

Проектирование архитектур программного обеспечения

лекция 1

Зозуля А.В.

Содержание

- Введение
- Информационные системы
- Архитектура ПО
- Архитектурные слои
- Моделирование

Место курса

- 1 курс: Программирование
- 2 курс: Java-технологии
- 5 курс: Технологии разработки ПО
- 5 курс: **Проектирование архитектур ПО**
- 6 курс: СПО РВС
- 6 курс: Языки программирования

Структура курса

- I. Объектно-ориентированный анализ**
- II. Шаблоны проектирования**
- III. Архитектурные шаблоны**
- IV. Архитектура веб-приложений**
- V. Шаблоны интеграции информационных систем**

Типы ПО

- Системное
- Прикладное
 - Офисное
 - САПР
 - Мультимедиа
 - Утилитарное
 - **Информационные системы**
 - ...
- Инструментальное

Информационные системы

- Конечный пользователь
- Роли
- Пользовательские интерфейсы
- Развитая бизнес-логика
- Интеграция с другими ИС
- Меняющиеся требования

Архитектура

Архитектура [IEEE 1471] - базовая организация системы, воплощенная в ее

- компонентах, объединенных для выполнения определенных функций,
- их отношениях между собой и с окружением, определяющим ход и обстоятельства влияния на систему, а также
- принципы, определяющие проектирование и развитие системы.
- Является реализацией нефункциональных требований к системе, в то время как проектирование ПО — реализация функциональных требований.

Архитектура

- **Участник (заинтересованная сторона)** – любой человек, правомочно заинтересованный в проекте. *Например, заказчики, разработчики, обслуживающий персонал, руководство.*
- **Точка зрения (разер)** – шаблон или обобщение представления, задает правила для представлений. Точки зрения подбираются в соответствии с соображениями участников. *Например, точка зрения развертывания, точка зрения защиты безопасности, точка зрения надежности.*
- **Представление** – это отображение целой системы через призму определенных взглядов. Представление можно рассматривать как экземпляр точки зрения. Представление состоит из одной или нескольких моделей.
- **Модель** – это абстракция или упрощенное отображение системы или ее части: UML (универсальный язык моделирования), диаграммы сущностей и связей (ER-диаграммы), нестандартизированные системы обозначений (блок-схемы, стрелочные диаграммы).

Архитектура

Рамочная модель разработки архитектуры по IEEE 1471



Зачем проектировать архитектуру?

- Программы читаю чаще, чем пишу
- Разработка не может быть окончательной
- Разработчику необходимо понимать как общую концепцию, так и детали
- Стоимость модификации старых программ многократно превосходит стоимость создания новых:
«Экономика — двигатель проектирования программ» (Эдвард Йордон, 1979г.)

S общ = S разработки + S сопровождения

S сопровождения = S понимания + S изменений + S тестирования + S поставки

Архитектурные стили

- *Структура*
 - Компонентная архитектура (Component-based)
 - Монолитное приложение (Monolithic application)
 - **Расслоение (Layered)**
- *Распределенные системы*
 - Клиент-сервер (Client-server)
 - Точка-точка (Peer-to-peer)
 - Сервис-ориентированная (Service-oriented)
- *Сообщения*
 - Событийная архитектура (Event-driven)
 - Публикация/Подписка (Publish-subscribe)
 - Шина сообщений (Message bus)

