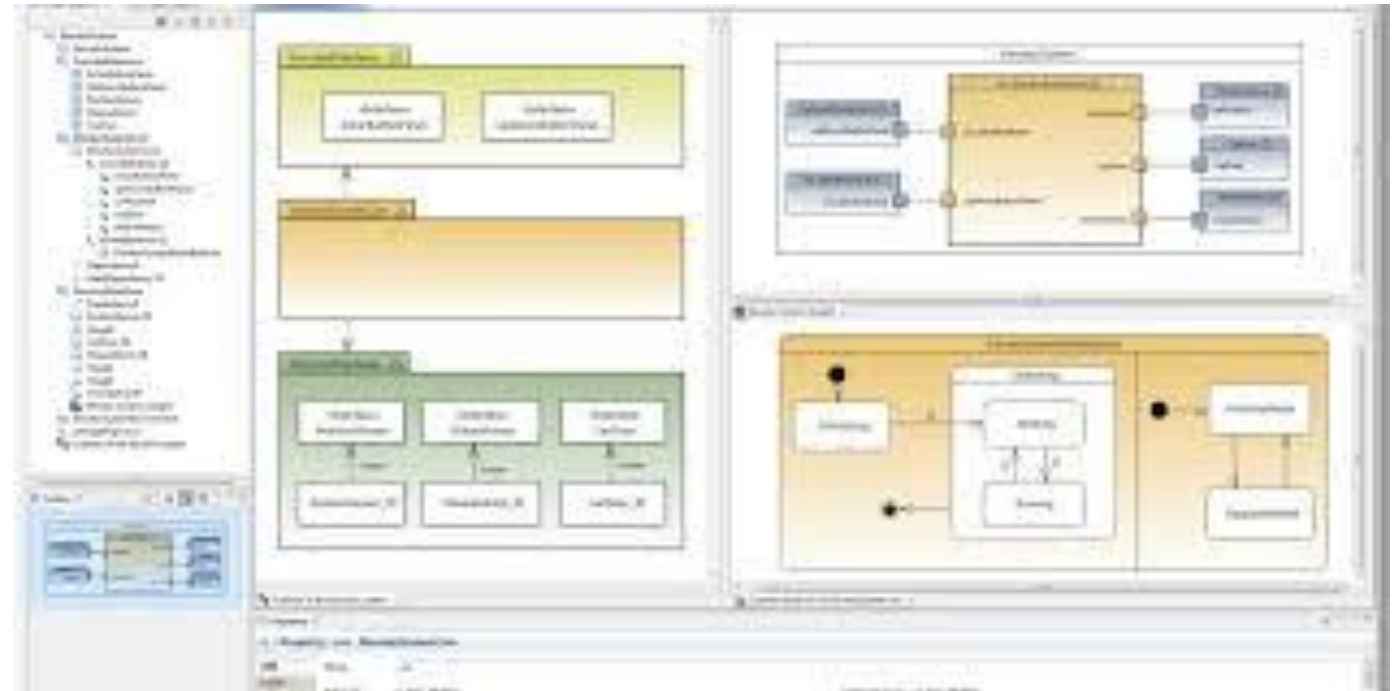


Концептуальная модель UML.

UML (англ. Unified Modeling Language — унифицированный язык моделирования)



Краткая история создания UML.

Создание UML фактически началось в конце 1994 г., когда **Гради Буч** и **Джеймс Рамбо** начали работу по объединению методов Booch и OMT (Object Modeling Technique) под эгидой компании Rational Software. К концу 1995 г. они создали первую спецификацию объединенного метода, названного ими Unified Method, версия 0.8. Тогда же, в 1995 г., к ним присоединился создатель метода OOSE (Object-Oriented Software Engineering) **Ивар Якобсон**.

Таким образом, UML является прямым объединением и унификацией методов Буча, Рамбо и Якобсона, однако дополняет ИХ НОВЫМИ ВОЗМОЖНОСТЯМИ

Унифицированный язык моделирования UML (Unified Modeling Language) - это «язык для спецификации, визуализации, проектирования, документирования элементов информационных систем».

UML пригоден для моделирования любых систем: от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени.

Конструктивное использование языка UML основывается на понимании общих принципов моделирования сложных систем и особенностей процесса объектно-ориентированного анализа и проектирования (ООАП).

