

Какие методологии проектирования ИВС будем рассматривать?

Функциональная		Потоков данных		Процессная	
Обозначение и язык	SADT	Обозначение и язык	DFD	Обозначение и язык	--
Стандарт	IDEF0	Стандарт	--	Стандарт	IDEF3
CASE - средства	BPWin (AllFusion), IDEF Doctor, MS Visio	CASE - средства	BPWin (AllFusion), IDEF Doctor, MS Visio	CASE - средства	BPWin (AllFusion), IDEF Doctor, MS Visio

Сущность - связь	
Модель структуры БД	
Обозначение и язык	ER – диаграммы
Стандарт	IDEF1X
CASE - средства	ERWin (AllFusion)

Объектно-ориентированная	
Обозначение и язык	UML
Стандарт	UML 2.0
CASE - средства	Rational Rose, Star UML, MS Visio

Процессная	
Обозначение и язык	eEPC/PCD, VAD
Стандарт	ARIS
CASE - средства	ARIS Toolset, MS Visio

Объектно-ориентированное проектирование ИВС на языке UML

<p>Язык UML (Unified Modeling Language) – язык визуального моделирования ИВС, который создан для конструирования и документирования компонентов программного обеспечения и бизнес-процессов</p>	
<p>Модель ИВС фиксируется на языке UML в виде 9-ти графических диаграмм:</p>	
1.	Диаграмма вариантов использования (use case)
2.	Диаграмма классов (class)
3.	Диаграмма состояний (state chart)
4.	Диаграмма деятельности (activity)
5.	Диаграмма последовательности (sequence)
6.	Диаграмма кооперации (collaboration)
7.	Диаграмма реализации (implementation)
8.	Диаграмма компонентов (component)
9.	Диаграмма развертывания (deployment)

UML: Диаграмма вариантов использования (use case)

Диаграмма показывает, какие пользователи и как могут использовать систему

Актер (Actor)	Сущность, взаимодействующая с ИВС извне (пользователь, техническое устройство, и т.п.). Разные Актеры взаимодействуют по разным сценариям
Вариант использования	Физически понятное законченное действие
Отношение ассоциации	Используется для Актеров
Отношение включения	Указывает, что вариант использования должен привлечь другой. Стрелка направлена от базового варианта к включаемому
Отношение расширения	Указывает, что вариант использования может привлекать другой. Стрелка направлена от включаемого варианта к базовому
Отношение обобщения	Указывает, что вариант использования является «более узким» к

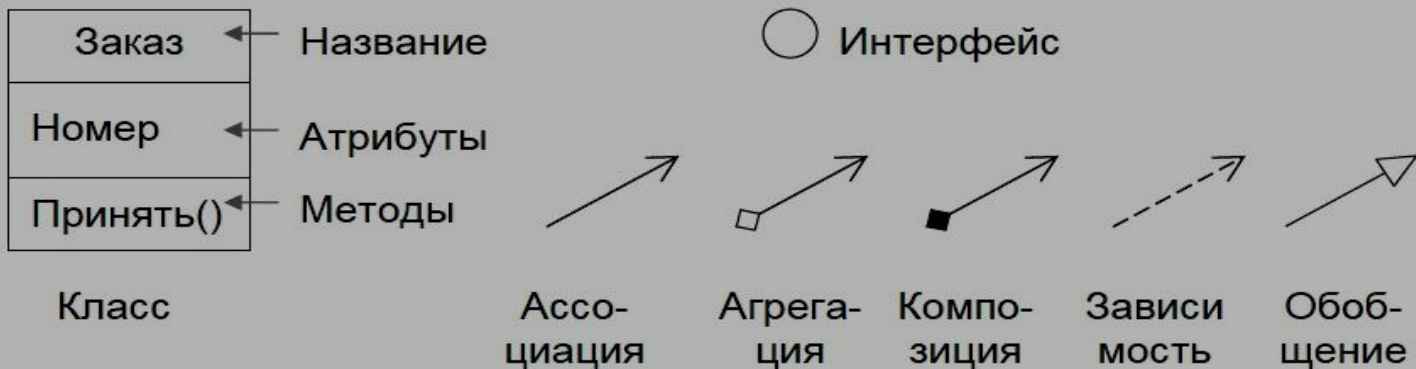


UML: Диаграмма классов (class)

