

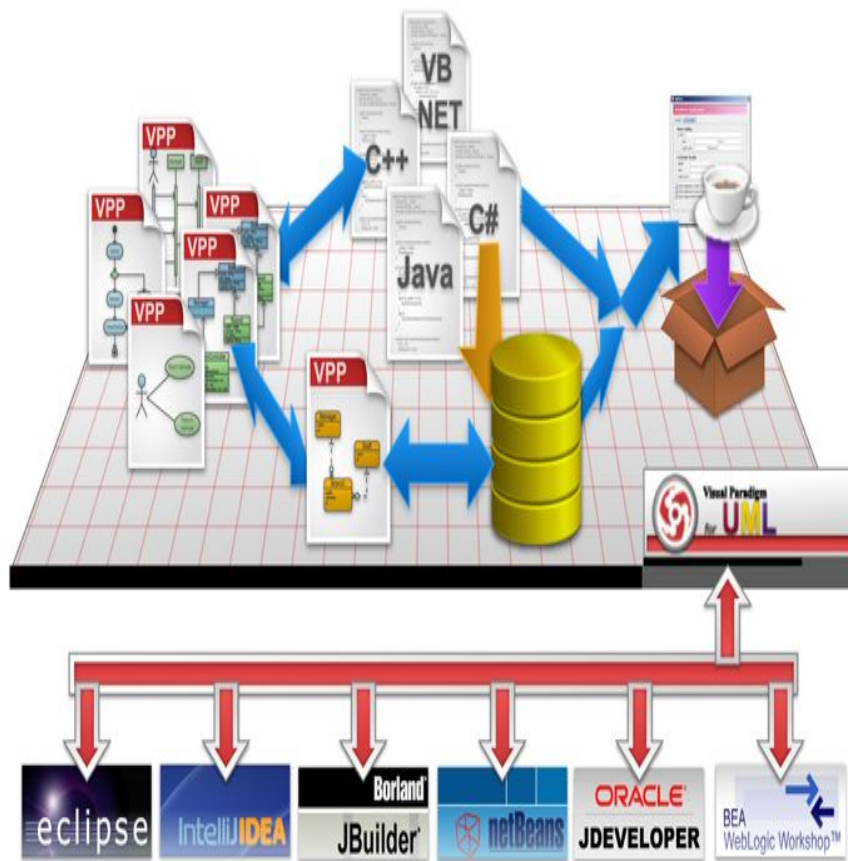
UNIFIED  
MODELING  
LANGUAGE



# Визуальное моделирование

В последнее время наблюдается общее повышение интереса ко всем аспектам, связанным с разработкой сложных программных приложений. Для многих компаний корпоративное программное обеспечение и базы данных (БД) представляют стратегическую ценность. Существует высокая заинтересованность в разработке и проверке методов и подходов, позволяющих автоматизировать процесс создания сложных программных информационных систем (ИС). Известно, что систематическое использование таких методов дает возможность значительно улучшить качество, сократить стоимость и время поставки ИС. Сегодня к этим методам можно отнести:

- компонентную технологию разработки моделей ИС;
- визуальное программирование (RAD-средства);
- использование образцов (patterns) при проектировании ИС;
- визуальное представление различных аспектов проекта (визуальное моделирование, CASE-средства).



- Визуальные модели широко используются в существующих технологиях управления проектированием систем, сложность, масштабы и функциональность которых постоянно возрастают
- Построить модель корпоративной ИС до ее программной разработки или до начала проведения архитектурной реконструкции столь же необходимо, как подготовить проектные чертежи, прежде чем приступить к строительству большого здания. Хорошая модель ИС позволит наладить плодотворное взаимодействие между заказчиками, пользователями и командой разработчиков, обеспечит ясность представления выбранных архитектурных решений и позволит "охватить" систему во всей ее полноте.

При проектировании сложной ИС ее разбивают на части (подсистемы), каждая из которых затем рассматривается отдельно. Возможны два различных способа такого разбиения: структурное (или функциональное) разбиение и объектная (компонентная) декомпозиция.

декомпозиции программной системы ее структура может быть

СВВ), опирается на иной принцип декомпозиции —

Если при проектировании информационная система разбивается на объекты (компоненты), то ее визуальное моделирование может осуществляться с помощью UML

С то следующим образом: он предоставляет выразительные средства для создания визуальных моделей, которые одинаково понимаются всеми разработчиками, вовлеченными в проект, и являются инструментом коммуникации в рамках проекта.

