

## Практическое занятие № 4.

### Универсальный язык моделирования UML (Unified Modelling Language). Пример (продолжение).

**Цель практического занятия:** изучить правила моделирования программного обеспечения с помощью UML. Построение диаграммы классов. Пример - "Проектирование фрагмента библиотечной поисковой системы". Формирование диаграммы классов.

#### План проведения занятия:

1. Ознакомиться с лекционным материалом по UML.
2. Ознакомиться с практическим занятием №4 по UML.
3. Дополнить диаграмму классов, отражающую модель предметной области, атрибутами.

#### Диаграмма классов.

*Диаграмма классов* - основной способ отображения структуры разрабатываемой системы, показывающий классы и отношения между ними. Диаграмма классов - это графическое представление статической модели, в которой собраны декларативные (статичные) элементы, такие как классы, типы, а также их содержимое и отношения. Диаграмма классов содержит некоторые конкретные элементы поведения (например, операции), однако их динамика отражается на других видах диаграмм.

Следует заметить, что диаграмма классов может также содержать интерфейсы, отношения, а также объекты и связи. Когда говорят о данной диаграмме, то имеют в виду статическую структурную модель проектируемой системы. Поэтому диаграмму классов принято считать графическим представлением таких структурных взаимосвязей логической модели системы, которые не зависят от времени.

#### Объекты

*Объект* - предмет или понятие из реального мира. Это дискретная сущность с четко определенными границами и индивидуальностью, инкапсулирующая состояние и поведение. Экземпляр класса.

Особенности объекта:

- имя объекта;
- состояние объекта. Состояние включает в себя имена свойств (атрибутов!) описывающих объект, и значения этих атрибутов в определенный момент времени;
- поведение объекта, определяющееся функциями (методами), которые используют или изменяют значения атрибутов объекта.

#### Классы

**Класс** - набор объектов, имеющих одинаковые характеристики (Обозначение класса см. рис. 5.1). Другими словами, является абстрактным описанием или представлением свойств множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Класс используют для объявления переменных. Это поименованное описание структуры данных и поведения некоего множества объектов. Класс описывает некоторую концепцию в моделируемой системе. В зависимости от типа модели класс может отражать концепцию реального мира (например, для аналитической модели) или содержать концепции, относящиеся к алгоритмической или компьютерной реализации (для модели проектирования).

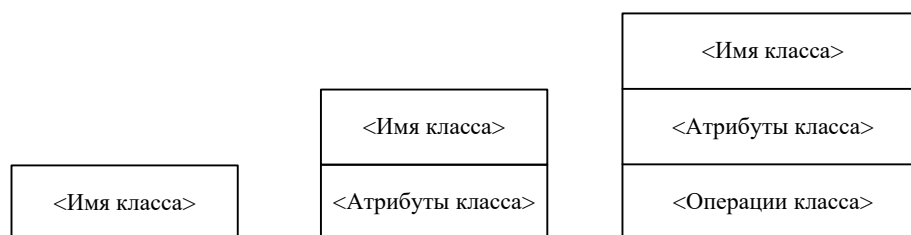


Рис. 5.1. Варианты обозначения класса.

#### **Особенности класса:**

- имя класса (имя объекта может включать в себя имя класса) – обязательный элемент;
- атрибуты каждого объекта класса;
- поведение в терминах операций, а не методов.

На начальных этапах разработки диаграммы отдельные классы могут обозначаться простым прямоугольником с указанием только имени соответствующего класса. По мере проработки отдельных компонентов диаграммы, описания классов дополняются атрибутами и операциями. Предполагается, что окончательный вариант диаграммы содержит наиболее полное описание классов, которые состоят из трех разделов или секций.

Имя класса – должно быть уникальным, указывается в самой верхней секции прямоугольника, записывается по центру секции имени полужирным шрифтом и должно начинаться с заглавной буквы, должны использоваться существительные без пробелов. Именно имена классов образуют словарь предметной области. Класс, не имеющий экземпляров или объектов, называется абстрактным классом, а для обозначения его имени используется наклонный шрифт.