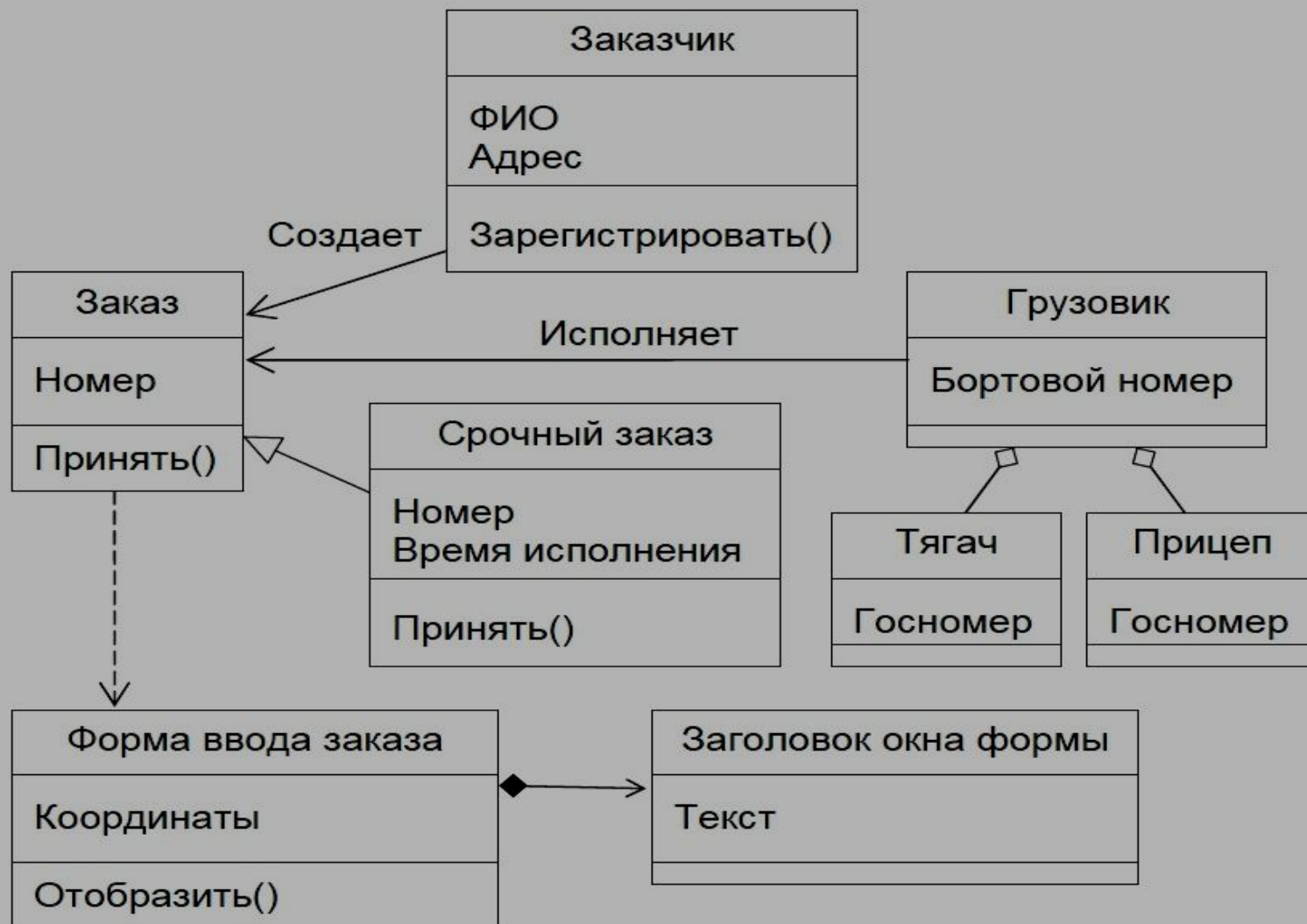
Графические примитивы, термины

Диаграмма классов – это граф, вершины которого - классы, соединенные различными отношениями

Класс	Полный аналог класса в C++. Имеет имя, атрибуты (в C++ - свойства), методы
Отношение ассоциации	Означает, что между классами есть некоторое отношение
Отношение агрегации	Класс включает в себя другие в качестве составных частей
Отношение композиции	Частный случай агрегации, составные части не могут существовать без целого
Отношение зависимости	Указывает, что внесение изменений в базовый класс потребует внесение изменений в зависимый
Отношение обобщения	



