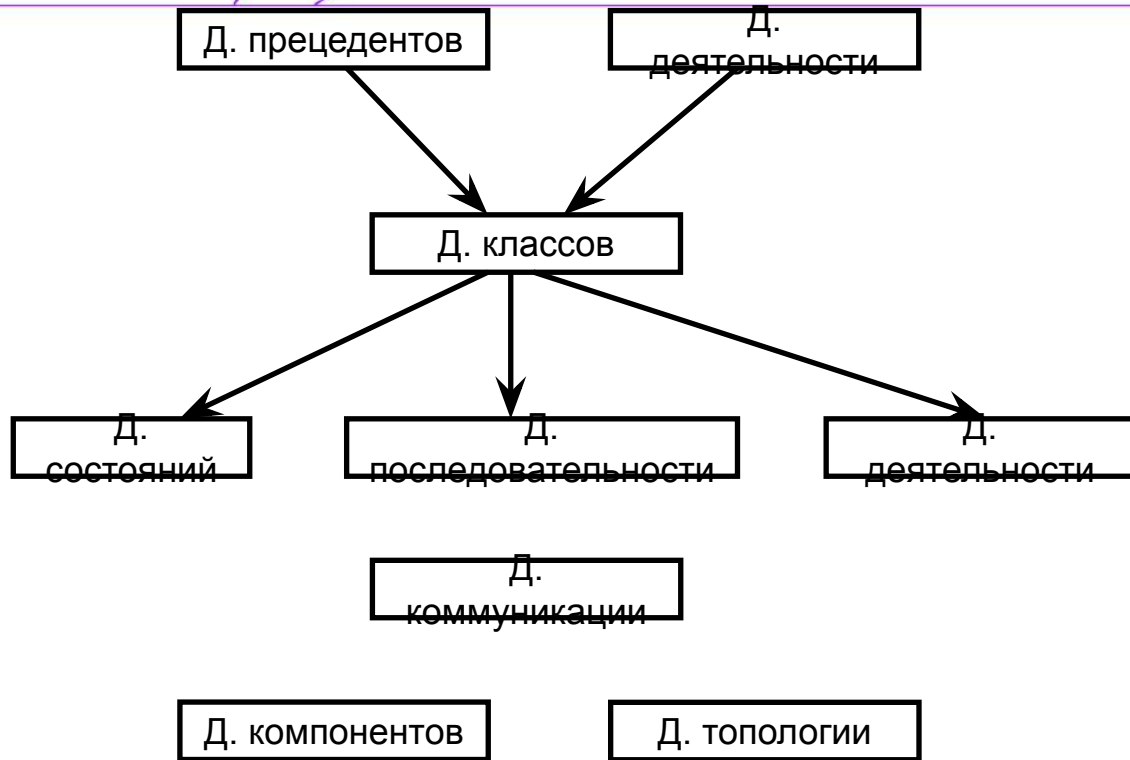
Последовательность построения диаграмм: способы

ІІТМО

- от функций ИС
- от физической реализации

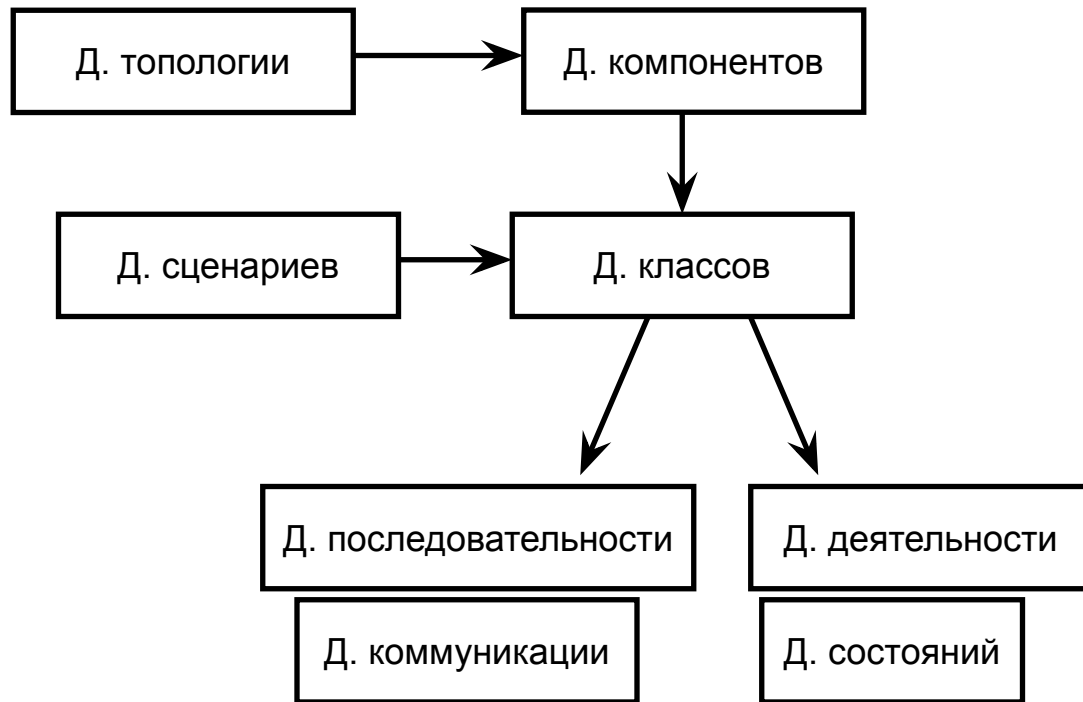
Последовательность построения диаграмм

ИТМО



Последовательность построения диаграмм

ІТМО



CASE – системы для построения диаграмм



CASE - системы

CASE (Computer Aided Software Engineering) – программные средства, поддерживающие процессы создания и сопровождения ИС, включая:

- анализ и формулировку требований,
- проектирование прикладного ПО (приложений) и баз данных,
- генерацию кода,
- тестирование,
- документирование,
- обеспечение качества,
- конфигурационное управление,
- управление проектом,
- другие процессы.



- CASE-технология представляет собой:
 - методологию проектирования ИС,
 - набор инструментальных средств, позволяющих в наглядной форме
 - моделировать предметную область,
 - анализировать эту модель на всех этапах разработки и сопровождения ИС,
 - разрабатывать приложения в соответствии с информационными потребностями пользователей.



CASE - системы

- CASE-средства вместе с системным ПО и техническими средствами образуют полную среду разработки ИС.
- Большинство существующих CASE-средств основано на методологиях структурного или объектно-ориентированного анализа и проектирования.



**Спасибо
за внимание!**

it'sMO *re than a*
UNIVERSITY

Ваши контакты