Одесский Национальный Политехнический Университет

Кафедра информационных систем

Лабораторная работа № 3

по дисциплине: «Технологии компьютерного проектирования»

на тему «Диаграммы классов»

Выполнил:

Ст. группы АИ-166

Дидух Э. Г.

Проверил:

Галченков О. Н.

Одесса, 2018

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ:

Цель работы: научиться строить диаграммы классов

Диаграмма классов является частью логической модели системы и представляет статическую картину системы.

Для каждой системы строится общая диаграмма классов и несколько дополнительных диаграмм классов, показывающих кооперации отдельных классов для реализации отдельных прецедентов и сценариев. На одних показывают подмножества классов, объединенные в пакеты, и отношения между ними, на других – отображают те же подмножества, но с атрибутами и операциями классов. Для представления системы разрабатывается столько дополнительных диаграмм классов, сколько потребуется.

Основные элементы диаграмм классов

Дадим некоторые определения и опишем основные элементы нотации диаграмм классов.

Объект – это некоторая сущность реального мира или концептуальная (абстрактная) сущность.

Состояние объекта – это одно из условий, в котором он может находиться. Состояние обычно изменяется со временем и характеризуется набором свойств, которые называются атрибутами. Поведение определяет, как объект реагирует на запросы других объектов и что может делать сам объект. Поведение характеризуется операциями объекта. Индивидуальность означает, что каждый объект уникален, даже если его состояние идентично состоянию другого объекта. Класс – это описание группы объектов с общими свойствами (атрибутами), поведением (операциями), отношениями с другими объектами и семантикой.

Каждый класс является шаблоном для создания объекта. А каждый объект – это экземпляр класса.В нотации UML классы и объекты изображаются в виде прямоугольниковПрямоугольник класса всегда делится на три секции (раздела), имя класса помещается в первую секцию, каждое слово в названии класса принято писать с большой буквы. Во второй и третьей секциях могут указываться атрибуты и операции класса соответственно, эти секции могут быть пустыми. Названия классов выбираются в соответствии с понятиями предметной области. Это должно быть существительное или словосочетание в единственном числе, наиболее точно характеризующее предмет. Класс должен описывать только одну сущность.Выявление классов можно начать с изучения потока событий. Имена существительные в описании этого потока дадут понять, что может являться классом. В общем случае существительное может оказаться действующим лицом, классом, атрибутом класса или выражением, не являющимся ни действующим лицом, ни классом, ни атрибутом класса. Некоторые возможные классы будут выявлены при рассмотрении трех стереотипов: сущность (entity), граница (boundary) и управление (control). Мы уже встречались со стереотипами отношений, когда говорили об отношениях на диаграммах прецедентов. Тот же принцип создания нового типа на основе уже существующего применим и для классов.

Стереотип – это механизм, позволяющий категоризировать классы. Он используется для создания нового типа элемента, в данном случае нового типа класса.

Например, Вы хотите выделить все экранные формы в модели. Для этого нужно создать стереотип Form (Форма).

Стереотипы помогают лучше понять ответственности каждого класса в модели, категоризировать выполняемые ими функции. В UML для этого применяют три основных стандартных вида стереотипов классов: классы-сущности, граничные классы и управляющие классы.

Класс-сущность содержит информацию, хранимую постоянно. Используется для моделирования данных и поведения с длинным жизненным циклом. Они могут представлять информацию о предметной области, а могут представлять элементы самой системы. Часто являясь абстракциями предметной области, они имеют наибольшее значение для пользователя, поэтому в их названиях применяются термины предметной области. Если существует проект базы данных, то можно обратиться к изучению названий таблиц, многие из них станут классами-сущностями. Обозначаются классы-сущности стереотипом <<entity>> либо специальной пиктограммой (рис. 35).Граничными классами называются классы, расположенные на границе системы со всем остальным миром, и т.о. они обеспечивают взаимодействие между окружающей средой и внутренними элементами системыУправляющий класс отвечает за координацию действий других классов. Они служат для моделирования последовательного поведения одного или нескольких прецедентов и координации событий, реализующих заложенное в них поведение. Обозначаются управляющие классы именем стереотипа <<control>> либо специальной пиктограммойВ StarUML документирование классов выполняется также как и описанное выше документирование прецедентов.Диаграммы классов относятся к логическому представлению системы Logical View. На диаграмме Main представления Logical View обычно размещают главную диаграмму пакетов, а диаграммы классов помещают на другие листы этого представления.Если в нашей модели немного классов, то нам легко ими управлять, однако многие системы содержат большое количество классов, поэтому необходим механизм, позволяющий классы группировать и облегчающий их повторное использование. Таким механизмом в UML являются пакеты.

Пакет (package) — общецелевой механизм для организации различных элементов модели в группы.

Подпакет (subpackage) — пакет, который является составной частью другого пакета.

Пакет в логическом представлении модели – это объединение классов или других пакетов. С помощью объединения классов в пакеты мы можем получить представление о системе на более высоком уровне. Напротив, рассматривая пакет, мы получаем более детальное представление модели.

Объединять классы в пакеты можно как угодно, однако, существует несколько наиболее распространенных подходов.

1. можно группировать классы по стереотипам: классы-сущности, граничные и управляющие классы.

2. группировка классов по их функциональности: например, пакет классов, отвечающих за безопасность системы или пакет классов Работа с сотрудниками и т.п.

3. наконец, применяют комбинацию двух указанных методов.

В дальнейшем можно вкладывать пакеты друг в друга. Механизм инкапсуляции в UML реализуется за счет объединения свойств и поведения в одном объекте. Свойства объекта описываются с помощью задания атрибутов класса, к которому относится объект, а поведение – заданием операций класса.