Концептуальная модель UML включает:

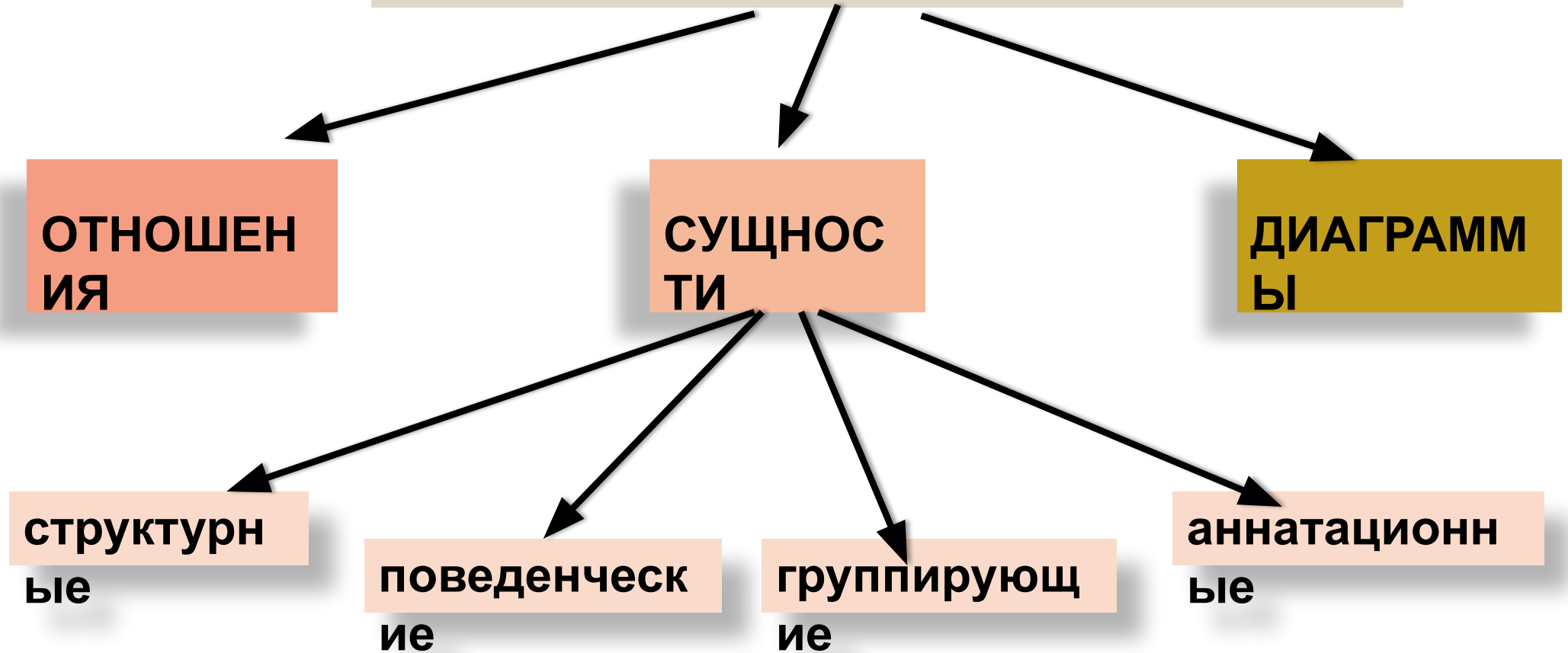
- основные строительные блоки языка;
- правила их сочетания;
- некоторые общие для всего языка механизмы.



2.

Словарь

Словарь языка UML включает три вида строительных блоков:



Структурные сущности – это имена существительные в моделях на языке UML. Они представляют собой статические части модели, соответствующие концептуальным или физическим элементам системы.

Класс (Class) – это описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. Класс реализует один или несколько интерфейсов.



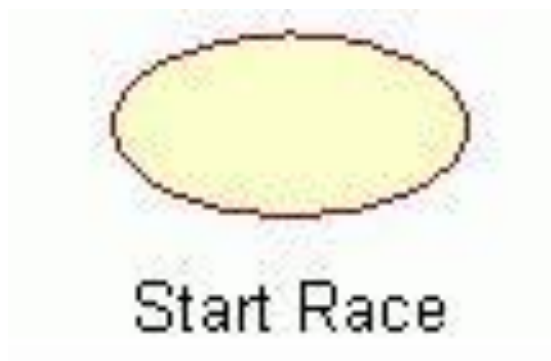
Интерфейс (Interface) – это совокупность операций, которые определяют сервис (набор услуг), предоставляемый классом или компонентом. Интерфейс описывает видимое извне **поведение элемента**. Интерфейс может представлять поведение класса или компонента полностью или частично; он определяет только спецификации операций (сигнатуры), но никогда – их реализации.