Архитектурные слои

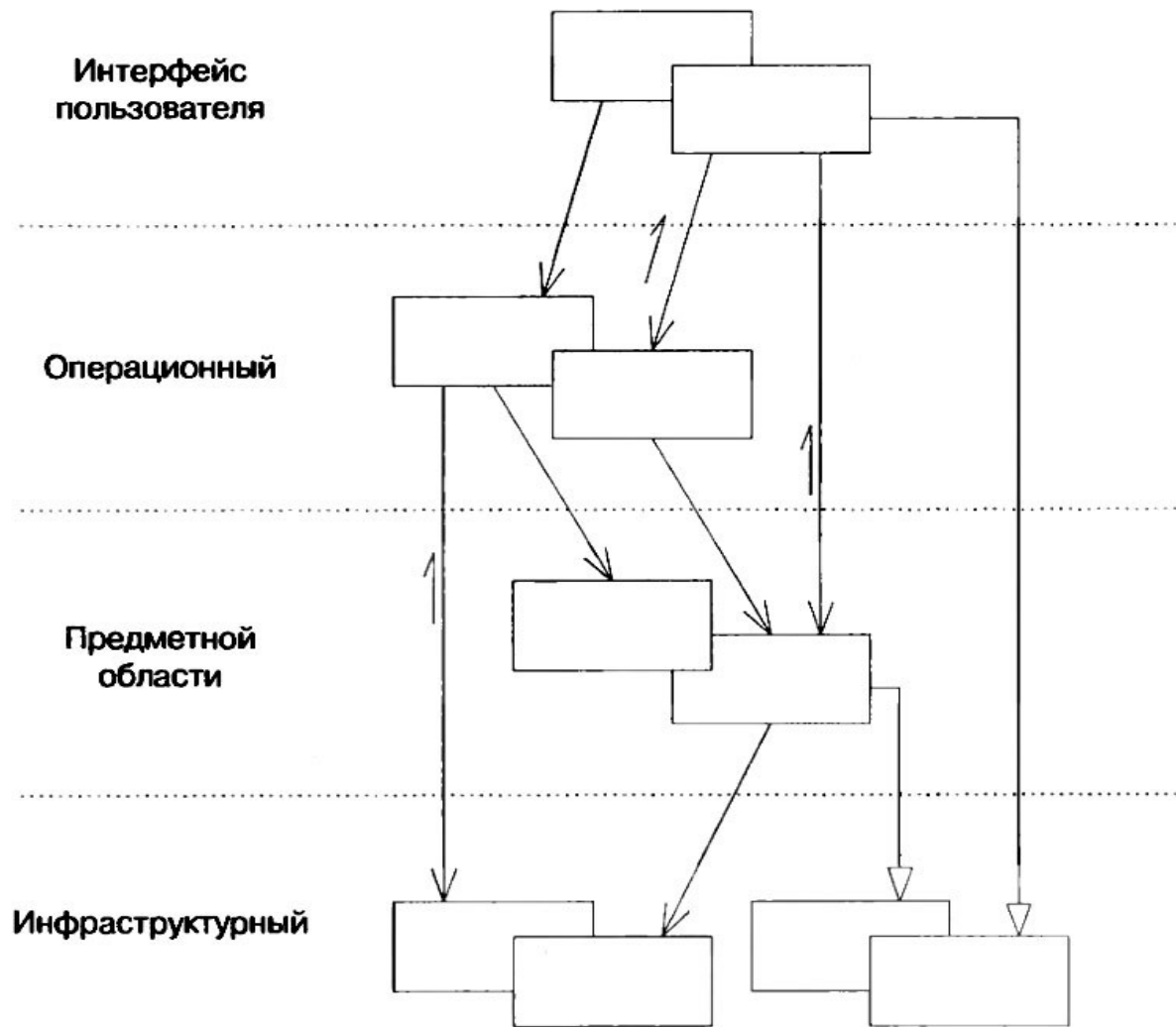
- «+»

- Слой — самодостаточное целое
- Возможна альтернативная реализация
- Минимальная зависимость слоев (Изоляция)
- Несколько вышележащих слоев

- «-»

- Каскадное изменение слоев
- Снижение производительности
- Размытые границы ответственности