Атрибут класса – служит для представления отдельного свойства или признака, который является общим для всех объектов данного класса. Каждый атрибут – отдельная

строка, записывается во второй сверху секции прямоугольника класса, принята определенная стандартизация записи, которая подчинена синтаксическим правилам. Из всех элементов обязательно указывать имя атрибута (идентификатор соответствующего атрибута) – должно быть уникальным в пределах данного класса, должно начинаться со строчной буквы и не должно содержать пробелов, используют существительные.

Операция класса – это некоторый сервис, который предоставляет каждый экземпляр или объект класса по требованию своих клиентов (других объектов, в том числе и экземпляров данного класса). Совокупность операций характеризует функциональный аспект поведения всех объектов данного класса. Каждая операция – отдельная строка, записывается в третьей сверху секции прямоугольника, принята определенная стандартизация записи, которая подчинена синтаксическим правилам. Из всех элементов обязательно указывать имя операции (идентификатор операции) – должно быть уникальным в пределах данного класса, должно начинаться со строчной буквы и не должно содержать пробелов, используют глагол или глагольный оборот.

**Операция** - служба, которая может быть запрошена объектом для изменения поведения. Или это абстракция того, что позволено делать с объектом.

**Метод** - реализация этой службы.

**Экземпляр класса** - объект, принадлежащий определенному классу.

### Отношения между классами

1. **Ассоциация** - структурная связь между классами (рис. 5.2). Соответствует наличию произвольного отношения или взаимосвязи между классами. Ассоциация двунаправлена - можно перемещаться от одного класса к другому и наоборот. Направление связи можно указать, используя стрелку (рис. 5.3). В качестве дополнительных специальных символов могут использоваться имя ассоциации, символ навигации, а также имена и кратность классов-ролей ассоциации (рис. 5.4).

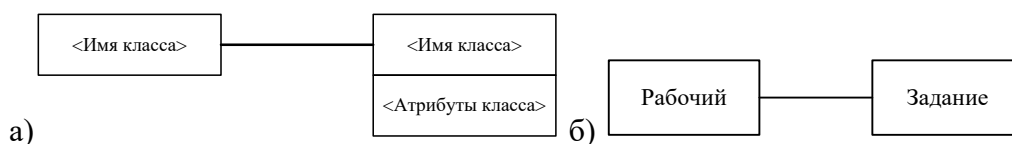


Рис. 5.2. Отношение ассоциации: а) обозначение двунаправленной связи, б) пример.

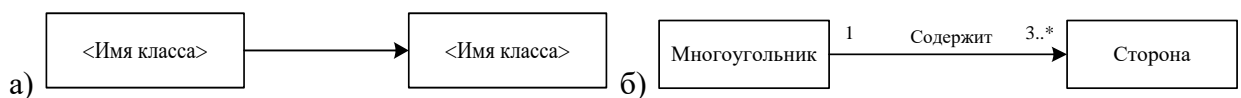


Рис. 5.3. Отношение ассоциации: а) обозначение однонаправленной связи, б) пример.

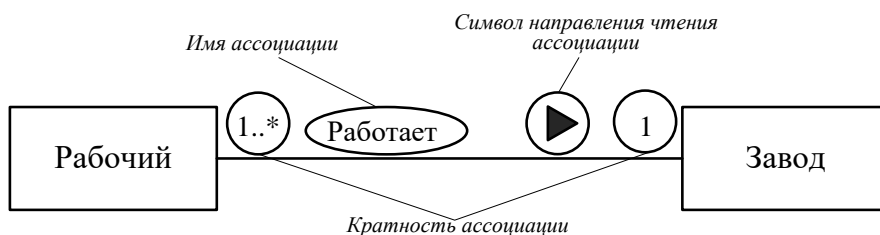


Рис. 5.4. Применение специальных символов.

