

# РОССИЙСКИЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ имени Г. В. ПЛЕХАНОВА



ДИСЦИПЛИНА Б1.В.03: Объектно-  
ориентированные технологии в программной  
инженерии

Специальность 38.03.05 Бизнес-информатика

➔ Тема : Unified Modeling Language (UML) -  
унифицированный язык моделирования

➔ АВТОР: к.т.н., доцент Горбенко А.О.



# UML ([англ.](#) *Unified Modeling Language* — унифицированный язык моделирования)

- **Язык UML** представляет собой общецелевой язык визуального моделирования, который разработан для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес-процессов и других систем. Язык UML может быть эффективно использован для построения концептуальных, логических и графических моделей сложных систем самого различного целевого назначения. Язык UML предлагает набор инструментальных средств, позволяющих проводить всесторонний анализ сложных ИС как с технической точки зрения, так и с точки зрения потребностей бизнеса.

UML упрощает процесс проектирования ИС, снижает его стоимость и повышает эффективность.

# Краткая история

- Решающую роль в создании языка UML сыграли Гарди Буч, Джеймс Рамбо и Айвар Джекобсон и созданные ими следующие методы моделирования различных сторон сложных систем:
- Метод Буча , ориентированный, в первую очередь, на моделирование программного обеспечения сложных систем.
- Метод Рамбо , ориентированный на анализ процессов обработки данных в информационных системах.
- Метод Джекобсона, ориентированный на анализ требований к бизнес-приложениям.

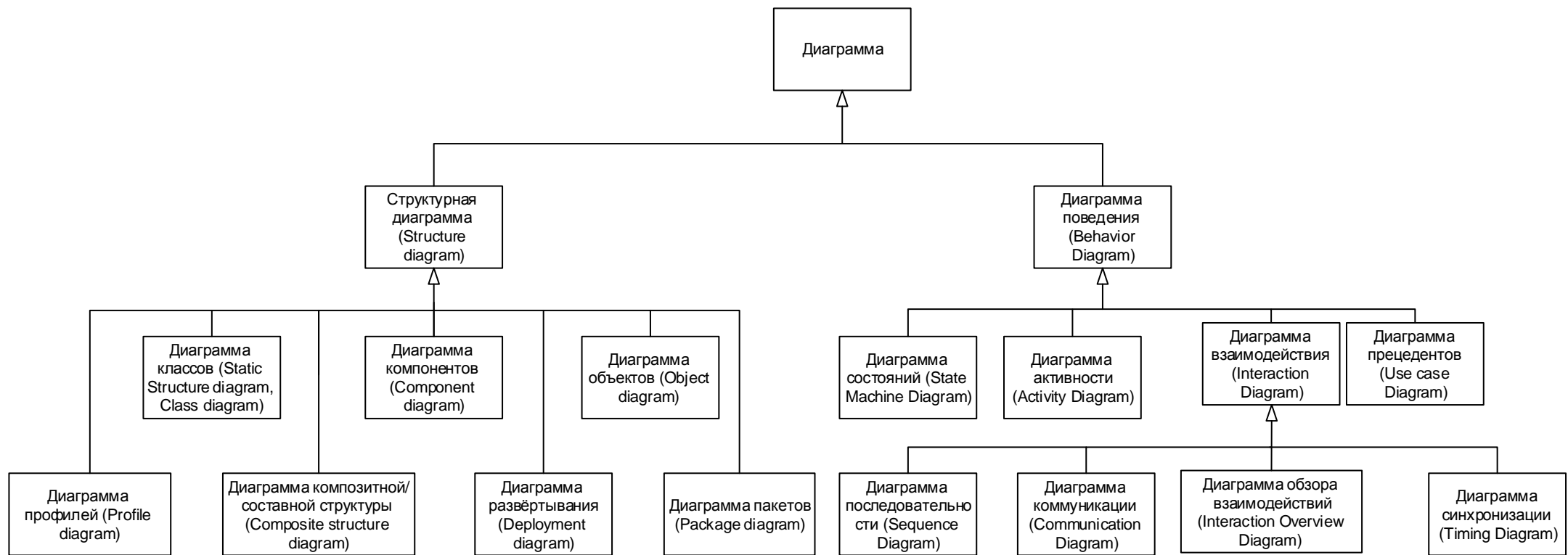
# Цели разработки

- **Главными в разработке UML были следующие цели:**
  - – предоставить пользователям готовый к использованию выразительный язык визуального моделирования, позволяющий разрабатывать осмысленные модели и обмениваться ими;
  - – предусмотреть механизмы расширяемости и специализации для расширения базовых концепций;
  - – обеспечить независимость от конкретных языков программирования и процессов разработки;
  - – обеспечить формальную основу для понимания этого языка моделирования (язык должен быть одновременно точным и доступным для понимания, без лишнего формализма);
  - – стимулировать рост рынка объектно-ориентированных инструментальных средств;
  - – интегрировать лучший практический опыт.