UML: Диаграмма классов (class) Пример



UML: Диаграмма состояний (state chart)

Диаграмма показывает возможные последовательности состояний одного экземпляра конкретного класса ИВС

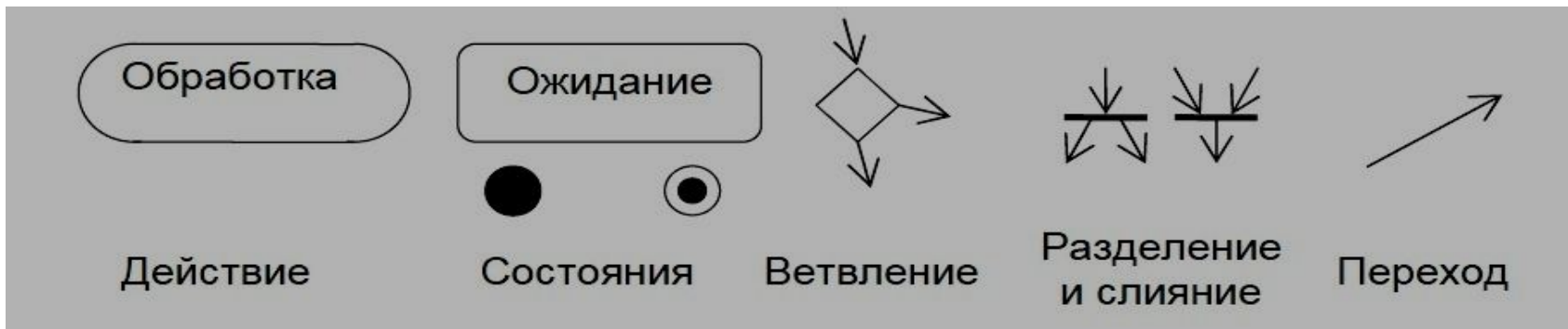
Состояние	Задается в виде конкретного значения атрибутов класса, определяющее его поведение. Например, флажок «Заказ принят». Имеет имя состояния и метки действия
Метки действия	Список внутренних действий в данном состоянии (entry, exit, do, include)
Дуги графа состояний	Обозначают переход из одного состояния в другое.
Срабатывание перехода из состояния в состояние	Происходит при наступлении некоторого события и от выполнения определенного условия – сторожевого условия