# Как создавался UML

В середине 90-х существовало более 50 различных объектно-ориентированных языков моделирования. И разработчики, и заказчики испытывали беспокойство при выборе метода проектирования ИС, который, как правило, включал в себя и собственную нотацию.

Возникла насущная потребность в их стандартизации и унификации.

Разработка UML была начата в октябре 1994 г. Грэди Бучем (Grady Booch) и Джимом Рамбо (Jim Rumbaugh) в Rational Software Corporation как унификация двух методов: Booch'93 и OMT. Первая версия унифицированного метода (Unified Method 0.8) была опубликована в октябре 1995 г. Тогда же к работе присоединился Айвер Якобсон (Ivar Jacobson), включив в процесс унификации свой метод OOSE. Таким образом, на первом концептуальном этапе UML имел трех авторов: Буча, Рамбо и Якобсона, каждый из которых являлся идеологом своего объектно-ориентированного метода визуального моделирования\*.

В октябре 1996 г. вышла редакция UML 0.91, в которой были учтены многочисленные пожелания пользователей, полученные "тремя друзьями" в течение 1996 г. В это же время выяснилось, что некоторые влиятельные организации, связанные с компьютерным бизнесом, начали рассматривать UML как стратегический элемент своей деятельности.

# ЭТАПЫ ПРОЕКТИРОВАНИЯ ИС С ПРИМЕНЕНИЕМ UML

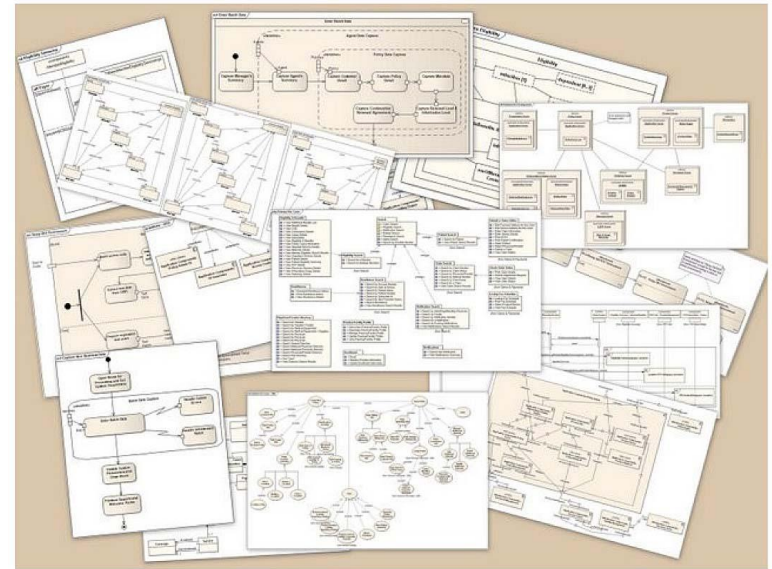
**UML** (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой *UML-моделью*. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.



UML обеспечивает поддержку всех этапов жизненного цикла ИС и предоставляет для этих целей ряд графических средств – диаграмм.

- На этапе создания концептуальной модели для описания бизнес-деятельности используются модели бизнес-прецедентов и диаграммы видов деятельности, для описания бизнес-объектов – модели бизнес-объектов и диаграммы последовательностей.
- На этапе создания логической модели ИС описание требований к системе задается в виде модели и описания системных прецедентов, а предварительное проектирование осуществляется с использованием диаграмм классов, диаграмм последовательностей и диаграмм состояний.
- На этапе создания физической модели детальное проектирование выполняется с использованием диаграмм классов, диаграмм компонентов, диаграмм развертывания.