# Типы диаграмм UML

- Структурные
  - Классов (Static Structured diagram, Class diagram)
  - Объектов (Object diagram)
  - Компонентов (Component diagram)
  - Развёртывания (Deployment diagram)
- Поведения
  - Прецедентов (Use case diagram, Вариантов использования)
  - Состояний (State diagram)
  - Активности (Activity diagram, Деятельности)
- Взаимодействия
  - Кооперации (Коллаборации, Коммуникации, Collaboration diagram) – подвид диаграммы композитной/составной структуры
  - Последовательности (Sequence diagram)

# Основные типы диаграмм нотации UML



# Основные понятия языка UML.

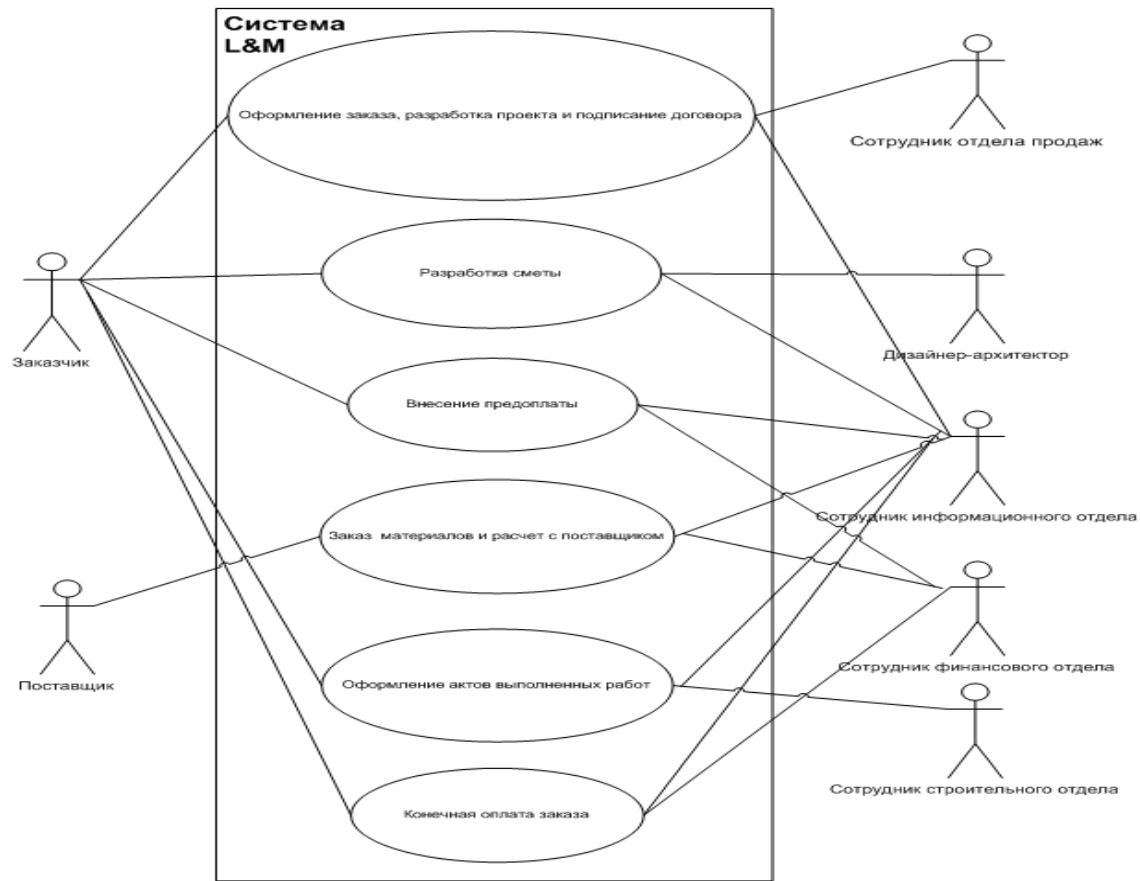
- **Объект**— сущность реального мира или концептуальная сущность, либо программная единица, состоящая из атрибутов (полей) и методов для их обработки. Характеризуется состоянием, поведением, и уникальностью. Состояние определяется атрибутами, а поведение – методами.
- **Класс** – описание множества объектов с общими атрибутами, определяющими состояние, и операциями, определяющими поведение. Класс изображается прямоугольником, разделенный на три секции (рис. 2). В первом приводится имя класса, во втором – список атрибутов, в третьем – список методов. **Ассоциация** – наиболее общий тип отношений первого рода, связанных с передачей сообщений ассоциированным классам путем делегирования

# Диаграмма прецедентов

- Диаграммы прецедентов применяются для моделирования вида системы с точки зрения внешнего наблюдателя. На диаграмме прецедентов графически показана совокупность прецедентов и субъектов, а также отношения между ними. **Прецедент** (use case) —это описание множества последовательностей действий (включая их варианты), которые выполняются системой для того, чтобы актер получил результат, имеющий для него определенное значение. Стандартным графическим обозначением прецедента на диаграммах является эллипс, внутри которого содержится краткое название прецедента или имя в форме глагола с пояснительными словами.