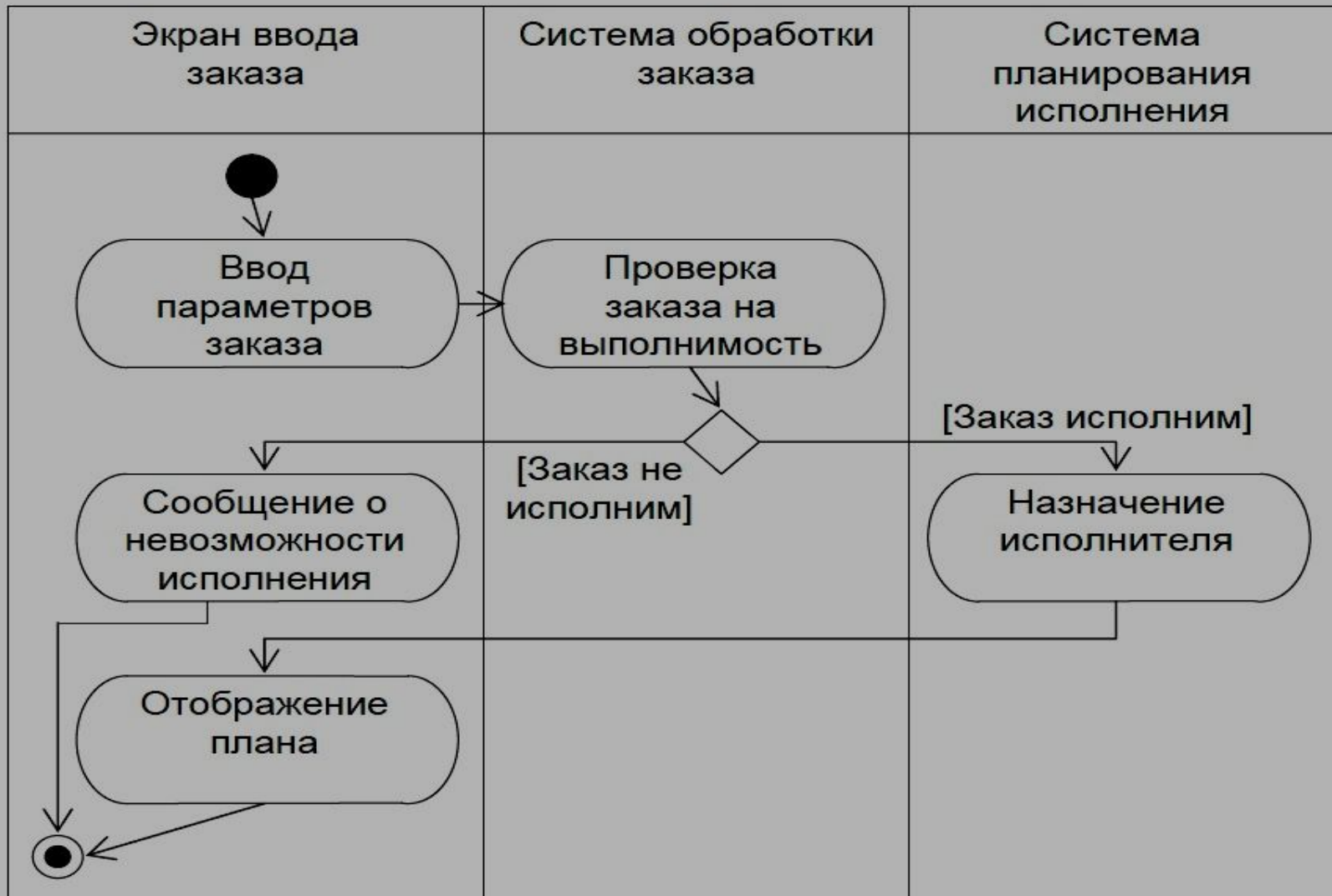
UML: Диаграмма деятельности (activity) Графические примитивы, термины

Диаграмма деятельности - это описание алгоритмов методов классов в виде, напоминающем блок-схему алгоритмов

Обработка	Функционально законченный блок подсистемы или метод класса
Состояние	Аналогично понятию «Состояние» из диаграммы состояний
Ветвление	Аналогично условному переходу в блок-схемах
Разделение и слияние	Используются для изображения параллельных вычислений
Дорожки	Используются для группировки блоков обработки по видам деятельности ИВС: «Экран ввода заказа», «Система обработки», ... Блоки группируются на диаграмме в вертикальные столбцы с названием столбца по виду деятельности.