2. **Обобщение** - отношение между более общим классом (суперклассом или родительским классом) и уточняющей версией этого класса (подклассом или потомком) (рис. 5.5). Подкласс наследует атрибуты и операции своего суперкласса. Может использоваться для представления иерархических взаимосвязей между классами и т.д.

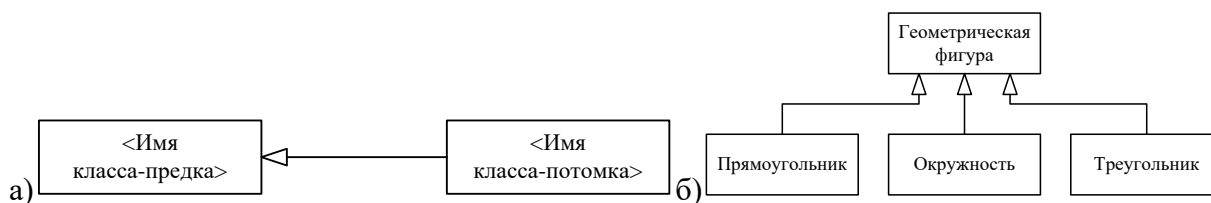


Рис. 5.5. Отношение обобщения: а) обозначение, б) пример.

3. **Агрегация** - специальный вид ассоциации, обозначающий отношение "целое/часть", при котором один или несколько классов являются "частями" одного "целого" (рис. 5.6). Имеет место между несколькими классами в том случае, если один из классов представляет собой некоторую сущность, которая включает в себя в качестве составных частей другие сущности. Раскрывая внутреннюю структуру системы отношение агрегации показывает, из каких элементов состоит система и как они связаны между собой. Порождаемая здесь иерархия принципиально отличается от порождаемой отношением обобщения – здесь части системы никак не обязаны наследовать ее свойства и поведение, поскольку являются вполне самостоятельными сущностями. Может указываться кратность.

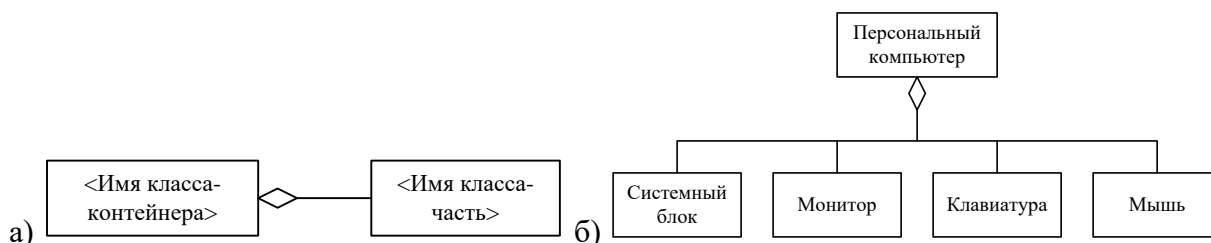


Рис. 5.6. Отношение агрегации: а) обозначение, б) пример.

4. **Композиция** – частный случай отношения агрегации. Служит для спецификации более сильной формы отношения «часть-целое», при которой составляющие части тесно

взаимосвязаны с целым (рис. 5.7). Специфика заключается в том, что части не могут выступать в отрыве от целого, то есть с уничтожением целого уничтожаются и все его составные части. Может указываться кратность.

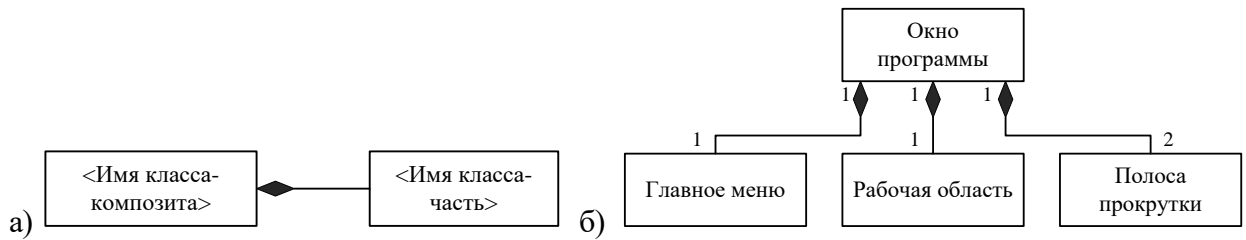


Рис. 5.7. Отношение композиции: а) обозначение, б) пример.