# Пример диаграммы прецедентов.

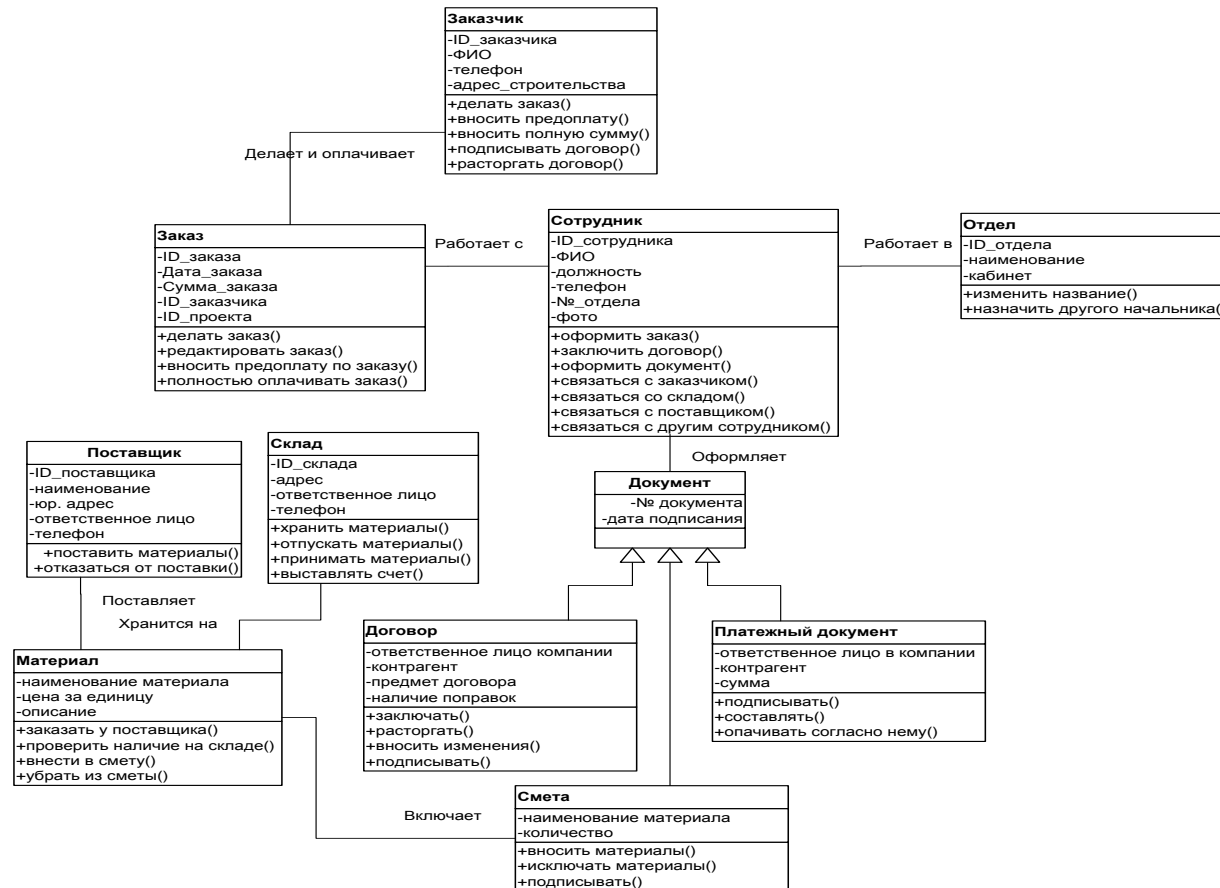


# Диаграмма классов

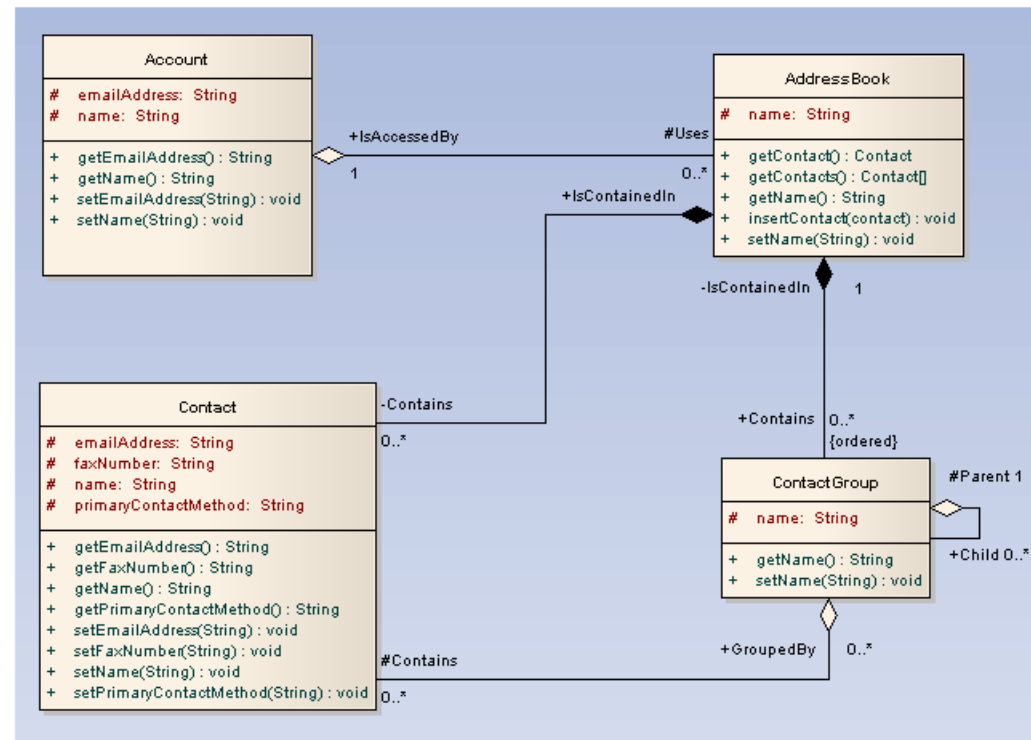
Заказчик
-ID_заказчика -ФИО -Телефон -Адрес строительства
+Делать заказ() +Вносить предоплату() +Вносить полную сумму() +Подписывать договор() +Расторгнуть договор()

- **Диаграмма классов** – основной способ описания структуры системы. Поскольку UML в первую очередь объектно—ориентированный язык, классы в нем являются основным, так называемым, «строительным материалом». Для построения диаграммы классов необходимо отобразить все выбранные классы. После этого необходимо составить список атрибутов и операций для выбранных классов.