На прямоугольнике класса атрибуты описываются во второй секции под именем, а операции – в третьей, под атрибутами.Атрибут (attribute) — содержательная характеристика класса, описывающая множество значений, которые могут принимать отдельные объекты этого класса. Операция (operation) - это сервис, предоставляемый каждым экземпляром или объектом класса по требованию своих клиентов, в качестве которых могут выступать другие объекты, в том числе и экземпляры данного класса. Видимость (visibility) — качественная характеристика описания свойств класса, характеризующая потенциальную возможность других объектов модели использовать это свойство (атрибут или операцию).

Видимость в языке UML обозначается с помощью квантора видимости (visibility), который может принимать одно из 4-х возможных значений и отображаться при помощи специальных символов.

* Открытый (public). Атрибут виден всем остальным классам. Любой класс, связанный с данным в рамках диаграммы или пакета, может просмотреть или изменить значение атрибута. Обозначается символом «+» перед именем атрибута.
* Защищенный (protected). Любой потомок данного класса может пользоваться его защищенными свойствами. Обозначается знаком «#» перед именем атрибута.
* Закрытый (private). Атрибут с этой областью видимости недоступен или не виден для всех классов без исключения. Обозначается знаком «-» перед именем атрибута.
* Пакетный (package). Атрибут является открытым, но только в пределах своего пакета. В StarUML данный атрибут обозначается значком «~». Кратность (multiplicity) — спецификация области значений допустимой мощности, которой могут обладать соответствующие множества.

Кратность указывает, сколько экземпляров данного атрибута может иметь экземпляр класса.Тип атрибута представляет собой выражение, семантика которого определяется некоторым типом данных, определенным в пакете «Типы данных языка UML» или самим разработчиком. Тип атрибута может определяется в зависимости от языка программирования, который предполагается использовать для реализации данной модели.Имя операции совместно с ее параметрами называют сигнатурой операции. Имя операции – это строка текста, обычно используют глагол или короткое глагольное выражение, если оно состоит из нескольких слов, то все слова, кроме первого, пишутся с большой буквы:

добавить или добавитьТоварВКорзину.

Так же как и для атрибутов класса, видимость (visibility) операции обозначается с помощью квантора видимости и имеет четыре допустимых значения.

Для того чтобы обнаружить связи классов исследуются сценарии и диаграммы последовательности: если объект посылает сообщение другому объекту, то по-видимому, между ними существует отношение.

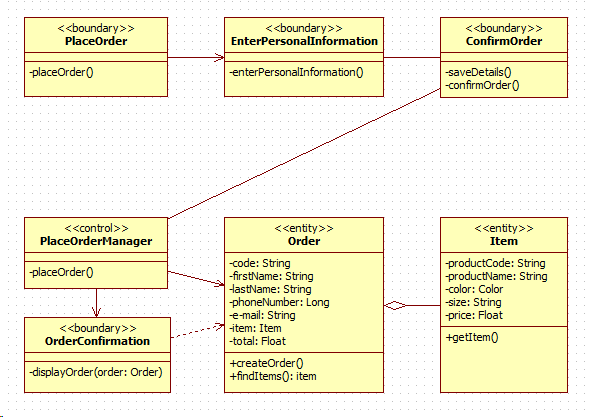
Отношения возникают также, если один класс использует другой в качестве параметра операции.

В нотации UML определены 5 видов отношений между классами:

* ассоциации;
* агрегации;
* композиции;
* зависимости;
* обобщения;

Ассоциация – это семантическая связь между классами. На диаграмме ее рисуют в виде обычной линии. Ассоциация может быть однонаправленной или двунаправленной. В первом случае ее изображают в виде стрелки, показывающей направление связи. Во втором случае – двойной стрелки либо просто линии без стрелок. Окончание линии ассоциации в месте, где она соединяется с классом, называется ролью ассоциации. Кратность (мощность) определяется для классов и указывает допустимое количество объектов (экземпляров класса), участвующих в отношении.

Кратность указывает, сколько экземпляров одного класса взаимодействуют с помощью отношения с одним экземпляром данного класса в данный момент. Агрегация - специальная форма ассоциации, которая служит для представления отношения типа "часть-целое" между агрегатом (целое) и его составной частью Композицией называется форма агрегирования с четко выраженным отношением владения, причем время жизни частей и целого совпадают Зависимостью называется отношение использования, определяющее, что изменение в спецификации одной сущности может повлиять на другую сущность, которая ее использует, причем обратное в общем случае не верно



Контрольные вопросы:

1. Каково назначение диаграммы классов?

Диграмма классов строится для показания кооперации отдельных классов для реализации отдельных прецедентов и сценариев.

2. Чем отличается агрегация от композиции?

Композиция — форма агрегации, при которой время жизни части и целого совпадают с четко выраженным отношением владения.

3. Какие бывают отношения между классами?

В нотации UML определены 5 видов отношений между классами:

* ассоциации;
* агрегации;
* композиции;
* зависимости;
* обобщения;

4. Какие бывают стереотипы классов?

Основные стереотипы классов — класс-сущность, граничный класс, управляющий класс.

5. Какие значения принимает “направление параметра”?

Направление параметра может принимать одно из нижеследующих значений:

* in – входящий параметр, который не может быть изменен;
* out – выходящий параметр, который может быть изменен, чтобы передать информацию вызвавшей процедуре;
* inout – входящий параметр, который может быть изменен.

6. Для чего используются пакеты?

Пакеты используются для организации различных элементов модели в группы.

7. Какие бывают отношения между пакетами?

Отношения между пакетами могут быть только отношениями типа зависимости.

8. Какие значения может принимать “видимость” атрибута и операции?

* Public
* Private
* Protected
* Package