5. **Зависимость** – в общем случае указывает на некоторое семантическое отношение между двумя элементами модели или двумя множествами таких элементов, которые не являются отношением ассоциации, обобщения, агрегации или композиции. Оно касается только самих элементов модели. Используется в такой ситуации, когда некоторое изменение одного элемента модели может потребовать изменения другого зависящего от него элемента модели. (рис. 5.8).

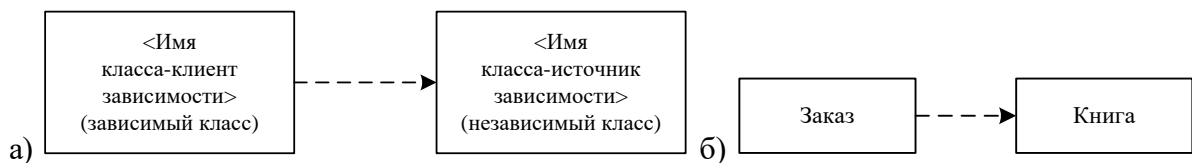


Рис. 5.8. Отношение зависимости: а) обозначение, б) пример.

Стрелка зависимости может помечаться необязательными, но стандартными ключевыми словами в кавычках, например: «access» (доступ), «bind» (связывание), «call» (вызов), «derive» (вызов), «friend» (дружественность), «import» (импорт), «instantiate» (создание экземпляра), «parameter» (параметр), «realize» (реализация), «refine» (уточнение), «send» (отправка), «trace» (трассировка), «use» (использование).

### Интерфейсы

Интерфейс является специальным случаем класса, у которого имеются только операции и отсутствуют атрибуты (рис. 5.9). Применительно к диаграммам классов, интерфейсы определяют совокупность операций, который обеспечивают необходимый набор сервисов или функциональности для актеров. Интерфейсы не могут содержать ни атрибутов, ни состояний, ни направленных ассоциаций. Они содержат только операции без указания особенностей их реализации.

Важность интерфейсов заключается в том, что они определяют стыковочные узлы в проектируемой системе, что совершенно необходимо для организации коллективной работы над проектом.

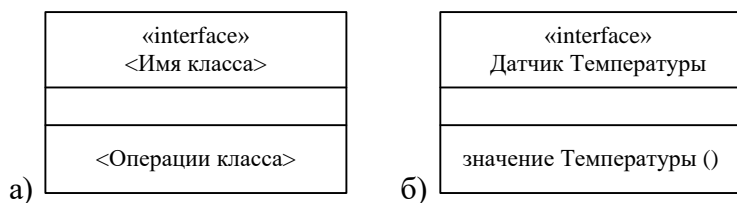


Рис. 5.9. Интерфейс: а) обозначение, б) пример.

## Элементы, связи, ограничения, манипуляции UML

### Элементы и связи

Элементы и связи между элементами UML:

- диаграмм классов - таблица 4.2.