# . Пример готовой диаграммы классов.

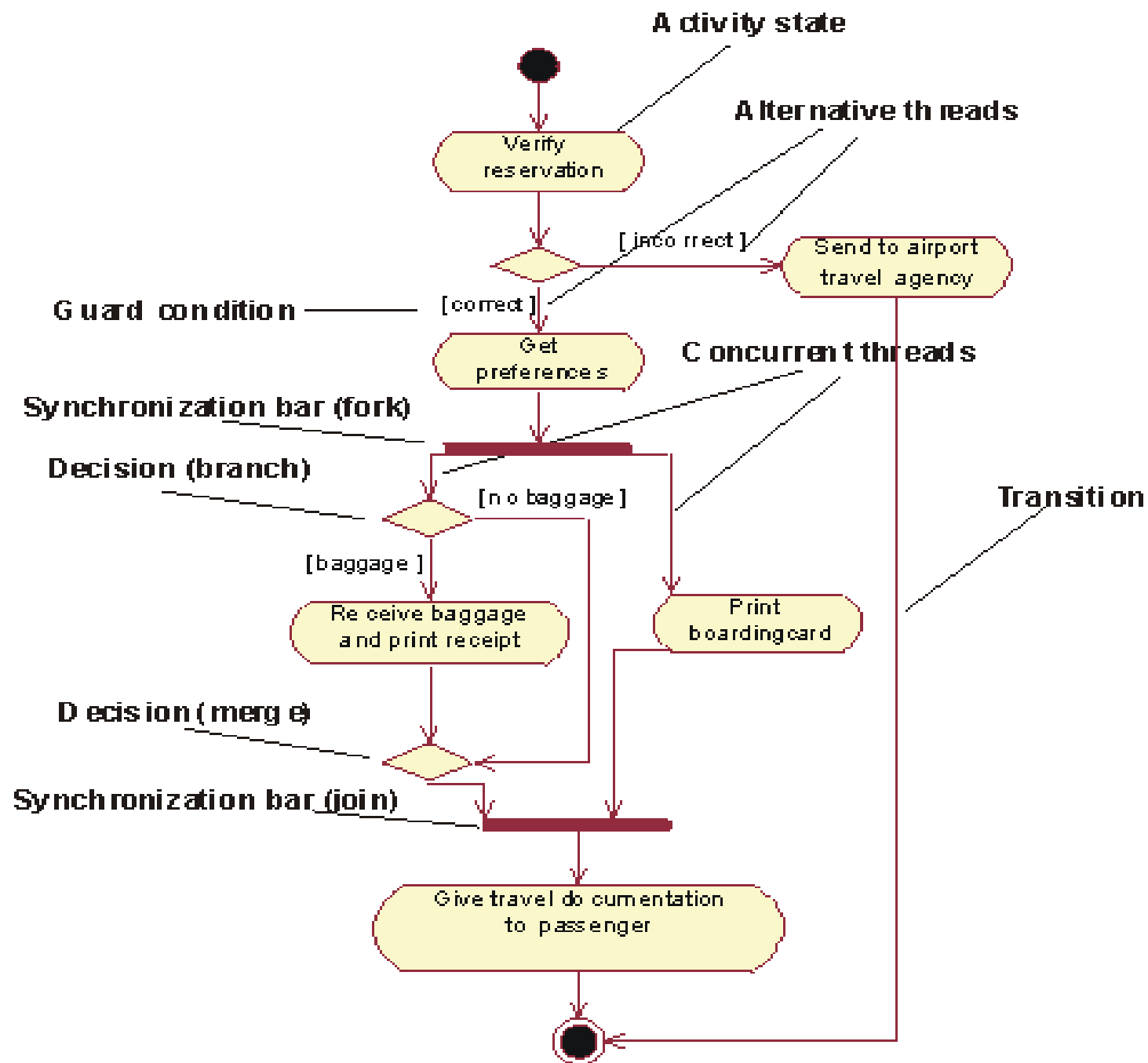


# Пример диаграммы классов



# Диаграмма деятельности может быть использована для моделирования

- бизнес-процессов;
- 2. бизнес-правил;
- 3. сценариев работы пользователя с системой
- 5. функций, процессов, алгоритмов системы;
- 4. правил системы;
- 5. сценариев тестирования



- Для разработки модели конкретного бизнес-процесса следует использовать диаграмму деятельности (**activity diagram**)