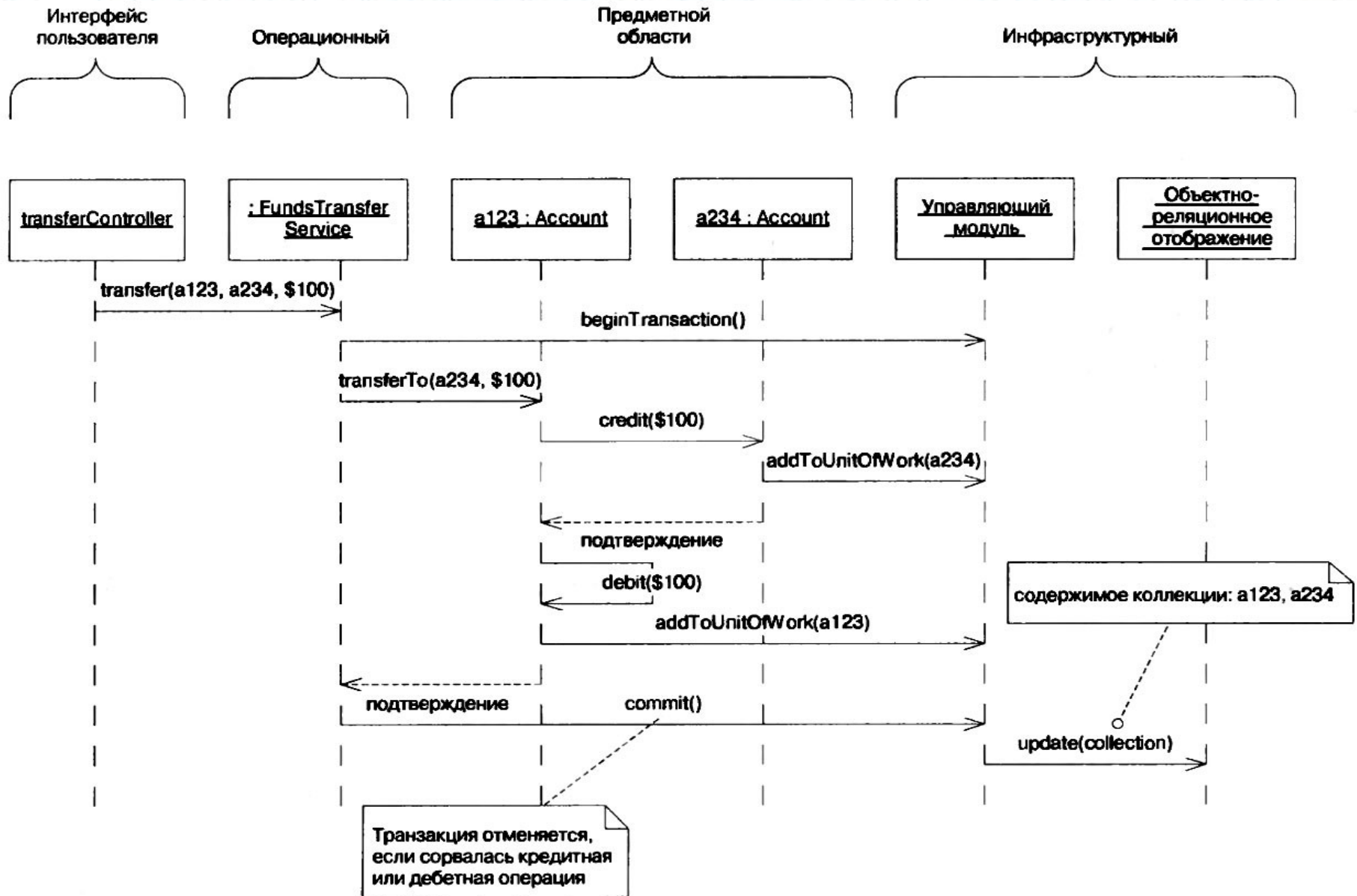
Концепция слоев



Расслоение информационной системы

- Представления (Интерфейсный)
 - Отображение
 - Обработка событий
- Сервисный (Операционный / Служб)
 - Координирование операций
 - Внешний API
- **Бизнес-логики (Предметной области / Домен)**
- Источников данных (Инфраструктурный)
 - Файловая система
 - СУБД
 - Сторонние API

Пример расслоения




Бизнес-логика

- Реализует основные функции приложения
- Не зависит от представления
- Чаще всего подвержена изменениям
- Меньше всего охвачена стеками технологий и средствами автоматизации проектирования
- Оперирует **сущностями предметной области**
- Определяется на **модели предметной области**

Процесс разработки ИС

- Сбор требований →
Варианты использования →
Сущности
- Анализ и проектирование →
Модель предметной области →
Сущности, отношения,
взаимодействие
- Реализация
- Тестирование