Табл. 4.2

Обозначение элемента	Название элемента		Что отражает
	Класс		Объект предметной области
	Класс с атрибутами		Объект предметной области с определением его состояния
	Класс с атрибутами и операциями		Объект предметной области с определением его состояния и поведения
	двунаправленная ассоциация	Связь между объектами предметной области (отношение)	Структурная связь между классами
	однонаправленная ассоциация		Структурная связь между классами
	обобщение		Отношение «класс-подкласс»
	агрегация		Отношение «целое-часть»
	композиция		Отношение «целое-часть»

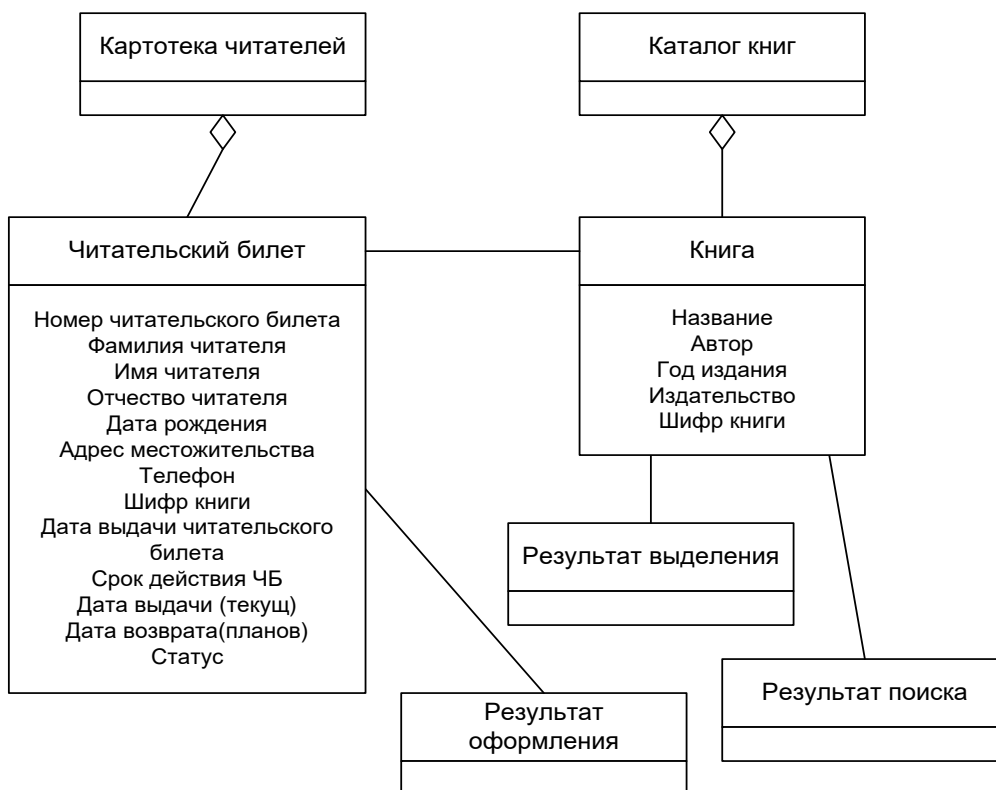
----->	зависимость		Семантическое отношение между элементами модели
--------	-------------	--	--

### Ограничения и манипуляции

1. Необходимо дать диаграмме имя, связанное с ее назначением.
2. На диаграмме необходимо располагать элементы так, чтобы свести к минимуму число пересекающихся линий.
3. Элементы должны быть пространственно организованы так, чтобы семантически близкие сущности располагались рядом.
4. Для привлечения внимания к важным особенностям диаграммы можно использовать примечания и цвет.
5. Необходимо стараться не показывать слишком много разных видов отношений, как правило, в каждой диаграмме классов должны доминировать отношения какого-либо одного вида.

### Диаграмма классов - модель предметной области с атрибутами.

На данном этапе описания классов, выделенных в модели предметной области, дополняются атрибутами. Для каждого класса формируется свой набор атрибутов, описывающих свойства данного класса. Также существует возможность дополнить диаграмму не учтенными ранее классами.



**Список контрольных вопросов:**

1. Дайте определение UML.
2. Назовите базовые элементы, используемые в диаграмме классов.
3. Сформулируйте правила формирования диаграммы классов.
4. Какие отношения применяют для установления связей между классами.
5. Приведите определение для отношения «агрегация».
6. Приведите определение для отношения «обобщение».
7. Приведите определение для отношений «агрегация» и «композиция».