# Элементы диаграммы деятельности

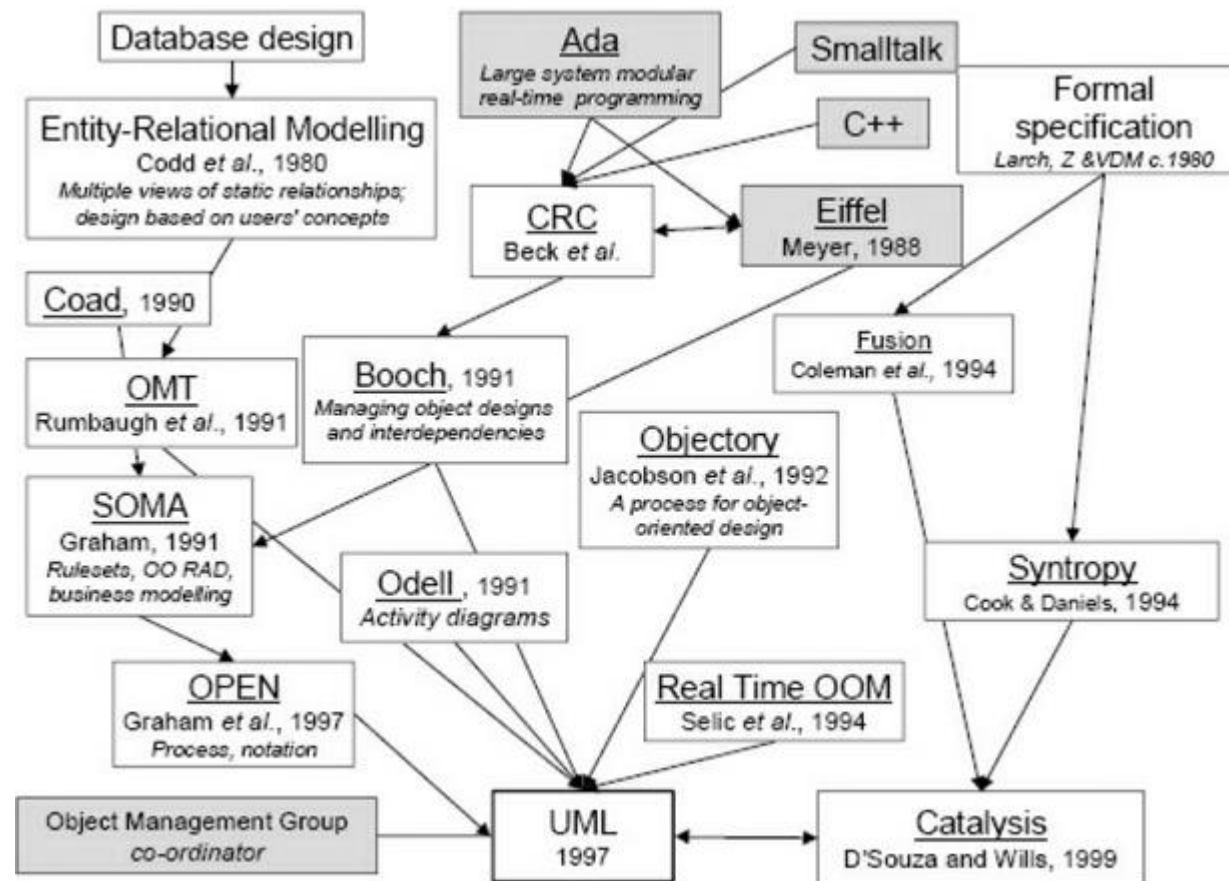
- начальное состояние (**start state**);
- - конечное состояние (**end state**);
- - деятельность (**activity**);
- - состояние (**state**);
- - переход (**state transition**);
- решение (**decision**);
- - горизонтальные синхронизаторы (**horizontal synchronization**);
- - вертикальные синхронизаторы (**vertical synchronization**);
- разделительные линии (**swimlane**);
- - объект (**object**);
- - поток объектов (**object flow**);
- - заметка



