МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра вычислительная техника

**ЭССЕ по дисциплине «Автоматизация проектирования информационных систем»**

**Глава №2 «МОДЕЛИРОВАНИЕ ДАННЫХ С ПОМОЩЬЮ ДИАГРАММЫ КЛАССОВ UML»**

Отчёт составил:  
Бакалавр гр. ИВТАСбд-42  
Сулейманов М.З.

Отчёт принял:  
Профессор кафедры ВТ  
Токмаков Г.П.

Ульяновск

УлГТУ

2024

2.1 ВВЕДЕНИЕ

Унифицированный язык моделирования UML был создан в 1994 г. специалистами по программной инженерии Гради Бучем (Grady Booch) и Джеймсом Рамбо (James Rumbaugh) из компании Rational Software. С его помощью удается построить абстракцию, модель системы, не привязанную к конкретному языку программирования. Для этого в языке UML имеется набор понятий и концепций, который доступен специалисту, не знакомому с программированием. Язык UML активно применяется для проектирования реляционных БД. С точки зрения проектирования реляционных БД модельные возможности не слишком отличаются от возможностей ER диаграмм.

2.2 ОСНОВНЫЕ ПОНЯТИЯ ДИАГРАММ КЛАССОВ UML

Язык UML базируется на трех фундаментальных понятиях: сущность, отношение и диаграмма.

Сущность – это объект проектируемой системы, который целостно представляется в абстрактном виде.

Отношение – это способ и форму связи между сущностями.

Диаграмма – это визуальное представление набора сущностей с отношениями между ними.

В результате диаграммы составляются из элементов четырех типов:

* двумерные графические объекты представлены простыми геометрическими фигурами. Такие фигуры изображают сущности;
* пути – это отрезки, соединяющие графические объекты. С помощью путей, например, демонстрируется отношения между классами, схема наследования между сущностями;
* символы (значки) – это графические маркеры, размер и форма которых не меняются. Чаще всего в виде символов представлены стрелки на концах путей; текстовые надписи связаны по смыслу с определенной сущностью.

Диаграмма классов – один из наиболее важных и распространенных среди программистов и разработчиков БД видов диаграмм UML. Диаграммой классов в терминологии UML называется диаграмма, на которой показан набор классов (и некоторых других сущностей, не имеющих явного отношения к проектированию БД), а также связей между этими классами.

2.2.1 КЛАССЫ

Класс в языке UML представляет собой структуру, шаблон абстрактного понятия. Это именованное описание совокупности объектов с общими атрибутами, операциями, связями и семантикой. Графически класс изображается в виде прямоугольника. У каждого класса должно быть имя, уникально отличающее его от всех других классов.

2.2.2 АТРИБУТЫ

Атрибутом класса называется именованное свойство класса, описывающее множество значений, которые могут принимать экземпляры этого свойства. Атрибут является абстракцией состояния объекта. Любой атрибут любого объекта класса должен иметь некоторое значение.

2.2.3 ОПЕРАЦИИ

Операцией класса называется именованная услуга, которую можно запросить у любого объекта этого класса. Операция – это абстракция того, что можно делать с объектом.

2.3 ЗАВИСИМОСТИ, ОБОБЩЕНИЯ И АССОЦИАЦИИ

Объекты (сущности) диаграммы связываются друг с другом разными отношениями или связями. В диаграмме классов могут участвовать связи трех разных категорий: ассоциация (association), обобщение (generalization) и зависимость (dependency).

2.3.1 ЗАВИСИМОСТИ

Зависимостью называют связь по применению, когда изменение в спецификации одного класса может повлиять на поведение другого класса, использующего первый класс.

Зависимость показывается прерывистой линией со стрелкой, направленной к классу, от которого имеется зависимость.

Связи-зависимости существенны для объектно-ориентированных систем, но при проектировании реляционных БД зависимости не используются.

2.3.2 ОБОБЩЕНИЯ И МЕХАНИЗМ НАСЛЕДОВАНИЯ КЛАССОВ

Связью-обобщением называется связь между общей сущностью, называемой суперклассом, или родителем, и более специализированной разновидностью этой сущности, называемой подклассом, или потомком. Класс потомок наследует все атрибуты и операции класса-предка, но в нем могут быть определены дополнительные атрибуты и операции. Объекты класса-потомка могут использоваться везде, где могут использоваться объекты класса-предка. Это свойство называют полиморфизмом. Графически обобщения изображаются в виде сплошной линии с большой незакрашенной стрелкой, направленной к суперклассу.

2.3.3 АССОЦИАЦИИ: РОЛИ, КРАТНОСТЬ, АГРЕГАЦИЯ, КОМПОЗИЦИЯ

Ассоциацией называется структурная связь, показывающая, что объекты одного класса некоторым образом связаны с объектами другого или того же самого класса. Графически ассоциация изображается в виде линии, соединяющей класс сам с собой или с другими классами.

С понятием ассоциации связаны четыре важных дополнительных понятия: имя, роль, кратность и агрегация.

Имя ассоциации. Во-первых, ассоциации может быть присвоено имя, характеризующее природу связи.

Другим способом именования ассоциации является указание роли каждого класса, участвующего в этой ассоциации. Роль класса, как и имя конца связи в ER-модели, задается именем, помещаемым под линией ассоциации ближе к данному классу.

Кратностью (multiplicity) роли ассоциации называется характеристика, указывающая, сколько объектов класса с данной ролью может или должно участвовать в каждом экземпляре ассоциации.