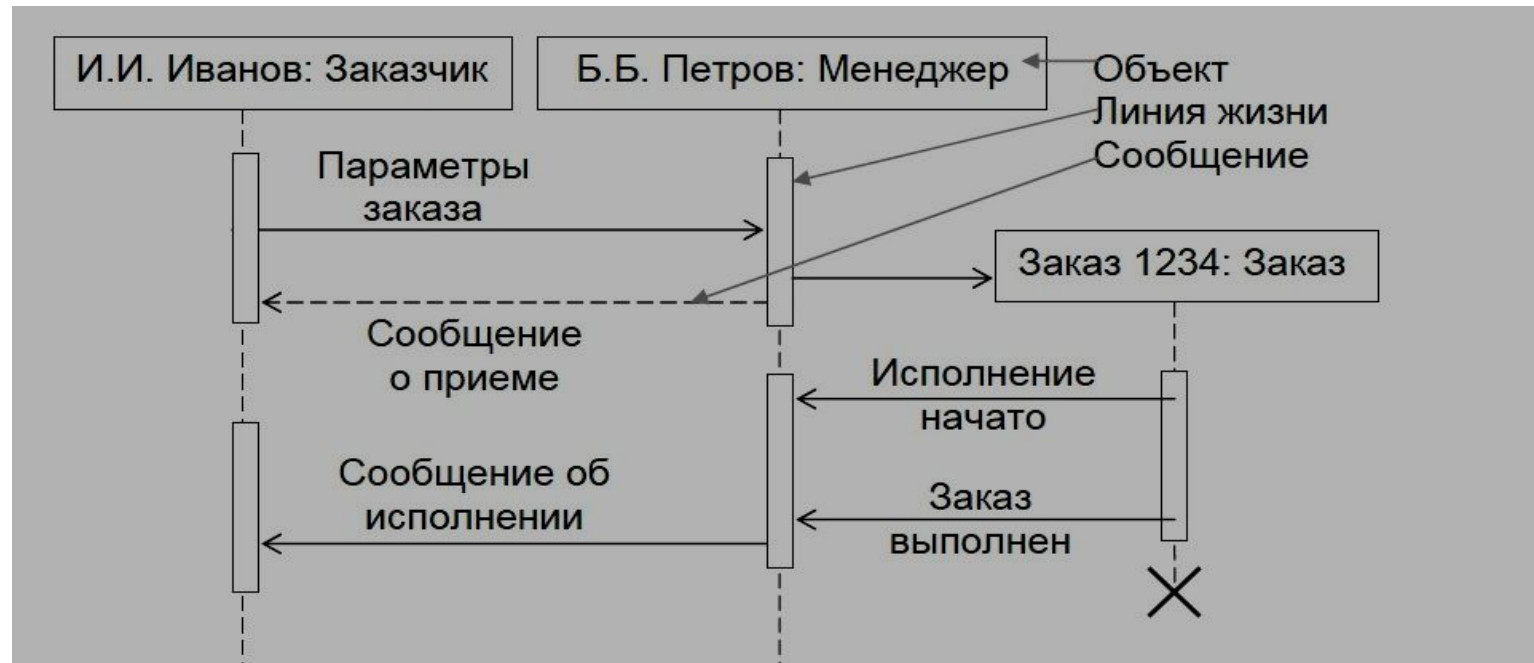
UML: Диаграмма деятельности (activity) Пример



UML: Диаграмма последовательности (sequence)

Диаграмма последовательности отображает взаимодействие экземпляров классов во времени через обмен сообщениями

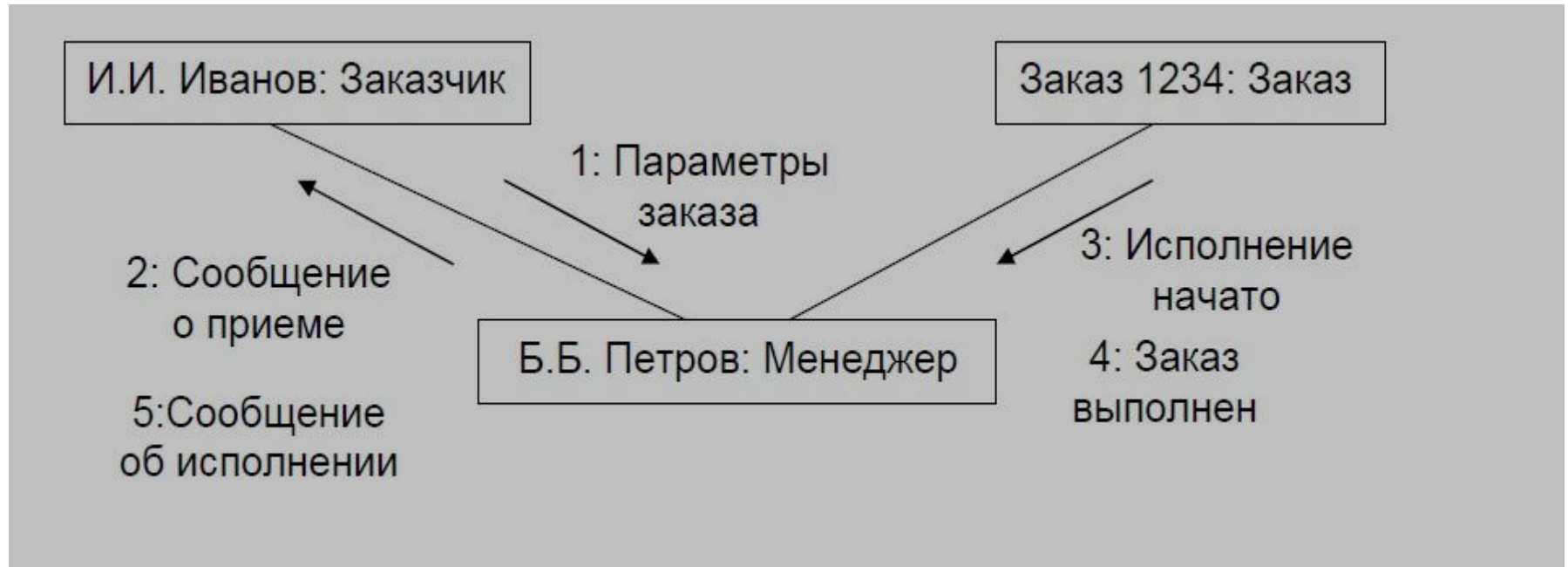
Объект	Конкретный экземпляр конкретного класса. Имя объекта состоит из Имени Экземпляра и Имени класса («Имя Экземпляра: Имя Класса»)
Сообщение	Законченный фрагмент информации, передаваемый одним объектом другому
Линия жизни	Вертикальная пунктирная линия. Верх – начало жизни, низ - окончание
Фокус управления	Отрезок времени на линии жизни, когда объект активен. Изображается на линии жизни, как вытянутый прямоугольник. Вершина – начало фокуса, низ – окончание фокуса