- Начальное состояние (**start state**) должно обозначаться черным маленьким кружком, с которым может быть связано название, например, «начало».
- Конечное состояние (**end state**) должно обозначаться большим черным кружком внутри круга, с которым может быть связано название, например, «конец»

● Начало

⦿ Конец



# Диаграмма деятельности

Диаграмма деятельности

**(activity diagram)** должна иметь только одно начальное состояние. Конечных же состояний может существовать множество. Новые начальные состояния могут быть только на диаграммах, которые декомпозируют отдельные виды деятельности

# Диаграмма объектов (Objectdiagram)

- – демонстрирует полный или частичный снимок моделируемой системы в заданный момент времени. На диаграмме объектов отображаются экземпляры классов (объекты) системы с указанием текущих значений их атрибутов и связей между объектами.

# Объект (object):

- **Объект (object):**
- конкретная материализация абстракции;
- сущность с хорошо определенными границами, в которой инкапсулированы состояние и поведение;
- экземпляр класса. Объект уникально идентифицируется значениями атрибутов, определяющими его состояние в данный момент времени.

# Применение диаграммы объектов

- **Применение диаграммы объектов**
- С помощью диаграмм объектов, как и с помощью диаграмм классов, моделируют статический вид системы с точки зрения проектирования или процессов, но принимая во внимание реальные экземпляры или прототипы. Этот вид описывает главным образом функциональные требования к системе, то есть услуги, которые она должна предоставлять конечным пользователям.
- При моделировании статического вида системы с точки зрения проектирования или процессов объектные диаграммы применяют для того, чтобы моделировать структуру объектов.

# Диаграмма состояний

- **Диаграмма состояний** используются для описания поведения, реализуемого в рамках варианта использования, или поведения экземпляров класса (объектов), компонента, узла или системы в целом.
- Диаграммы состояний чаще всего используются для описания поведения отдельных объектов, но также могут быть применены для спецификации функциональности других компонентов моделей, таких как варианты использования, актеры, подсистемы, операции и методы.

# Состояние

- **Состояние** – абстрактный класс для моделирования отдельной ситуации. Состояние может быть задано в виде набора конкретных значений атрибутов класса или объекта. Изменение отдельных значений атрибутов будет отражать изменение состояния моделируемого класса или объекта.



# Диаграмма компонентов (Component diagram)

- **Диаграмма компонентов (Component diagram)** — статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами.
- **Компонент (component)** — реализует некоторый набор интерфейсов и служит для общего обозначения элементов физического представления модели.
- Для графического представления компонента используется специальный символ — прямоугольник, составленный из левого и правого более мелких прямоугольников. Внутри большого прямоугольника записывается имя компонента, при необходимости, некоторая дополнительная информация.

# Диаграмма развертывания

- **Диаграмма развертывания - (Deploymentdiagram, диаграмма размещения)** — модель физической архитектуры системы, которая показывает аппаратную конфигурацию ИС.
- **Узел (node)** представляет собой некоторый физически существующий элемент системы, обладающий некоторым вычислительным ресурсом.
- Графически на диаграмме развертывания узел изображается в форме трехмерного куба. Узел имеет собственное имя, которое указывается внутри этого графического символа.