**Модель
предметной
области**

Модель предметной области

- Модель — строго организованная выборка знаний предметной области и способ их структуризации [Эрик Эдванс]
- Модель и архитектура взаимно определяют друг друга
- Модель определяет язык группы разработчиков
- Модель — дистиллированное знание, сосредоточенное на сути проблемы
- UML — язык описания модели

UML

- Унифицированный язык моделирования (не метод и не процесс моделирования)
- Является стандартом консорциума OMG (Object Management Group)
- Графическая нотация, средство общения
- Допускает использование неформализованных, интуитивных элементов

UML диаграммы

- Структурные диаграммы
 - **Диаграммы классов**
 - Диаграммы пакетов
 - Физические диаграммы (развертывания)
 - ...
- Диаграммы поведения
 - **Диаграммы вариантов использования**
 - **Диаграммы последовательности**
 - Диаграммы деятельности
 - Диаграммы состояний
 - ...

Варианты использования (прецеденты)

- Отражают функциональные требования
- Описывают сценарии работы
- Выявляют роли
 - Пользователи
 - Внешнее ПО
- Оперируют сущностями (объектами) словесно
- Имеют отношения
 - Включение (часть-целое)
 - Обобщение (частный случай прецедента)
 - Расширение (дополнение прецедента)

UML. Диаграмма вариантов использования

Иллюстрации: UML. Основы. 2-е издание. Фаулер М. Скотт К.



Варианты использования

Покупка товара

1. Покупатель просматривает каталог и выбирает товары для покупки.
2. Покупатель оценивает стоимость всех товаров.
3. Покупатель вводит информацию, необходимую для доставки товара.
4. Система предоставляет полную информацию о цене товара и его доставке.
5. Покупатель вводит информацию о кредитной карточке.
6. Система осуществляет авторизацию счета покупателя.
7. Система выполняет немедленную оплату товаров.
8. Система подтверждает оплату товаров для покупателя по адресу его электронной почты.

Альтернатива: Неудача авторизации

На шаге 6 система получает отрицательный ответ на запрос о состоянии счета покупателя.

Необходимо предоставить покупателю возможность повторно ввести информацию о кредитной карточке и выполнить ее авторизацию.

Альтернатива: Постоянный покупатель

3а. Система предоставляет информацию о текущей покупке и ее цене, а также последние 4 цифры информации о кредитной карточке.

3б. Покупатель может согласиться или отказаться от предложенной системой информации.

После этого перейти на шаг 6 исходного сценария.

Объект (сущность)

«Конкретный опознаваемый предмет, единицу или сущность (реальную или абстрактную), имеющую четко определенное функциональное назначение в данной предметной области» [Гради Буч]

Объект характеризуется:

- Идентичностью
- Состоянием
- Поведением

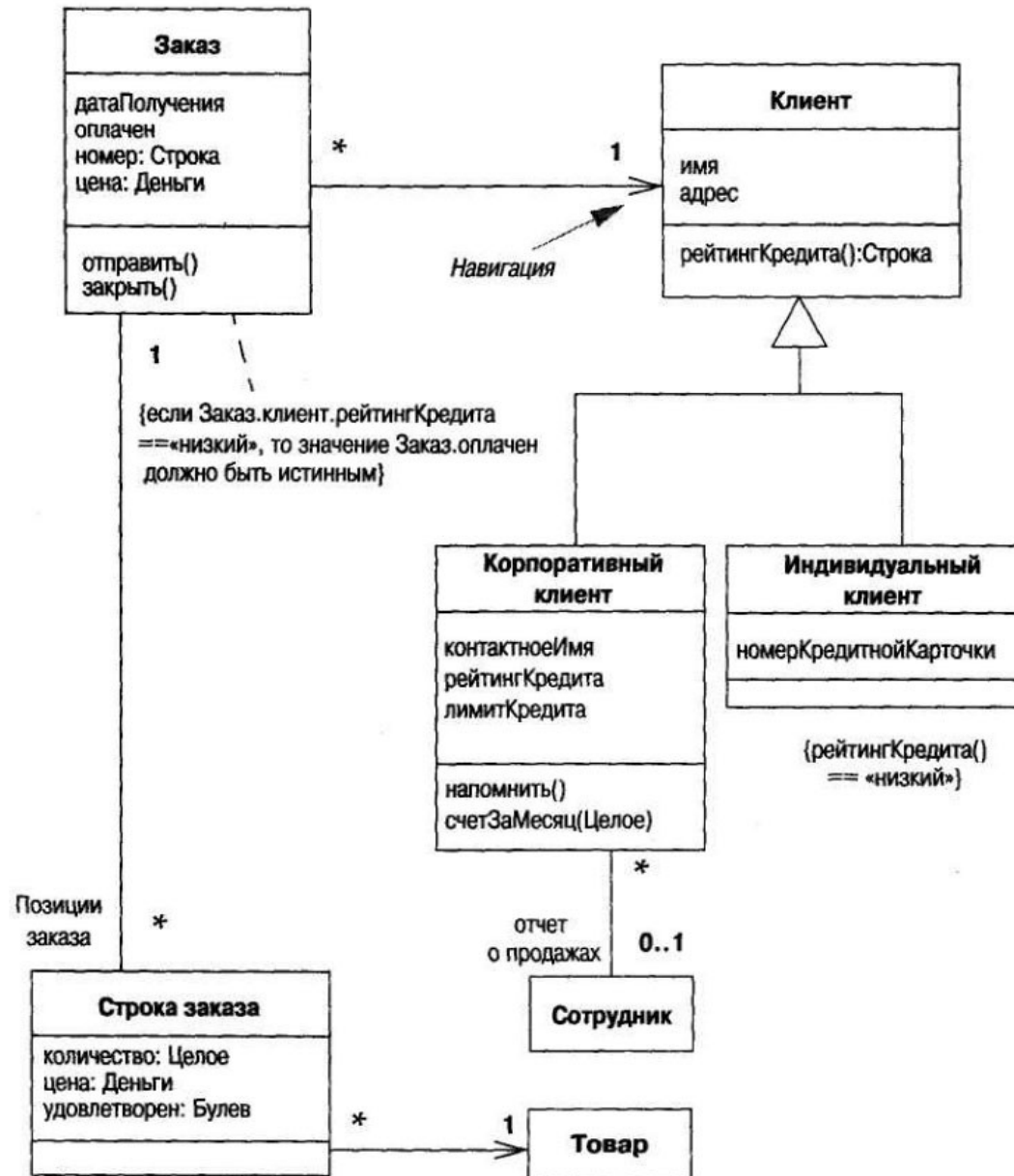
Идентичность объекта

- Совокупность отличительных черт
- Самодостаточность
- Отчуждаемость
- Свойства
 - Статические
 - Динамические
- Свойства могут быть объектами