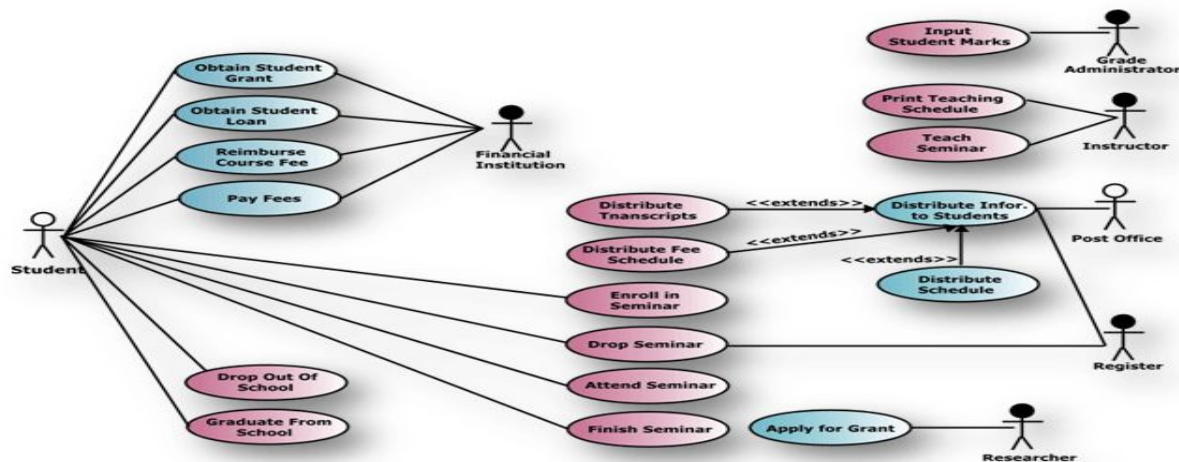
При визуальном моделировании на UML используются восемь видов диаграмм, каждая из которых может содержать элементы определенного типа. Типы допустимых элементов и отношений между ними зависят от вида диаграмм

- **Диаграммы прецедентов** (диаграммы вариантов использования, use case diagrams) – это обобщенная модель функционирования системы в окружающей среде.
- **Диаграммы видов деятельности** (диаграммы деятельностей, activity diagrams) – модель бизнес-процесса или поведения системы в рамках прецедента.
- **Диаграммы взаимодействия** (interaction diagrams) – модель процесса обмена сообщениями между объектами, представляется в виде диаграмм последовательностей (sequence diagrams) или кооперативных диаграмм (collaboration diagrams).
- **Диаграммы состояний** (statechart diagrams) – модель динамического поведения системы и ее компонентов при переходе из одного состояния в другое.
- **Диаграммы классов** (class diagrams) – логическая модель базовой структуры системы, отражает статическую структуру системы и связи между ее элементами.
- **Диаграммы базы данных** (database diagrams) — модель структуры базы данных, отображает таблицы, столбцы, ограничения и т.п.
- **Диаграммы компонентов** (component diagrams) – модель иерархии подсистем, отражает физическое размещение баз данных, приложений и интерфейсов ИС.
- **Диаграммы развертывания** (диаграммы размещения, deployment diagrams) – модель физической архитектуры системы, отображает аппаратную конфигурацию ИС.



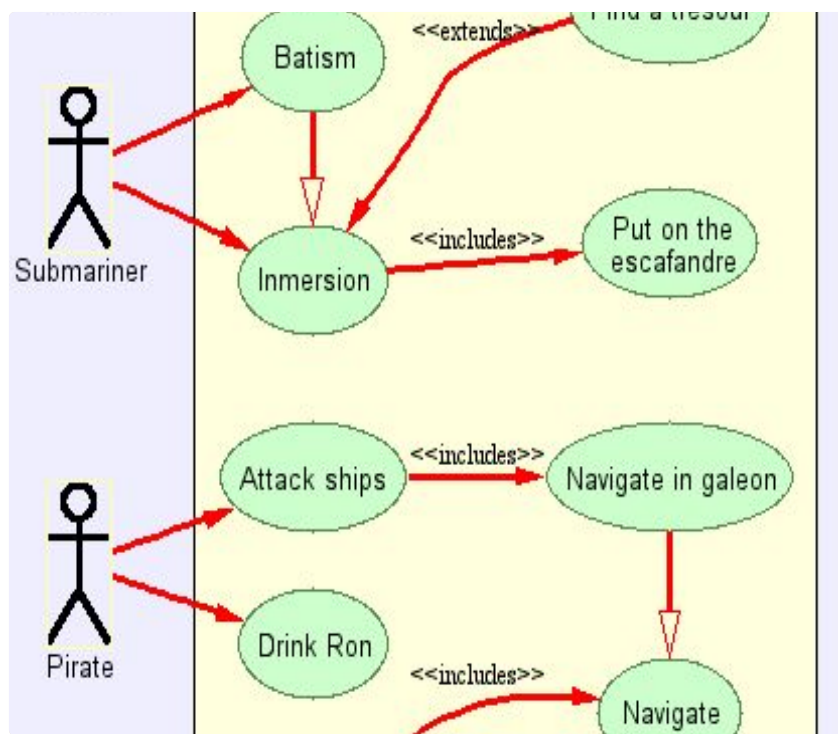
На рисунке показаны отношения между различными видами диаграмм UML