## Структура языка UML

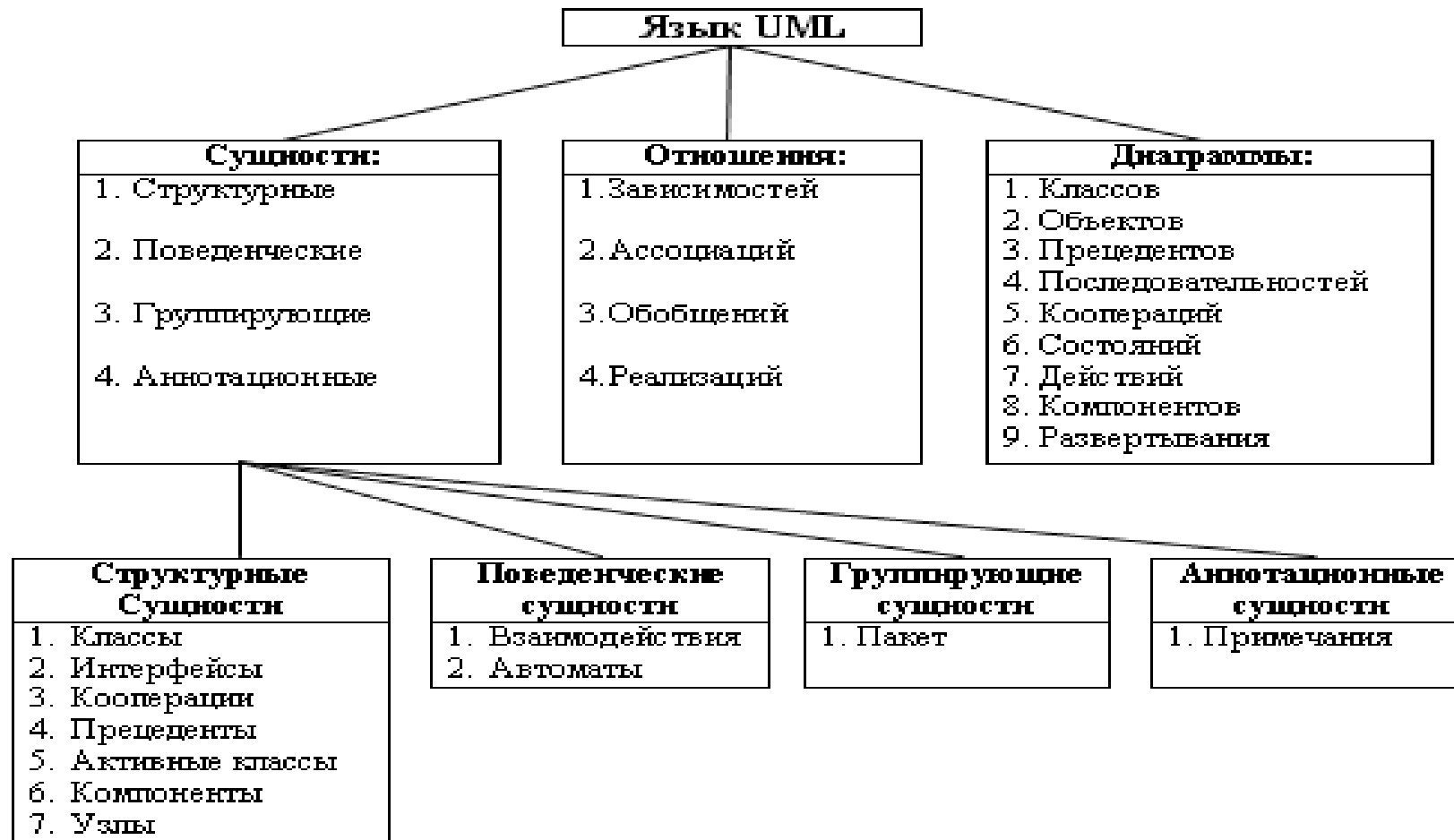
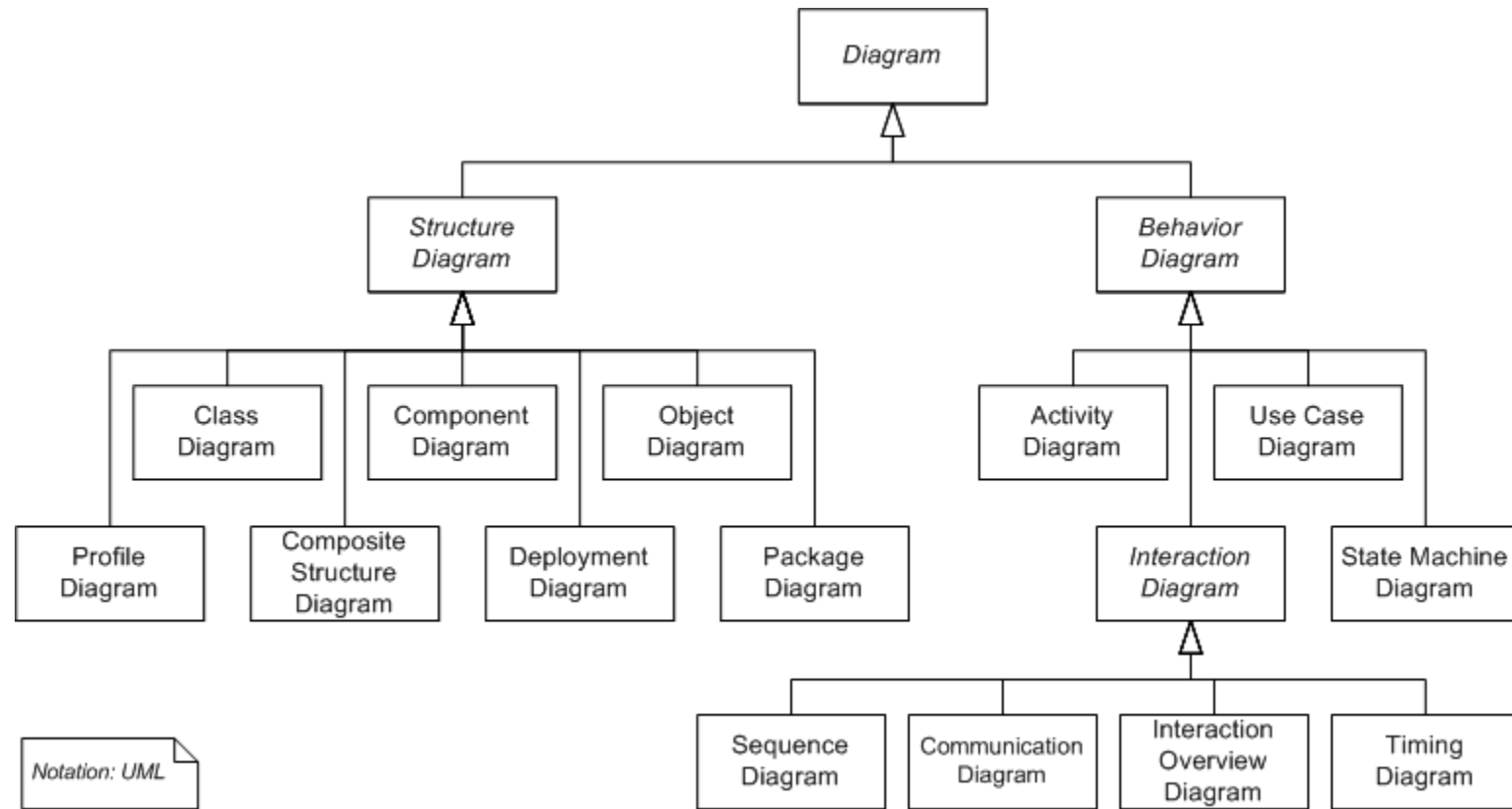