UML: Диаграмма кооперации (collaboration)

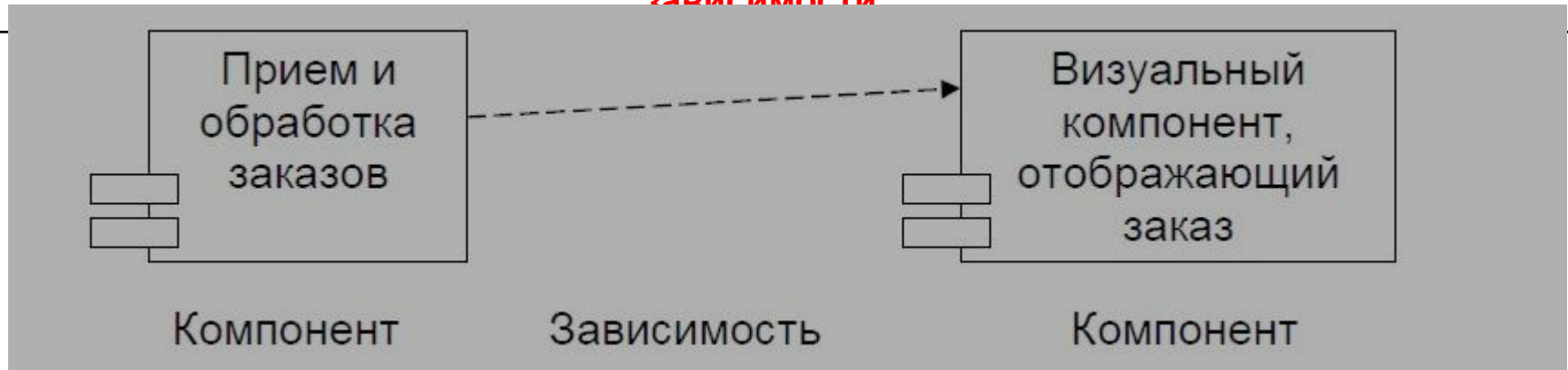
Диаграмма кооперации отображает все возможные отношения между объектами без привязки ко времени

Объект	Конкретный экземпляр конкретного класса. Имя объекта состоит из Имени Экземпляра и Имени класса («Имя Экземпляра: Имя Класса»)
Ассоциации	Связи между объектами. Изображаются сплошными линиями
Отношения	Информация, которая передается между объектами. Направление сообщения указывается стрелками. Текст сообщения отображается рядом со стрелкой и нумеруется