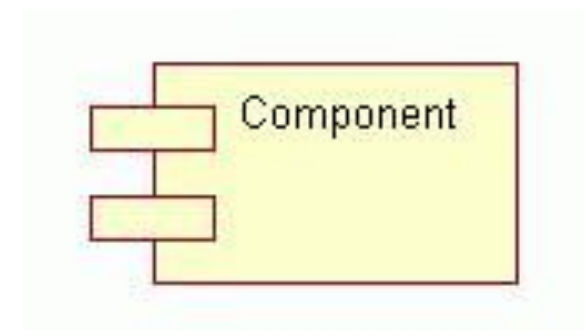
Кооперация (Collaboration) определяет взаимодействие; она представляет собой совокупность ролей и других элементов, которые, работая совместно, производят некоторый кооперативный эффект, не сводящийся к простой сумме слагаемых. Один и тот же класс может принимать участие в нескольких кооперациях; таким образом, они **являются реализацией образцов поведения**, формирующих систему.



Прецедент (Use case) – это описание последовательности выполняемых системой действий, которая производит наблюдаемый результат, значимый для какого-то определенного **актера (Actor)**. Прецедент применяется для **структурирования поведенческих сущностей модели.**



Компонент (Component) – это физическая заменяемая часть системы, которая соответствует некоторому набору интерфейсов и обеспечивает его реализацию (например файлы исходного кода). Компонент, как правило, представляет собой физическую упаковку логических элементов, таких как классы, интерфейсы и кооперации.



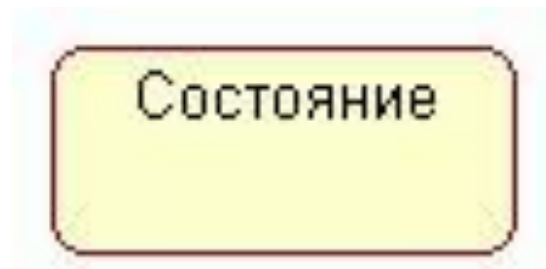
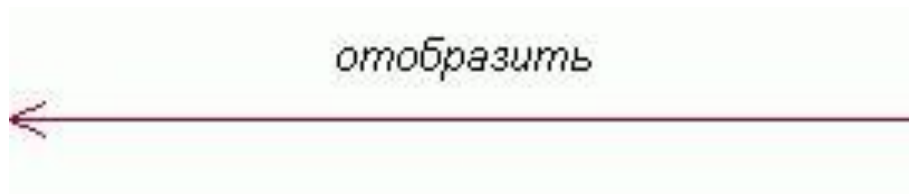
Узел (Node) — это элемент, представляющий вычислительный ресурс, обладающий памятью и способностью обработки. Совокупность компонентов может размещаться в узле, а также мигрировать с одного узла на другой.



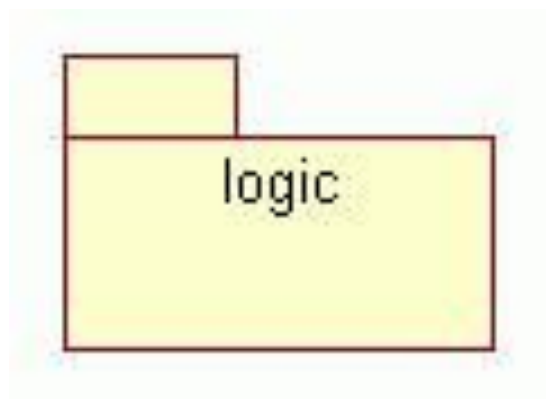
Поведенческие сущности (Behavioral things) являются динамическими составляющими модели UML. Это глаголы языка: они описывают поведение модели во времени и пространстве. Существует всего два основных типа поведенческих сущностей.

Взаимодействие (Interaction) – это поведение, суть которого заключается в обмене **сообщениями** между объектами в рамках конкретного контекста для достижения определенной цели.

