

Графические примитивы UML стандарта**Парадигмы UML стандарта****А Словарь языка****Б Правила над словарем****В Механизмы****Словарь языка**

- Сущности
- Отношения
- Диаграммы
- Аннотирующие сущности

Сущности

- Структурны
- Поведенческие
- Группирующие

Поведенческие сущности

- Взаимодействие
- Автоматы

Группирующие сущности

- Пакеты

Структурные сущности

- Прецеденты
- Классы
 - Актеры
 - Сигналы
 - Утилиты
- Интерфейсы
- Активные классы
 - Процессы
 - Нити
- Кооперации
- Компоненты
- Узлы

Аннотирующие сущности

- Примечания

Правила над словарем

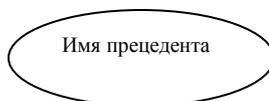
Правила использования графических примитивов в диаграммах.

Механизмы

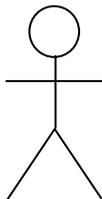
Механизмы построения диаграмм.

Основные понятия**Прецедент**

Это последовательность действий, значимых для некоторого актера.

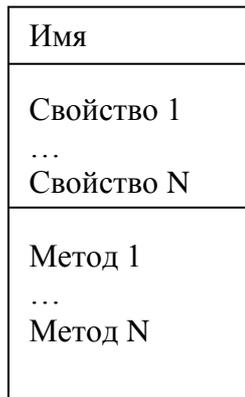
**Актер**

Это внешняя по отношению к системе сущность.

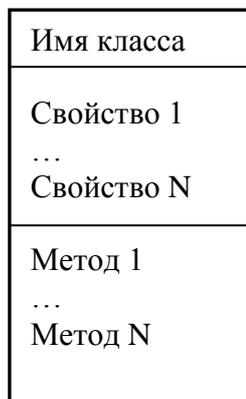


Класс

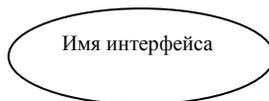
Являет собой статическую часть модели, соответствующую концептуальному или физическому элементу системы.

**Активный класс**

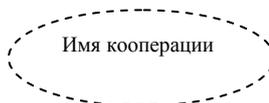
Это класс, отличающийся тем, что его объекты активны во время работы других объектов системы и владеют потоком или процессом с возможностью генерации управляющих воздействий.

**Интерфейс**

Это совокупность обязанностей класса, его сервис, набор услуг, которые он предоставляет.

**Кооперация**

Это совокупность ролей, которые играет класс в своих взаимодействиях так, что кооперативный эффект не есть просто сумма ролей.

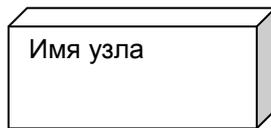
**Компонент**

Это физический образ логических элементов и их интерфейсов.



Узел

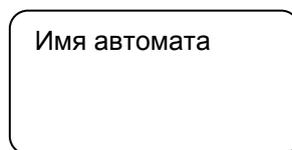
Это физический образ коммутирующего ресурса, обладающего собственной памятью и средствами обработки.

**Взаимодействие**

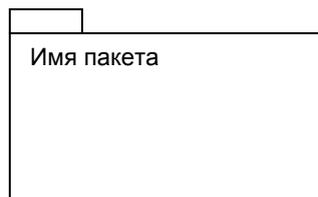
Это поведение, в основе которого лежит сообщение (событие).

**Автомат**

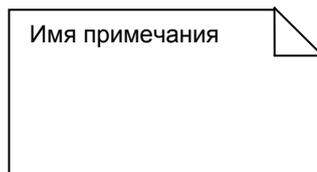
Это алгоритм поведения, отражающий последовательность состояний, через которую объект или взаимодействие проходят на протяжении своего жизненного цикла в ответ на поток событий.

**Пакет**

Это организационная, группирующая сущность концептуального уровня проекта, включающая структурные, поведенческие сущности или другие пакеты.

**Примечание**

Это пояснения к элементу проекта.

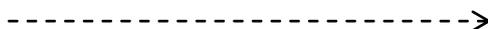
**Отношения**

- Зависимость
- Ассоциация
- Обобщение
- Агрегация
- Композиция

Зависимость

Это такое отношение между сущностями, когда изменение одной может отразиться на другой.

