

Министерство транспорта Российской Федерации
Федеральное агентство железнодорожного транспорта
Омский государственный университет путей сообщения

Институт автоматизации, телекоммуникаций и
информационных технологий (ИАТИТ)

Кафедра «Высшая математика»

СОЗДАНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ МОДЕЛЕЙ БИЗНЕС-
ПРОЦЕССОВ НА ОСНОВЕ ШАБЛОНА «СХЕМА МОДЕЛИ UML»
MS OFFICE VISIO 2007

Часть 1

Построение диаграммы классов на основе шаблона «Схема модели UML»
MS Office Visio 2007

Омск 2014

УДК 004.9(075.8)

ББК 32.97я73

К84

Создание объектно-ориентированных моделей бизнес-процессов на основе шаблона «Схема модели UML» MS Office Visio 2007. Часть 1: Методические указания для индивидуальной самостоятельной работы студентов, обучающихся по направлению подготовки 080700 «Бизнес–информатика» очной формы обучения /Т.Ю.Круковская; Омский гос.ун-т путей сообщения. Омск, 2011. 36 с.

Соответствуют действующей программе курса для технических вузов, содержат базовые теоретические сведения о построении с помощью шаблона «Схема модели UML» диаграммы классов, включающего объекты классов, их атрибуты, операции, а также отношения между объектами классов.

Библиогр.: 6 назв., рис.45.

Рецензент:

Шарова А.Н. кандидат педагогических наук, доцент кафедры информатики и вычислительной техники Омского государственного педагогического университета

Оглавление

ВВЕДЕНИЕ	4
Построение диаграммы классов на основе шаблона Схема модели UML MS Office Visio 2007	5
1. Основные теоретические сведения.....	5
1.1. Классы.....	5
1.2. Атрибуты классов.....	11
1.3. Операции классов	13
1.4. Отношения между классами.....	15
1.4.1.Отношение зависимости	15
1.4.2.Отношение ассоциации	16
1.4.3.Отношение агрегации.....	19
1.4.4.Отношение композиции	20
1.4.5.Отношение обобщения.....	21
2. Практические задания	24
Список литературы и ссылки на интернет–ресурсы.....	35

ВВЕДЕНИЕ

Предлагаемые методические указания предназначены для приобретения практических навыков работы с инструментальной средой MS Visio 2007 в приложении к задачам моделирования системы процессов предметной области на основе методов объектно-ориентированного анализа, моделирования и проектирования языка графического описания Unified Modeling Language (UML).

Язык визуализации UML, прежде всего, предназначен для моделирования и разработки программных систем, однако его выразительность позволяет описывать деятельность непрограммных систем, в частности описывать модель организации, включающую описание деловых объектов: подразделений, должностей, ресурсов, операций и пр. с указанием связей между ними.

Методические указания содержат базовые теоретические сведения о создании диаграмм классов, используемых для визуализации, специфицирования, документирования структурных моделей статического представления системы процессов, показан обзор классов, интерфейсов, атрибутов, операций объектов классов, приведена классификация отношений и их краткое описание.

Практическая часть методических указаний ориентирована на демонстрацию основных возможностей MS Visio в приложении к задаче создания и описания фрагмента бизнес-процесса формирования аналитической отчетности по различным информационным срезам перевозки грузов с использованием собственного и арендованного парка контейнеров.

Содержание методических материалов основано на рабочих материалах, которые автор использовал при проведении практических занятий по дисциплине «Моделирование и анализ бизнес-процессов» со студентами, обучающимися по направлению подготовки 080700 «Бизнес-информатика».

Подбор содержательной части заданий производился с учетом взаимосвязи с общепрофессиональными и специальными дисциплинами, изучаемыми студентами на последующих курсах.

С благодарностью примем замечания, комментарии и пожелания уважаемых читателей по содержанию и оформлению методических указаний, которые можно направлять по адресу feano@inbox.ru

Построение диаграммы классов на основе шаблона Схема модели UML MS Office Visio 2007

Ключевые слова: модель, класс, атрибуты класса, операции класса, сотрудник, сотрудник для связи с окружением, бизнес–сущность, интерфейс, пакет, отношения между классами.

1. Основные теоретические сведения

Диаграмма классов является одной из канонических диаграмм UML, создаваемой для визуализации структурированной статической модели предметной области. Этот вид диаграмм представляет собой графическое изображение объектов – классов с присущими им атрибутами, операциями и различных отношений между классами.

1.1. Классы

Класс (class) служит для обозначения множества объектов, обладающих функциональным набором одинаково описывающих параметров (атрибутов), реализуемых операций и однотипными отношениями с объектами других классов.

На диаграмме класс изображается габаритной прямоугольной рамкой, которая дополнительно может быть разделена горизонтальными линиями на секции, каждая из которых предназначена для указания имени, атрибутов (свойств) и реализуемых операций объектов данного класса (рис.1 а, б, в).