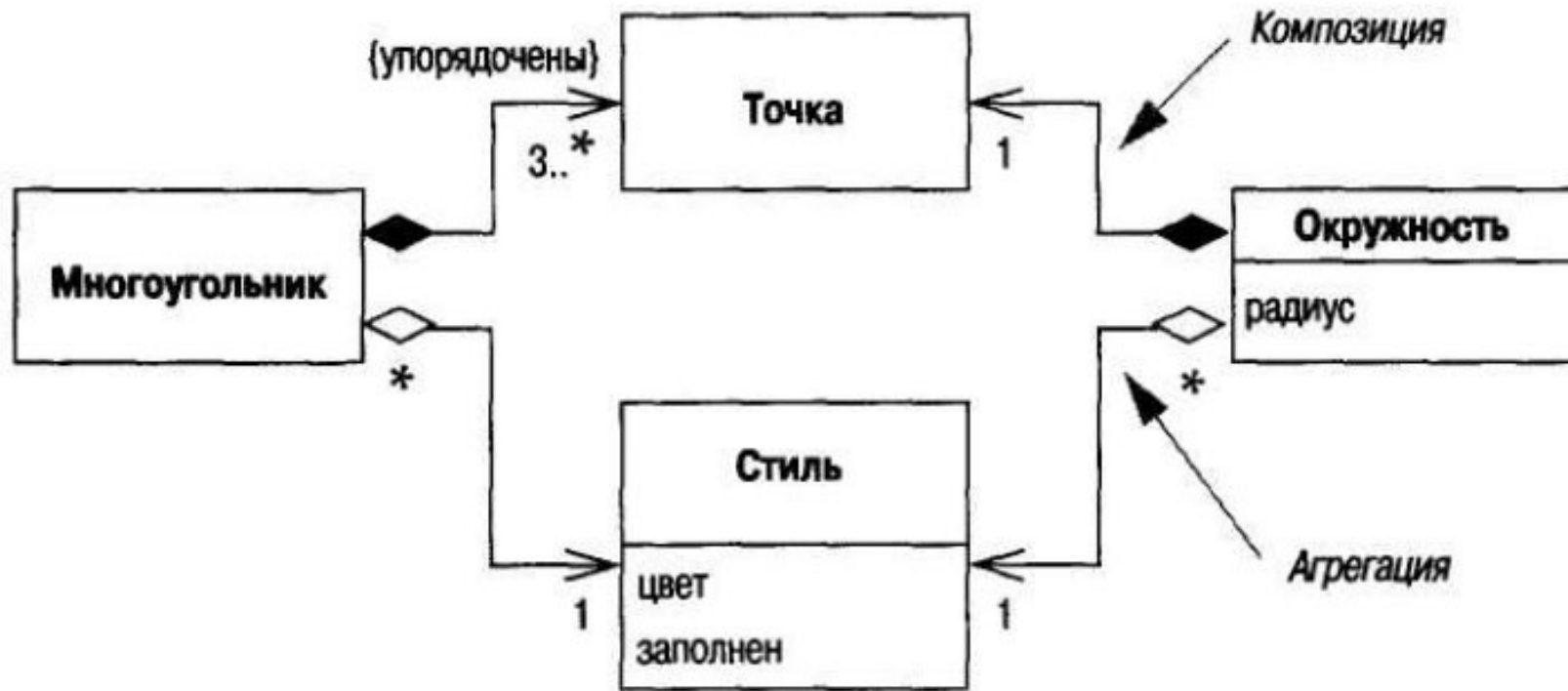
UML. Диаграмма классов

- Описывает типы объектов системы
- Определяет статические отношения
 - Ассоциация
 - Простая («связано»)
 - Агрегация («содержит»)
 - Композиция («состоит из»)
 - Зависимость («зависит от»)
 - Обобщение
 - Расширение
 - Реализация