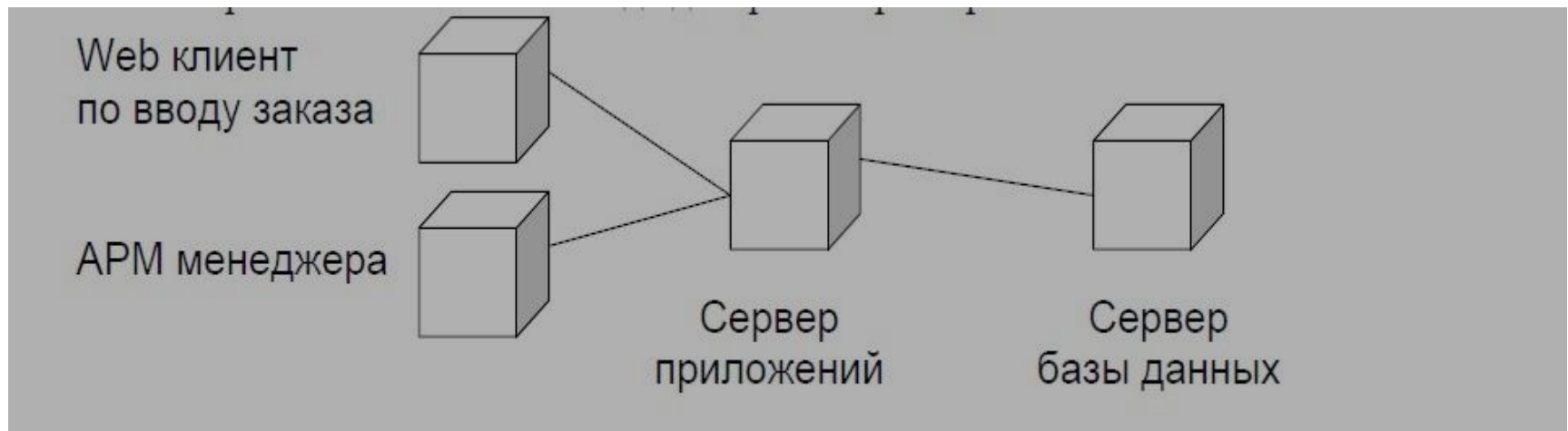
UML: Диаграмма компонентов (component)

Диаграмма компонентов отображает компоненты программного обеспечения и их зависимости



UML: Диаграмма развертывания (deployment)

Диаграмма развертывания отображает топологию (размещения) аппаратных средств ИВС и маршруты передачи информации между устройствами



ИТАК. Идеальная выпускная квалификационная работа – это:

1.	Изучить предметную область. Изучить существующую ИВС
2.	Разработать диаграмму вариантов использования существующей системы. Не забыть сказать, что такая диаграмма строится как AS-IS, т.е. для существующей, и ее (диаграмму) до сих пор никто не делал. Если нет существующей – опишите, как сейчас происходит бизнес-процесс, например, как сейчас в библиотеке бабки врукопашную выписывают тебе читательский билет (в ГЭКе будет смешно) .
3.	Разработать диаграмму классов существующей системы. Не забыть сказать, что такая диаграмма строится как AS-IS, т.е. для существующей, и ее (диаграмму) до сих пор никто не делал.
4.	Сформулировать в виде текста постановку задачи , как части существующей ИВС.
5.	Сформулировать цель . Например «провести реинжиниринг с целью убрать процедуру ручной регистрации полученных пятидесяти тысяч банкнот на кассе».
6.	Определить точку зрения : хозяин магазина/ бухгалтер для отчета для налоговой/ продавец на кассе.
7.	Говорить о выборе программных средств??? Только в случае мобильных приложений – там ответ очевиден – Java (только она работает и на Андроиде, и на iOS). Все остальное – получишь лишние глупые вопросы. Но, если спросят (а спросят)) – уверенно отвечать: существующая система создана на таком-то языке с такой-то БД, поэтому выбора не было. Либо, если вновь делаешь, сказать, что сегодня специалисты, работающие на C#/Java/Python/PHP... с БД MS SQL/... ценятся, поэтому я решил получить практические навыки именно в этой среде.

ИТАК. Идеальная выпускная квалификационная работа – это :

8.	Разработать диаграмму вариантов использования Вашей подсистемы.
9.	Разработать диаграмму классов Вашей подсистемы.
10.	<i>Желательно</i> разработать диаграмму состояний Вашей подсистемы.
11.	Разработать блок-схемы программ, <i>желательно</i> в формате диаграммы деятельности Вашей подсистемы.
12.	<i>Обязательно</i> разработать диаграмму компонентов и диаграмму развертывания Вашей подсистемы (это не сложно, но дает эффект на докладе).
13.	Модель структуры базы данных в стандарте IDEF1X (ER-модель): логическая и физическая.
14.	Листинг программы.