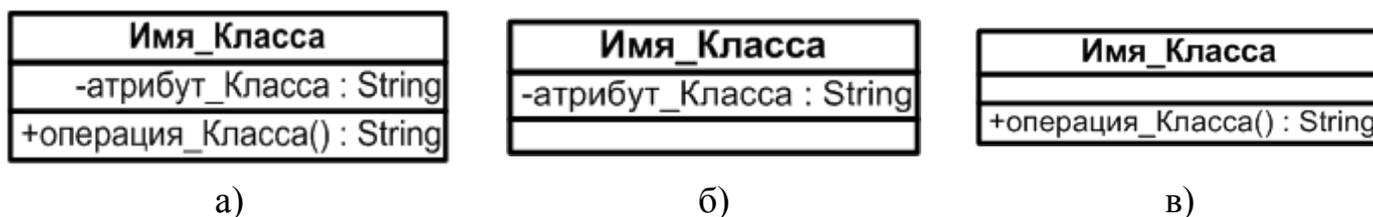


Рис. 1. а), б), в). Обозначение класса, атрибутов, операций

Имя класса является обязательным элементом в его обозначении и должно быть уникальным (хотя бы в пределах пакета), имеющее непосредственное отношение к контексту моделируемой предметной области.

В соответствии с принятым в языке UML общим соглашением в качестве имени класса используются существительные и прилагательные, каждое из которых начинается с заглавной буквы, записанные без пробелов. Например, в качестве имен классов могут быть использованы профессиональные термины: «Сотрудник», «Компания», «Руководитель», «Клиент», «Продавец», «Менеджер», «Офис», «Покупатель», «Датчик_Температуры» и др. Такие имена классов являются простыми.

Иногда возникает необходимость в явном указании пакета, к которому относится данный класс. С этой целью в условном обозначении перед именем класса указывается имя пакета и специальный символ разделитель – двойное двоеточие "::<". Такое имя класса является квалифицированным. Текстовая строка имени класса в этом случае записывается в формате <Имя_пакета>::<Имя_класса>. Например, если определен пакет с именем «Банк», то имя класса «Счет» может быть записано так, как показано на рис.2.



Рис. 2. Обозначение квалифицированного имени класса

Одним из существенных достоинств языка UML является наличие механизмов расширения в виде стереотипов, именованных значений, графических обозначений, которые позволяют уточнить синтаксис и семантику модели определенной предметной области.

Язык UML включает в себя два специальных расширения: профиль для процесса разработки программного обеспечения (The UML Profile for Software Development Processes) и профиль для бизнес-моделирования деятельности организации (The UML Profile for Business Modeling).

В рамках первого профиля The UML Profile for Software Development Processes существуют специальные графические обозначения, используемые для уточнения семантики отдельных классов:

Управляющий класс (control class) — класс, отвечающий за координацию действий других классов. Как правило, данный класс является активным и инициирует рассылку множества сообщений другим классам модели. Управляющий класс может быть изображен в форме прямоугольника класса со стереотипом <<control>> (рис. 3, а).

Класс-сущность (entity class) — пассивный класс, информация о котором должна храниться постоянно и не уничтожаться с выключением системы или завершением программы. Этот класс принимает сообщения от других классов модели. Как правило, этот класс соответствует отдельной таблице базы данных. Класс-сущность может быть изображен также стандартным образом в форме прямоугольника класса со стереотипом <<entity>> (рис. 3, б).

Граничный класс (boundary class) — класс, который располагается на границе системы с внешней средой, но является составной частью системы. Граничный класс может быть изображен также стандартным образом в форме прямоугольника класса со стереотипом <<boundary>> (рис. 3, в).

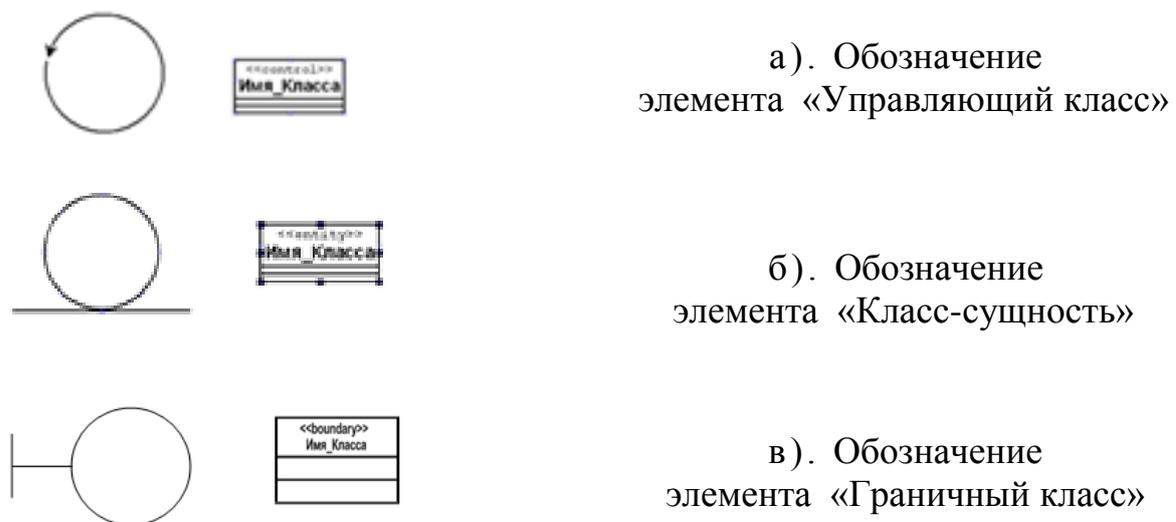
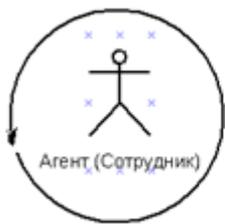


