The background is a collage of four images: top-left shows a stack of papers on a blue background; top-right shows a clock on a red background; bottom-left shows papers on a green background; bottom-right shows a clock on a yellow background.

Объектно-ориентированное моделирование. Unified Process and UML

БГТУ – 2020

лектор: Парамонов А.И.

Сущность объектно-ориентированного подхода

закljučается в декомпозиции системы на классы, которые соответствуют однотипным объектам предметной области, и построении из них иерархии в виде ориентированного графа с использованием отношений композиции и наследования.



ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД

Структурный подход

Функциональная декомпозиция
(система = иерархия функций)

Данные (атрибуты) хранятся
отдельно от сущности

Иерархия модулей =
отношение композиции

модуль представляется в
виде дерева

ООП

Объектная декомпозиция
(система = набор объектов)

Объект = набор
атрибутивных данных
(характеристики) и
поведения системы (методы)

Иерархия отношений =
композиции и наследования

модуль представляется в
виде ориентированного
графа

Преимущества ООП проектирования:

- описание системы больше соответствует содержательному смыслу предметной области;
- сущности реального мира обладают поведением, что отражается с помощью определения методов класса;
- большая внутренняя и меньшая внешняя связности между компонентами системы;
- более легкая организация параллельных вычислений;
- большая степень автоматизации кодогенерации.





Методологии поддерживающие ООП:

- Унифицированный процесс (Unified Process, UP);
- экстремальное программирование (eXtreme Programming, XP);
- гибкое моделирование (Agile Modeling, AM).



Унифицированный процесс (Unified Process)

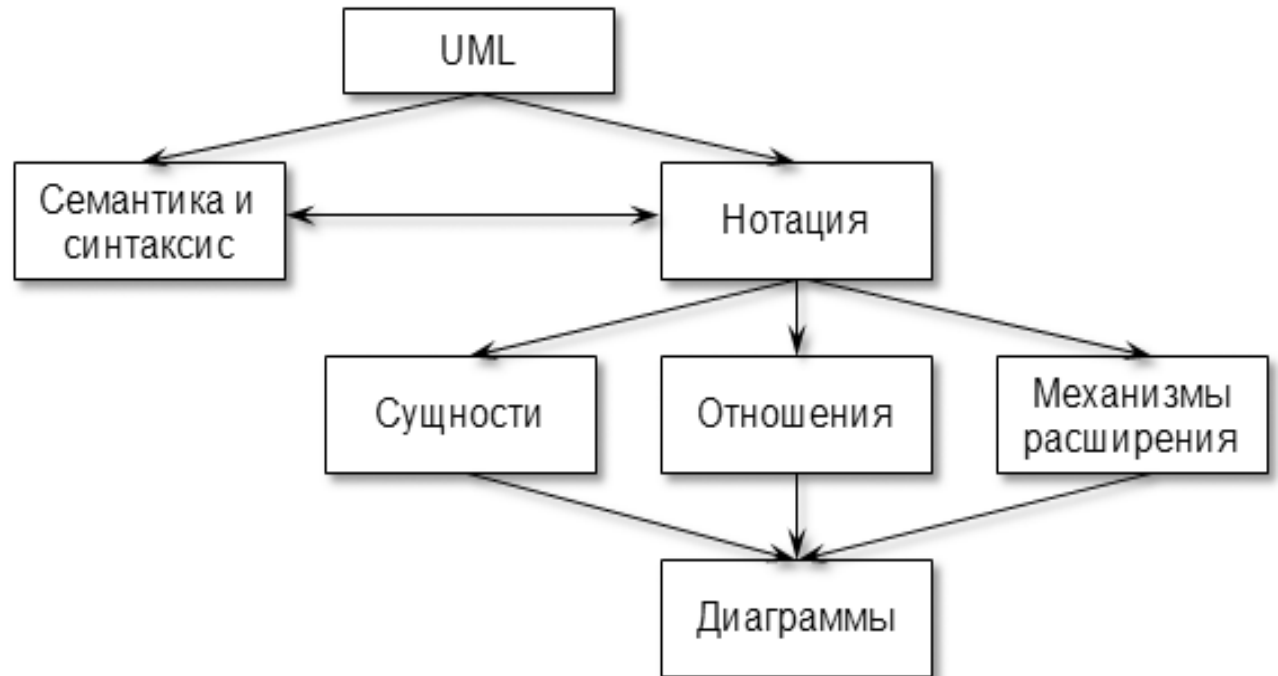
– это процесс разработки программного обеспечения (ПО), который обеспечивает упорядоченный подход к распределению задач и обязанностей в организации-разработчике.