# Разработка модели бизнес-прецедентов



Deployment Diagram

• **Модель бизнес-прецедентов** описывает бизнес-процессы с точки зрения внешнего пользователя, т.е. отражает взгляд на деятельность организации извне.



Проектирование системы начинается с изучения и моделирования бизнес-деятельности организации. На этом этапе вводится и отображается в модели ряд понятий, свойственных объектно-ориентированному подходу:

- **Исполнитель (Действующее лицо, Actor) – личность, организация или система, взаимодействующая с ИС; различают внешнего исполнителя (который использует или используется системой, т.е. порождает прецеденты деятельности) и внутреннего исполнителя (который обеспечивает реализацию прецедентов деятельности внутри системы).**

На диаграмме исполнитель представляется стилизованной фигуркой человека.

- Прецедент — законченная последовательность действий, инициированная внешним объектом (личностью или системой), которая взаимодействует с ИС и получает в результате некоторое сообщение от ИС. Прецедент рисуется как овал, связанный с типичными пользователями, называемыми актерами (actors), (актантами).

- Класс — описание совокупности однородных объектов с их атрибутами, операциями, отношениями и семантикой. На диаграмме представляется прямоугольником, содержащим описания атрибутов и операций класса.

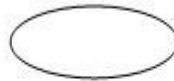
- Ассоциация – связь между двумя элементами модели. На диаграмме представляется линией.
- Обобщение – связь между двумя элементами модели, когда один элемент (подкласс) является частным случаем другого элемента (суперкласса). На диаграмме представляется



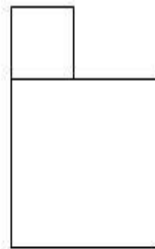
Класс



Актер



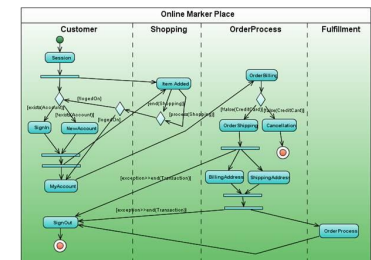
Прецедент  
(use case)




Пакет



Примечание

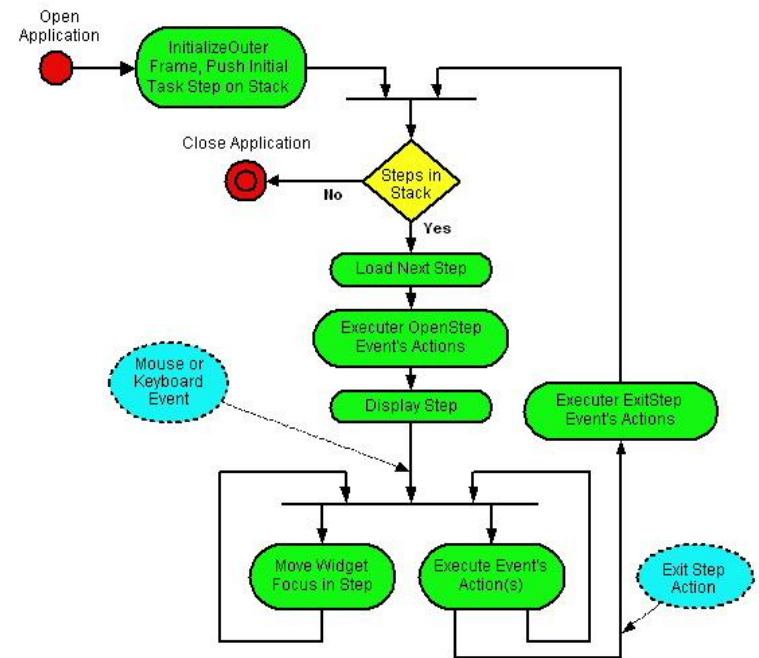




- 
- Агрегация – отношение между элементами модели, когда один элемент является частью другого элемента (агрегата). На диаграмме представляется стрелкой с ромбовидным концом.
  - Общее поле диаграммы деятельности делится на несколько «плавательных дорожек», каждая из которых содержит описание действий одного из исполнителей. Основными элементами диаграмм видов деятельности являются обозначения состояния («начало», «конец»), действия (овал) и момента синхронизации действий (линейка синхронизации, на которой сходятся или разветвляются несколько стрелок).

