АО «АЛМАТИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»
https://atu.edu.kz/

Факультет «Информационные технологии»
Кафедра «Компьютерная инженерия»

Название курса: «Инструментальные средства разработки программ»



Зиятбекова Гулзат Зиятбеккызы – PhD, ассоциированный профессор
g.ziyatbekova@atu.edu.kz

ЛЕКЦИЯ №1

Введение. Единый язык моделирования UML

СОДЕРЖАНИЕ

План лекции

1

Основы

Что такое UML, зачем нужно моделирование, история языка

2

Принципы

Принципы моделирования и строительные блоки языка

3

Диаграммы

Структурные и поведенческие диаграммы, отношения

4

Практика

Применение UML, сравнение с IDEF0 и DFD, инструменты

ВВЕДЕНИЕ

Что такое UML

Определение

Unified Modeling Language — унифицированный графический язык для визуализации, специфицирования, конструирования и документирования программных систем.

Язык, а не метод

UML задаёт нотацию и семантику, но не диктует процесс разработки. Его применяют в каскадных, итеративных и гибких методологиях.

Стандарт

Поддерживается консорциумом OMG, международный стандарт ISO/IEC 19501. Единая нотация понятна аналитикам, архитекторам и разработчикам.

14
типов диаграмм в UML 2.x

3
автора: Буч, Рамбо, Якобсон

1997
принят стандартом OMG

МОТИВАЦИЯ

Зачем нужно моделирование

1. **Управление сложностью**
   Модель скрывает несущественные детали и позволяет охватить систему целиком.

2. **Общий язык команды**
   Заказчик, аналитик и разработчик читают одни и те же диаграммы.

3. **Раннее выявление ошибок**
   Дефекты требований и архитектуры дешевле исправить до написания кода.

4. **Документирование**
   Модель фиксирует решения и облегчает сопровождение и передачу проекта.

5. **Повторное использование**
   Проверенные архитектурные решения переносятся в новые проекты.

ИСТОРИЯ

Как появился UML

1980-е

Война методов

Десятки несовместимых нотаций объектно-ориентированного анализа и проектирования.

1994

Объединение

Гради Буч и Джеймс Рамбо в Rational Software начинают создавать единый метод.

1995

Якобсон

К ним присоединяется Ивар Якобсон с методом OOSE и вариантами использования.

1997

UML 1.0

Спецификация принята организацией OMG как стандарт.

2005

UML 2.0 и ISO

Появляются 14 типов диаграмм, UML становится стандартом ISO/IEC 19501.

Современная редакция — UML 2.5.1. Три автора получили прозвище «три амиго».

ТЕОРИЯ

Принципы моделирования

1

Выбор модели

От выбранной модели зависит, как будет решаться задача и какое решение получится.

2

Уровни абстракции

Любая модель может быть выражена с разной степенью точности: от общей схемы до деталей.

3

Связь с реальностью

Лучшие модели ближе к реальности. Упрощение не должно скрывать критичных свойств.

4

Несколько моделей

Одной модели недостаточно: нужен набор почти независимых взглядов на систему.

СТРУКТУРА ЯЗЫКА

Строительные блоки UML

Сущности

Структурные: класс, интерфейс, компонент, узел, вариант использования.
Поведенческие: взаимодействие, состояние.
Группирующие: пакет.
Аннотационные: примечание.

Отношения

Зависимость — изменение одного влияет на другой.
Ассоциация — связь между объектами.
Обобщение — наследование.
Реализация — класс реализует интерфейс.

Диаграммы

Графическое представление набора элементов в виде графа вершин и рёбер. Строятся для визуализации системы с разных точек зрения. Всего 14 типов.

Расширение языка: стереотипы, помеченные значения и ограничения.

СТАТИКА

Структурные диаграммы

Диаграмма	Что показывает	Когда применяют
Классов	Классы, атрибуты, операции, связи	Проектирование структуры
Объектов	Экземпляры классов в момент времени	Иллюстрация примеров
Компонентов	Модули и их интерфейсы	Архитектура системы
Развёртывания	Размещение ПО на узлах	Планирование инфраструктуры
Пакетов	Группировка элементов модели	Организация крупных моделей
Составной структуры	Внутреннее устройство классов	Детальная архитектура
Профилей	Расширения языка	Предметные области

ДИНАМИКА

Диаграммы поведения

Диаграмма	Что показывает	Когда применяют
Вариантов использования	Функции системы глазами актёров	Анализ требований
Деятельности	Действия, ветвления, параллелизм	Бизнес-процессы, алгоритмы
Состояний	Жизненный цикл объекта	Реактивные системы
Последовательности	Обмен сообщениями во времени	Сценарии взаимодействия
Коммуникации	Связи объектов и сообщения	Структура взаимодействия
Обзора взаимодействий	Связь сценариев в общий поток	Сложные сценарии
Времени	Изменения состояний по времени	Системы реального времени

СВЯЗИ

Отношения между элементами

1

Зависимость

Пунктирная стрелка. Клиент использует поставщика и зависит от его изменений.

2

Ассоциация

Сплошная линия с кратностью. Частные случаи: агрегация и композиция («часть–целое»).

3

Обобщение

Сплошная линия с полым треугольником. Потомок наследует свойства предка.

4

Реализация

Пунктир с полым треугольником. Класс выполняет контракт интерфейса.

ПРАКТИКА

Применение UML на этапах разработки

Этап	Задачи	Диаграммы
Анализ требований	Выявить функции и участников	Вариантов использования, деятельности
Проектирование	Описать структуру и взаимодействие	Классов, компонентов, последовательности
Реализация	Генерация кода, обратная инженерия	Классов, состояний
Тестирование	Сценарии и проверка поведения	Последовательности, вариантов использования
Внедрение	Описать развёртывание	Развёртывания, компонентов

СРАВНЕНИЕ

UML, IDEFo и DFD

Нотация	Что описывает	Назначение
UML	Структуру и поведение программных систем	Объектно-ориентированное проектирование
IDEF0	Функции системы, входы, выходы, управление, механизмы	Функциональное моделирование бизнес-процессов
DFD	Потоки данных между процессами и хранилищами	Анализ обработки данных

Практика №1 и лабораторная работа №1 посвящены построению диаграмм IDEF0 и DFD.

ИНСТРУМЕНТЫ

Средства построения диаграмм

Профессиональные CASE

Enterprise Architect, Visual Paradigm, StarUML: полный набор диаграмм, генерация кода, обратная инженерия.

Онлайн-редакторы

draw.io (diagrams.net), Lucidchart: быстрый старт, совместная работа без установки.

Диаграммы как код

PlantUML, Mermaid: модель описывается текстом и хранится вместе с кодом проекта.

ИТОГИ

Выводы

1. **UML — стандартный язык**
   Единая нотация для моделирования программных систем, стандарт OMG и ISO.

2. **14 типов диаграмм**
   Структурные описывают статику, поведенческие — динамику системы.

3. **Модель снижает сложность**
   Помогает находить ошибки на ранних этапах и улучшает коммуникацию.

4. **UML дополняет IDEF0 и DFD**
   Эти нотации применяют на этапе анализа процессов и данных.

ОБСУЖДЕНИЕ

Вопросы для обсуждения

1. **Модель и реальность**
   Чем модель отличается от реальной системы и какие детали допустимо опустить?

2. **Набор диаграмм**
   Почему одной диаграммы недостаточно для описания системы?

3. **Выбор диаграмм**
   Какие диаграммы нужны на этапе анализа, а какие при проектировании?

4. **Границы применения**
   Когда UML оправдан, а когда достаточно набросков на доске?