UML. Диаграмма классов



UML. Диаграмма классов



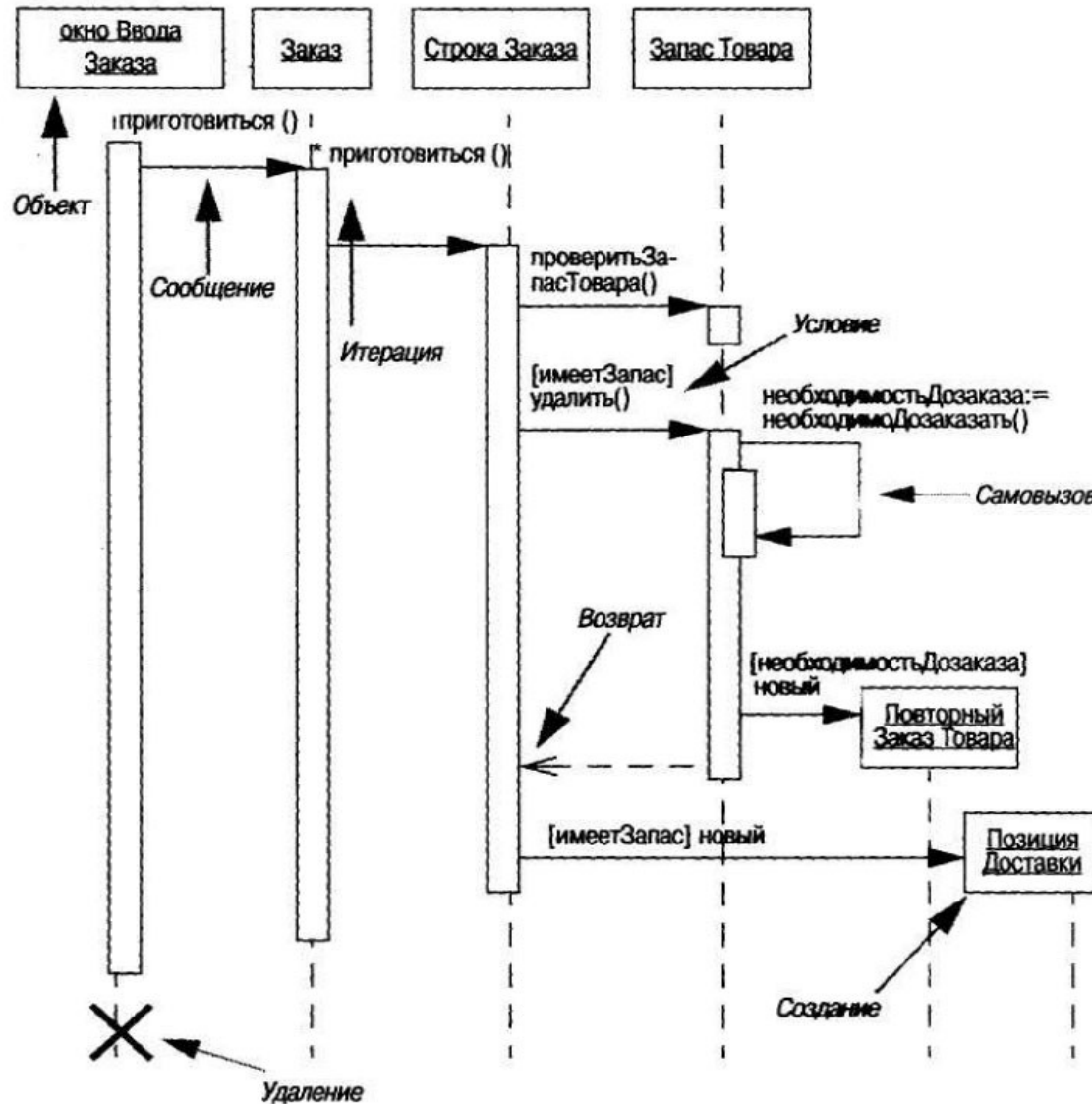
Поведение объекта

- Объекты подвергаются воздействию
- Объекты воздействуют на другие объекты
- Поведение - реакция и действие объекта
- Поведение выражается в терминах состояния объекта и передачи сообщений
- Поведение определяется состоянием и операцией (воздействием)
- Классификация операций [Липпман]:
 - Функции управления
 - Функции реализации
 - Модификаторы — изменение состояния объекта
 - Селекторы — считывание состояния объекта

UML. Диаграммы последовательности

- Описывают взаимодействие группы объектов
- Раскрывают один вариант использования
- Для каждого объекта изображается «линия жизни»
- Сообщения между объектами — стрелки между «линиями жизни»