Имя зависимости



Исходная сущность Зависимая сущность

Ассоциация

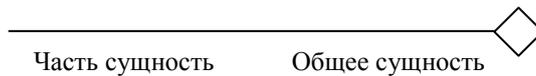
Это связь равноправных сущностей.

**Обобщение**

Это наследование одной сущности другой.

**Агрегация**

Это когда одна сущность есть часть другой сущности.

**Композиция**

Это агрегация, когда часть не может существовать в отрыве от целого.

**Диаграммы UML стандарта**

Диаграммы – используются для графического отображения различных аспектов системы:

- структуры
- состояний
- архитектуры

с целью облегчения восприятия и понимания работы системы.

Хорошо продуманная система обозначений важна при разработке системы по следующим причинам:

- способствует пониманию системы
- являясь стандартными графическими примитивами допускает использование CASE технологий

Виды диаграмм

1 Вариантов использования (Прецедентов)	User case
2 Классов	Class
3 Кооперации	Collaboration
4 Последовательности	Sequence
5 Состояний	StateChart
6 Деятельности	Activity
7 Компонентов	Component
8 Развертывания	Deployment

1 Анализ прецедентов.

Диаграммы прецедентов (Вариантов использования User Case)

Вариант использования (прецедент) – это последовательность действий (транзакций), выполняемых в ответ на внешнее событие, инициированное пользователем, другой системой, временем и т.п. – «внешним актером».

Виды «внешних актеров»

- 1 Пользователь
- 2 Другая система
- 3 Время

Пример

Система - «Частный предприниматель»

Вариант использования - «Сдача выручки в банк»

Что дают диаграммы Вариантов использования?

Ответ

Перечень «актеров», то есть классов, то есть объектов.

Схема документирования Варианта использования

Вариант использования «Название»

Кратко словами смысл

Предусловия

Кратко словами перечень того, что надо делать и иметь для работы перед началом Варианта использования.

Основной сценарий

Логистическая модель сценария прецеденте

Альтернативный сценарий 1

Логистическая модель альтернативного сценария прецеденте

Постусловия

Кратко словами перечень того, чем надо делать и иметь для завершения Варианта использования.

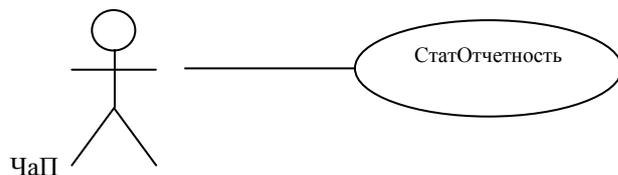
Отношения на диаграммах прецедентов

Ассоциация

Используется для обозначения связи между актером и прецедентом.

Обозначение

Пример



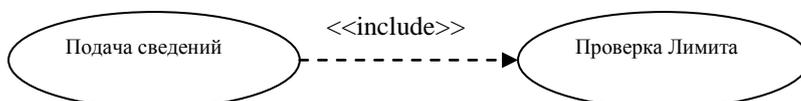
Включение

Используется для указания связи между базовым прецедентом и тем прецедентом, сценарий которого есть часть сценария базового прецедента. Используется слово <<include>>

Обозначение

----->

Пример



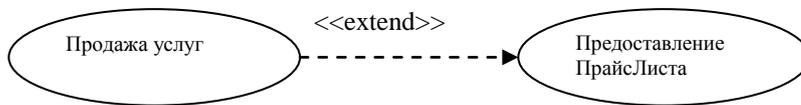
Расширение

Используется для указания связи между прецедентом и тем прецедентом, сценарий которого может быть использован базового прецедента при выполнении некоторых условий. Используется слово <<extend>>.

Обозначение



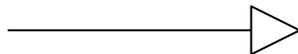
Пример



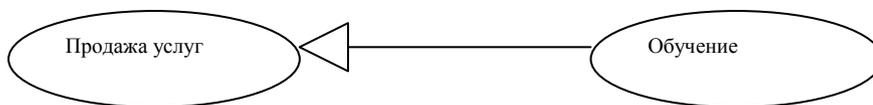
Обобщение

Используется для указания связи между предком прецедентом и потомком прецедентом, сценарий которого наследует сценарий предка прецедента с, может быть, его дополнением.