Автомат (State machine) – это алгоритм поведения, определяющий **последовательность состояний**, через которые объект или взаимодействие проходят на протяжении своего жизненного цикла в ответ на различные события, а также реакции на эти события.



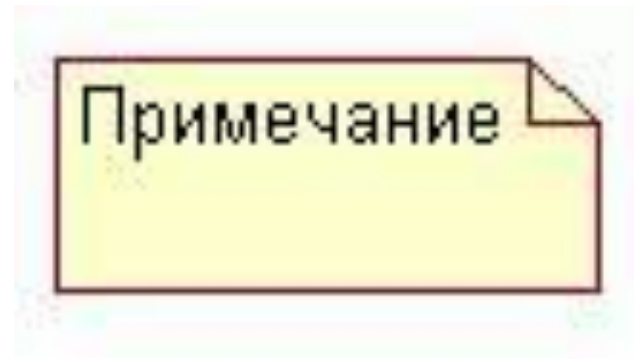
Группирующие сущности являются организующими частями модели UML. Это блоки, на которые можно разложить модель. Есть только одна первичная группирующая сущность, а именно пакет.



Пакеты (Packages) представляют собой универсальный механизм организации элементов в группы. В пакет можно поместить структурные, поведенческие и даже другие группирующие сущности. В отличие от компонентов, существующих во время работы программы, пакеты носят чисто концептуальный характер, то есть существуют только во время разработки.

Аннотационные сущности – пояснительные части модели UML. Это комментарии для дополнительного описания, разъяснения или замечания к любому элементу модели. Имеется только один базовый тип аннотационных элементов – **примечание** (Note).

Примечание – это просто символ для изображения комментариев или ограничений, присоединенных к элементу или группе элементов.



3.

Отношения

В языке UML определены четыре типа отношений:

- зависимость;
- ассоциация;
- обобщение;
- реализация.

Зависимость (Dependency) – это семантическое отношение между двумя сущностями, при котором изменение одной из них, независимой, может повлиять на семантику другой, зависимой.



Ассоциация (Association) – структурное отношение, описывающее совокупность связей; связь – это соединение между объектами. Разновидностью ассоциации является агрегирование (Aggregation) – так называют структурное отношение между целым и его частями.



Обобщение (Generalization) – это отношение «специализация/обобщение», при котором объект специализированного элемента (потомок) может быть подставлен вместо объекта обобщенного элемента (родителя или предка). Таким образом, потомок (Child) наследует структуру и поведение своего родителя (Parent).



Реализация (Realization) – это семантическое отношение между классификаторами, при котором один классификатор определяет «контракт», а другой гарантирует его выполнение.

Отношения реализации встречаются в двух случаях: во-первых, между интерфейсами и реализующими их классами или компонентами, а во-вторых, между прецедентами и реализующими их кооперациями.



Диаграмма в UML – это графическое представление набора элементов, изображаемое в виде связанного графа с вершинами (сущностями) и ребрами (отношениями), используемое для визуализации системы с разных точ



Правила языка UML (семантические)

Name – правила задания имен сущностей, отношений и диаграмм;

Scope – определения области существования (контекст, в котором имя имеет некоторое значение);

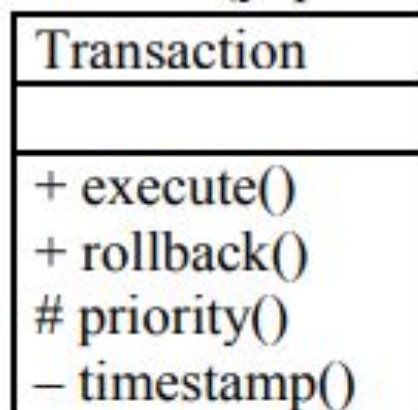
Visibility – правила определения области видимости имен;

Integrity – обеспечения целостности (как элементы должны правильно и согласованно соотноситься друг с другом);

Execution – правила выполнения (что значит выполнить или имитировать некоторую динамическую модель).

Общие механизмы UML.

- 1) спецификации (Specifications);
- 2) дополнения (украшения – adornments);



- 3) принятые деления (common divisions);
Класс | Объект Интерфейс | Реализация
- 4) механизмы расширения (extensibility mechanisms)
 - стереотипы (Stereotype);
 - помеченные значения (Tagged value);
 - ограничения (Constraints).