Унифицированный язык визуального моделирования - Unified Modeling Language (UML)

— это стандартная нотация
визуального моделирования
программных систем, принятая
консорциумом **Object Managing
Group (OMG)** осенью 1997г., и
на сегодняшний день
поддерживаемая многими
объектно-ориентированными
CASE-продуктами.

Общая структура UML





Три режима использования UML разработчиками:

- режим эскиза
- режим проектирования
 - *прямая разработка (forward-engineering)*
 - *обратная разработка (reverse-engineering)*
- режим языка программирования



MDA (Model Driven Architecture
– архитектура, управляемая
моделью)



- ***PIM*** (Platform Independent Model
– модель, не зависящая от
платформы)



- ***PSM*** (Platform Specific Model –
модель, зависящая от
платформы)



исполняемый UML (Executable UML)



UML 2

(13 официальных типов диаграмм)

| № | Диаграмма | Цель |
|----|-----------------------|--|
| 1 | Деятельности | Процедурное и параллельное поведение |
| 2 | Классов | Классы, свойства и отношения |
| 3 | Взаимодействия | Взаимодействие между объектами; акцент на связях |
| 4 | Компонентов | Структура и взаимосвязи между компонентами |
| 5 | Составных структур | Декомпозиция класса во время выполнения |
| 6 | Развертывания | Развертывание артефактов в узлы |
| 7 | Обзора взаимодействий | Комбинация диаграммы последовательности и диаграммы деятельности |
| 8 | Объектов | Вариант конфигурации экземпляров |
| 9 | Пакетов | Иерархическая структура времени компиляции |
| 10 | Последовательности | Взаимодействие между объектами; акцент на последовательности |
| 11 | Конечных автоматов | Как события изменяют объект в течение его жизни |
| 12 | Временная | Взаимодействие между объектами; акцент на синхронизации |
| 13 | Прецедентов | Как пользователи взаимодействуют с системой |

Классификация типов диаграмм UML

■ Структурные диаграммы

- Диаграммы классов
- Диаграммы пакетов
- Физические диаграммы (развертывания)
- ...

■ Диаграммы поведения

- Диаграммы вариантов использования
- Диаграммы последовательности
- Диаграммы деятельности
- Диаграммы состояний
- ...



| ДИАГРАММА | | | НАЗНАЧЕНИЕ | МОДЕЛИ ИС | | |
|--|-------------------------------------|---|--|----------------------------------|-------------------------|------------------------------|
| | | | | По степени физической реализации | По отображению динамики | По отображаемому аспекту |
| <u>Вариантов использования</u> (use case) | | | Отображает функции ИС, взаимодействие между актерами и функциями | Логическая | Статическая | Функциональная |
| <u>Классов</u> (class) | | | Отображает набор классов, интерфейсов и отношений между ними | Логическая или физическая | Статическая | Функционально-информационная |
| <u>Пакетов</u> (package) | | | Отображает набор пакетов и отношений между ними | Логическая или физическая | Статическая | Компонентная |
| Поведения (behavior) | <u>Автоматов</u> (state machine) | | Отображает состояния сущности и переходы между ними в процессе ее жизненного цикла | Логическая | Динамическая | Поведенческая |
| | <u>Деятельности</u> (activity) | | Отображает бизнес-процессы в системе (описание алгоритмов поведения) | | | |
| | Взаимодействия (interaction) | <u>Последовательности</u> (sequence) | Отображает последовательность передачи сообщений между объектами и актерами | | | |
| | | <u>Коммуникации</u> (communication) | Аналогична диаграмме последовательности, но основной акцент делается на структуру взаимодействия между объектами | | | |
| Реализации (implementation) | <u>Компонентов</u> (component) | | Отображает компоненты ИС (программы, библиотеки, таблицы и т.д.) и связи между ними | Физическая | Статическая | Компонентная |



- При разработке отдельной модели системы в Унифицированном процессе строят **несколько видов диаграмм**.
- При разработке модели сложной системы, как правило, строят **несколько диаграмм одного и того же вида**.
- Можно **не создавать отдельные виды диаграмм**, если в этом нет необходимости.
- Часть диаграмм после их построения требует развития и уточнения в рамках разработки следующей модели (технологического процесса).

Г.Буч, Д.Рамбо, А.Джекобсон

Язык UML

Руководство пользователя