Обозначение



Пример

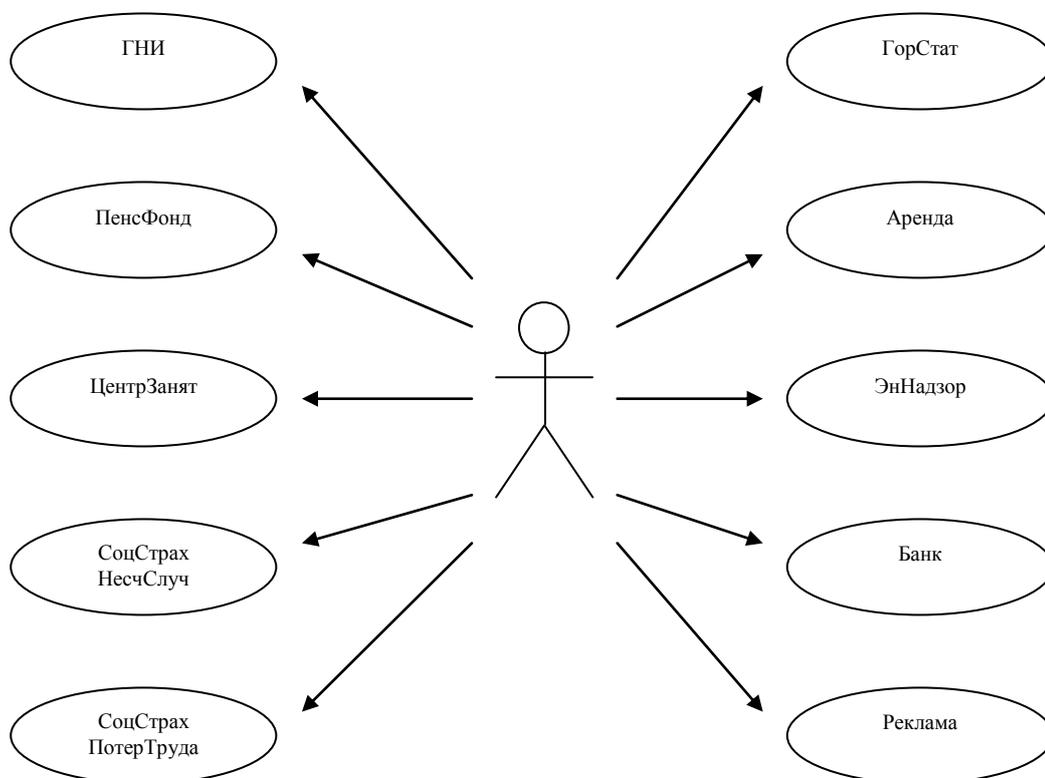


Замечание

Стрелка указывает на предка

Пример «дПрецедентов»

Диаграмма Прецедентов системы «Частное предприятия» (Стиль UML)



Пример «дПрецедентов»

Диаграмма Прецедентов системы «Частное предприятие» (Стиль Microsoft)

SystemChaP

- |_ EnNadzor
- |_ GNI
- |_ Stat
- |_ PensFond
- |_ CentrZanat
- |_ SocStrach.NechSluch
- |_ SocStrach.PoterTruda
- |_ Reclama
- |_ Client
- |_ Arenda

Как делать диаграммы Прецедентов (Вариантов использования?)**Ответ 1**

«Разработчик должен представить себя «внешним актером» и пожелать исполнения своего желания».

Пример

Разработчик, представив себя предпринимателем (ЧаП) рассуждает - «Надо подавать сведения в ЭнергоНадзор».

Представляет

Мн_ЭнНадзор – Км_ПодачаСведений – Длг_ПодачаСведений – СтрВв_Расход – ввожу значение расхода электро-энергии и другое необходимое – Кн_Ок

Рисует

диаграмму Варианта использования “Подача сведений в энергонадзор”

System_ChaP

- |_ EnNadzor
- |_ Svedenia

Ответ 2**Пример**

«Разработчик, представив себя сервером (системой) откликается на желание клиента подать сведения в Статистическое управление».

Представляет

Вкл_Камп – Мн_Статистика – Км_ПодачаСведений – Длг_ПодачаСведений – ввожу сведения – Кн_Ок

Рисует

диаграмму варианта использования “Подача сведений в Статистическое управление”

System_ChaP

- |_ EnNadzor
- |_ Svedenia
- |_ Statistica
- |_ Svedenia