«1» - каждый объект класса с этой ролью должен участвовать в некотором экземпляре данной ассоциации, в каждом экземпляре ассоциации может участвовать только в один объект класса.

«0..1» - каждый объект класса с этой ролью может (а может и не участвовать) участвовать в некотором экземпляре данной ассоциации, в каждом экземпляре ассоциации может участвовать только в один объект класса.

«1..\*» - каждый объект класса с этой ролью должен участвовать в некотором экземпляре данной ассоциации, в каждом экземпляре ассоциации может участвовать несколько объектов класса.

«0..\*» - каждый объект класса с этой ролью может (а может и не участвовать) участвовать в некотором экземпляре данной ассоциации, в каждом экземпляре ассоциации может участвовать несколько объектов класса.

Ассоциация рода «часть-целое» называется агрегатной. Изображаются в виде простой ассоциации с незакрашенным ромбом на стороне класса. Объекты класса могут существовать независимо друг от друга.

Ассоциации называются композитными, если объекты класса «части» не могут существовать без «целого».

Ассоциация с указанным направлением имеет ограничение в навигации: из одного объекта класса можно найти связанный с ним объект, но не в обратном направлении.

2.4 ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ И ЯЗЫК OCL

Один из самых серьезных и справедливо критикуемых недостатков языка моделирования UML является предоставление только средств визуального представления моделей, т. е. UML – это просто графический редактор диаграмм различного вида. В результате проектировщик не находит достаточно выразительных средств, позволяющих уточнить ограничения на применение создаваемых структур, специфицировать способы и нюансы их внутреннего функционирования, ввести условия использования и т. д.

Поэтому со временем на основе этого языка был создан язык объектных ограничений OCL, который применялся в то время как язык моделирования. Он активно применялся для описания особенностей информационных систем и в 1997 году вошел в стандарт языка UML.

2.4.2 ОБЩАЯ ХАРАКТЕРИСТИКА ЯЗЫКА OCL

В дополнение к скалярным типам данных, заимствованным из UML, в OCL предопределены структурные типы, которые являются разновидностями типов коллекций (collection):

* математическое множество (set), неупорядоченная коллекция, не содержащая одинаковых элементов;
* мультимножество (bag), неупорядоченная коллекция, которая может содержать повторяющиеся элементы-дубликаты;
* последовательность (sequence), упорядоченная коллекция, которая может содержать элементы-дубликаты.

Язык OCL предназначен, главным образом, для определения ограничений целостности данных, соответствующих модели, которая представлена в терминах диаграммы классов UML.

2.4.3 ИНВАРИАНТ КЛАССА

Под инвариантом класса в OCL понимается условие, которому должны удовлетворять все объекты данного класса. Его вычисление должно давать true в течении всего времени существования объекта.

context <class\_name> inv:

<OCL-выражение>.

inv определяет, что это именно инвариант.

сontext определяет, что мы работаем с объектами класса <class\_name>.

В общем случае OCL-выражение в определении инварианта формируется как композиция операций, которым посвящена большая часть определения языка.

Операции над значениями предопределенных типов данных.

В OCL поддерживаются следующие заимствованные из определения UML скалярные типы данных: Boolean, Integer, Real и String.

Операции над объектами.

В OCL определены три операции над объектами:

* получение значения атрибута;
* переход по соединению,
* вызов операции класса (последняя операция для целей проектирования реляционных БД несущественна)

2.4.4 ОПЕРАЦИИ НАД МНОЖЕСТВАМИ, МУЛЬТИМНОЖЕСТВАМИ И ПОСЛЕДОВАТЕЛЬНОСТЯМИ

Общий синтаксис применения операции к коллекции следующий: <коллекция> <имя операции> (<список фактических параметров>).

Операция select

Результатом операции является новое множество, мультимножество или последовательность. При этом множество, мультимножество или последовательность формируется из тех элементов входной коллекции, для которых результатом вычисления логического выражения является true.

Операция collect

Результатом операции является новое множество, мультимножество или последовательность. При этом результирующая коллекция соответствующего типа (коллекция значений или объектов) состоит из результатов применения выражения к каждому элементу входной коллекции.

Операции exists, forAll, size

Для exists в результате каждой из этих операций выдается true в том и только в том случае, когда хотя бы для одного элемента входной коллекции значением логического выражения является true. В противном случае результатом операции является false.

Операции forAll отличаются от операций exist тем, что в результате каждой из них выдается true в том и только в том случае, когда для всех элементов входной коллекции результатом вычисления логического выражения является true. В противном случае результатом операции будет false.

Операция size применяется к коллекции и выдает число содержащихся в ней элементов.

Операция union

Результатом операции union, определенной над множеством и мультимножеством, является мультимножество, т. е. из результата объединения таких двух коллекций дубликаты не исключаются.

Результатом же операции union, определенной над двумя множествами, является множество, т. е. в этом случае возможные дубликаты должны быть исключены.

2.5 ПОЛУЧЕНИЕ СХЕМЫ РЕЛЯЦИОННОЙ БД ИЗ ДИАГРАММЫ КЛАССОВ UML

Рекомендация 1. Прежде чем определять в классах операции, подумайте, что вы будете делать с этими определениями в среде целевой реляционной СУБД.

Рекомендация 2. В реляционной СУБД сравнительно эффективно реализуются только ассоциации видов «один ко многим» и «многие ко многим».

Рекомендация 3. Для технологии реляционных БД композитные ассоциации неестественны.

Рекомендация 4. Не злоупотребляйте возможностями OCL.