# Итак:

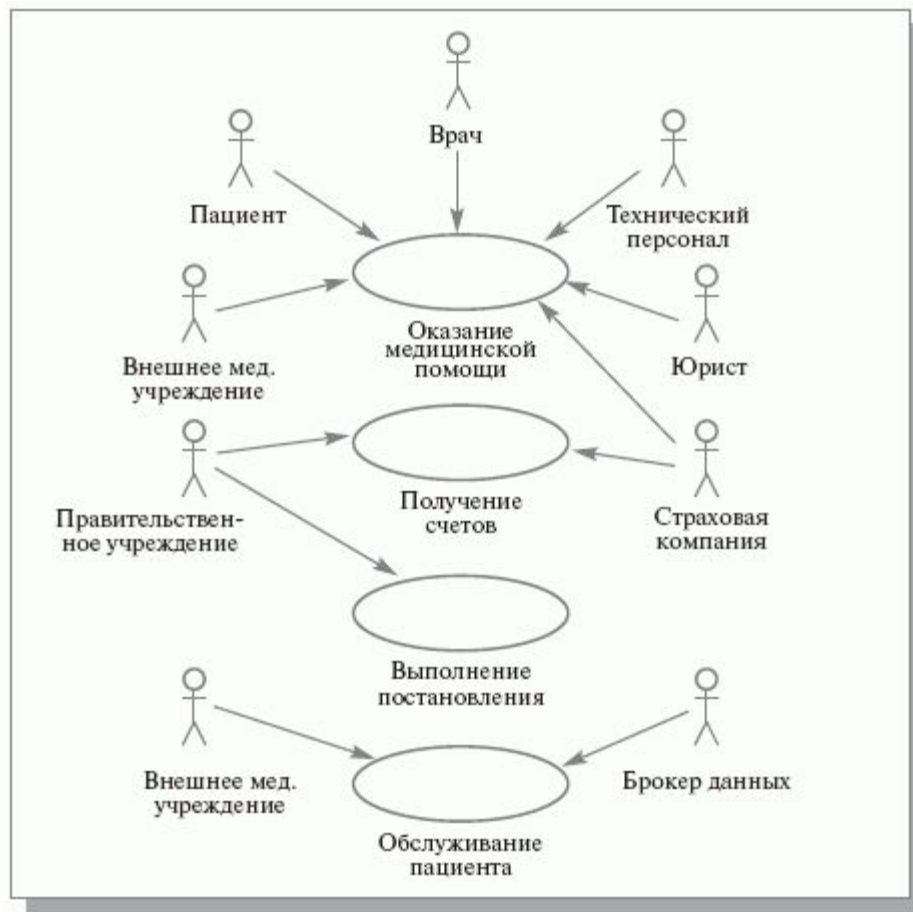
диаграмма прецедентов (Use case diagram) - диаграмма поведения, на которой показано множество прецедентов и актеров, а также отношения между ними

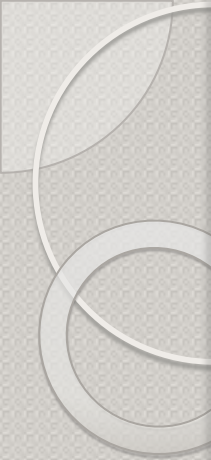


# общая модель деятельности центра в виде диаграммы прецедентов



# Модель бизнес-прецедентов, составляющих обслуживание пациента





Для включения в диаграмму выбранные прецеденты должны удовлетворять следующим критериям:

- прецедент должен описывать, **ЧТО** нужно делать, а не **КАК**;
- прецедент должен описывать действия с точки зрения **ИСПОЛНИТЕЛЯ**;
- прецедент должен возвращать исполнителю некоторое **СООБЩЕНИЕ**;
- последовательность действий внутри прецедента должна представлять собой одну **НЕДЕЛИМУЮ** цепочку.