UML. Диаграммы последовательности



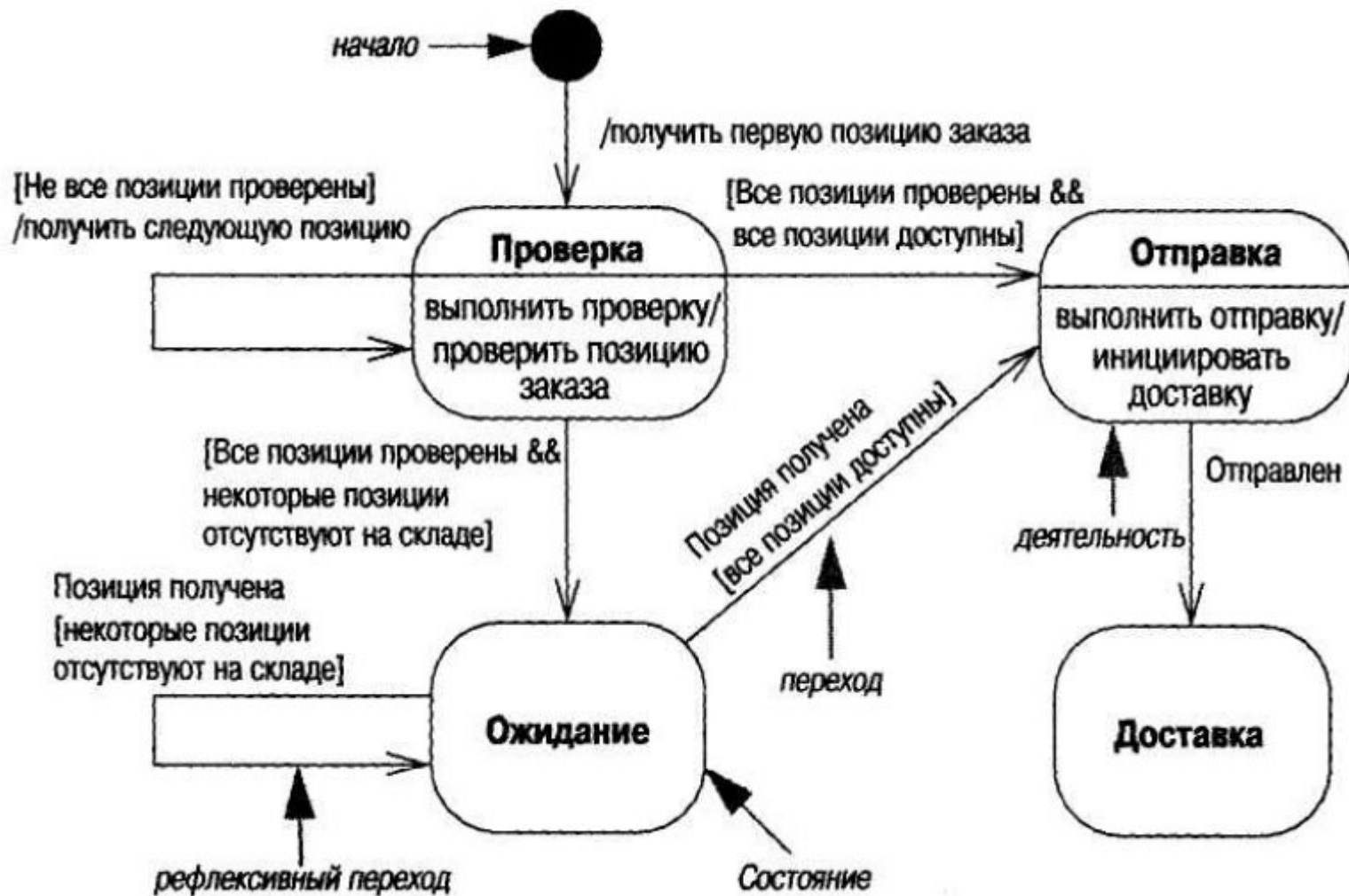
Состояние объекта

- Детерминировано
- Композиция значений свойств объектов
- Изменяется под внешним воздействием
- Объект имеет жизненный цикл

UML. Диаграммы состояний

- Описывают поведение объекта
- Отображают все возможные состояния объекта
- Состояние характеризуется *деятельностью*
- Отображают переходы между состояниями
- Переход характеризуется:
 - Событием — инициация перехода
 - Сторожевым условием — условие перехода
 - *Непрерываемым действием*
- Состояния могут объединяться в суперсостояния

UML. Диаграммы состояний



далее..

Архитектурные шаблоны реализации бизнес-логики

Курсовое проектирование

- 1) Сбор функциональных требований
- 2) Разработка вариантов использования (обобщенная диаграмма прецедентов)
- 3) Подробное описание всех вариантов использования (текстовое описание с альтернативами)
- 4) Разработка статической объектной модели предметной области (диаграммы классов)
- 5) Разработка динамической объектной модели предметной области (диаграммы последовательности)
- 6) Проектирование слоя бизнес-логики (выбор архитектурного шаблона уровня бизнес-логики)
- 7) Реализация слоя бизнес-логики (Java, NetBeans), тестирование (JUnit)
- 8) Проектирование слоя источников данных (выбор архитектурного шаблона уровня доступа к данным: DB + внешний сервис)
- 9) Реализация слоя источников данных (JavaDB), тестирование
- 10) Проектирование сервисного слоя и слоя представления: GUI (Swing), внешний сервис
- 11) Реализация слоев представления, сервисного слоя, тестирование
- 12) Комплексное тестирование системы
- 13) Пояснительная записка