Рис. 3.2



# Рекомендуемая последовательность разработки диаграмм:

- диаграмма прецедентов
- диаграмма классов
- диаграмма объектов
- диаграмма последовательностей
- диаграмма коопераций
- диаграмма состояний
- диаграмма деятельности
- диаграмма компонентов
- диаграмма развертывания.

# Выводы по теме:

- UML – унифицированный язык визуального моделирования, который является языком широкого профиля; это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем. Использование UML не ограничивается моделированием программного обеспечения. Его также используют для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.



# Рекомендуемая литературы по теме

- *Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. Язык UML. Руководство пользователя = The Unified Modeling Language user guide* — 2-е изд. — М., СПб.: [ДМК Пресс](#), [Питер](#), 2004. — 432 с. — [ISBN 5-94074-260-2](#).
- Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с. [ISBN 5-469-00599-2](#)
- **Рекомендуемая литература**
- Чен П.П. Модель "сущность-связь" - шаг к единому представлению данных. СУБД, N3, 1995 г.
- Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. М., Финансы и статистика, 1998.
- Вендров А.М. Проектирование программного обеспечения экономических информационных систем. М., Финансы и статистика, 2000.
- Фаулер М., Скотт К. UML в кратком изложении. Применение стандартного языка объектного моделирования. М., Мир, 1999.
- Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд. М., Издательство Бином, СПб., Невский диалект, 1999.



# Учебно-методическое обеспечение для самостоятельной работы студентов

- Литература О-1, Д-3,4,5.
- Вопросы для самопроверки:
- Методология объектно-ориентированного анализа и проектирования информационных систем. Сущность объектно-ориентированного подхода к анализу и проектированию ИС. Варианты использования. Описание варианта использования. Виды сценариев вариантов использования. Диаграммы вариантов использования, их элементы и связи между элементами. Примеры.
- Диаграммы взаимодействия UML, их элементы и связи между элементами. Виды диаграмм взаимодействия и область их применения. Примеры.
- Диаграммы классов UML, их элементы и связи между элементами. Область применения. Примеры.
- Диаграммы состояний, их элементы и связи между элементами. Область применения. Примеры.
- Диаграммы деятельности, их элементы и связи между элементами. Область применения. Примеры.
- Диаграммы компонентов и диаграммы размещения, их элементы и связи между элементами. Область применения. Примеры.



Спасибо за внимание  
Ваши вопросы