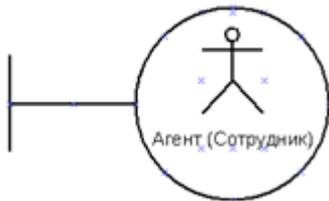
Рис. 3. (а, б, в). Обозначение классов профиля The UML Profile for Software Development Processes

В рамках второго профиля The UML Profile for Business Modeling также существуют специальные графические примитивы, которые могут быть использованы для уточнения семантики отдельных классов при построении моделей бизнес-систем:

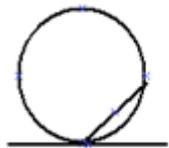
Сотрудник (business worker) — класс, служащий на диаграмме классов для представления любого сотрудника, который является элементом бизнес-системы и взаимодействует с другими сотрудниками при реализации бизнес-процесса. Этот класс также может быть изображен в форме прямоугольника класса со стереотипом <<worker>> или <<internalWorker>> (рис. 4, а).



а). Обозначение элемента «Сотрудник»



б). Обозначение элемента «Сотрудник для связи с окружением»



в). Обозначение элемента «Бизнес-сущность»

Рис.6. а), б), в). Обозначение классов профиля The UML Profile for Business Modeling

Сотрудник для связи с окружением (caseWorker) – класс, служащий для представления в бизнес-системе такого сотрудника, который, являясь элементом бизнес-системы, непосредственно взаимодействует с актерами (бизнес-актерами) при реализации бизнес-процесса. Условно графически класс может быть представлен в форме изображения Агента (Сотрудника), заключенного в окружность или в форме прямоугольника класса со стереотипом <<caseWorker>> (рис.4, б).

Бизнес-сущность (business entity) — специальный случай класса-сущности, который также не инициирует никаких сообщений. Этот класс служит для сохранения информации о результатах выполнения бизнес-процесса в модели-

руемой бизнес–системе или организации. Этот класс также может быть изображен в форме прямоугольника класса со стереотипом <<business entity>> (рис. 4, в).

Для расширения возможностей моделирования язык UML предоставляет некоторые дополнительные классификаторы, в частности интерфейс - классификатор, который представляет собой набор операций, используемых для спецификации сервиса класса. Такие классы служат для спецификации тех элементов модели, которые видимы извне, но их внутренняя структура остается скрытой.

Обозначение интерфейса на диаграмме может быть создано несколькими способами, на одном из которых указывается стереотип <<interface>> (рис.5).

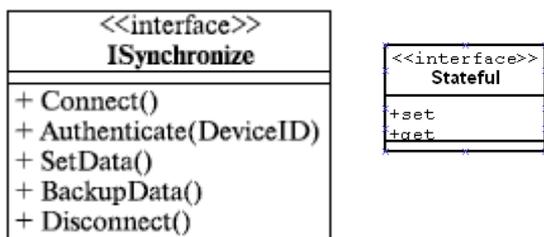


Рис.5. Обозначение стереотипа <<interface>>

Часто используемым в моделировании классификатором является классификатор Подсистема – компонент, представляющий главную часть системы. Этот классификатор изображается графически так, как показано на рис.6.

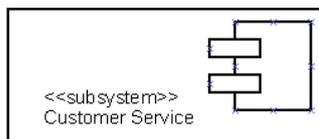


Рис.6. Обозначение классификатора Подсистема – компонент

В организационной части моделирования языка UML присутствуют группирующие сущности, одна из которых – пакет, представляющий собой некоторый универсальный механизм организации набора классов (пакетов), если группировка последних уместна.

Пакет физически содержит сущности (классы, интерфейсы и др.), определенные в нем («сущности принадлежат пакету»), что означает, что если будет уничтожен пакет, то будет уничтожено все его содержимое.

Графически условное обозначение пакета напоминает «папку» с закладкой. Имя пакета указывается на папке, если ее содержимое не показывается, в противном случае на самой закладке (рис.7,а),б)).

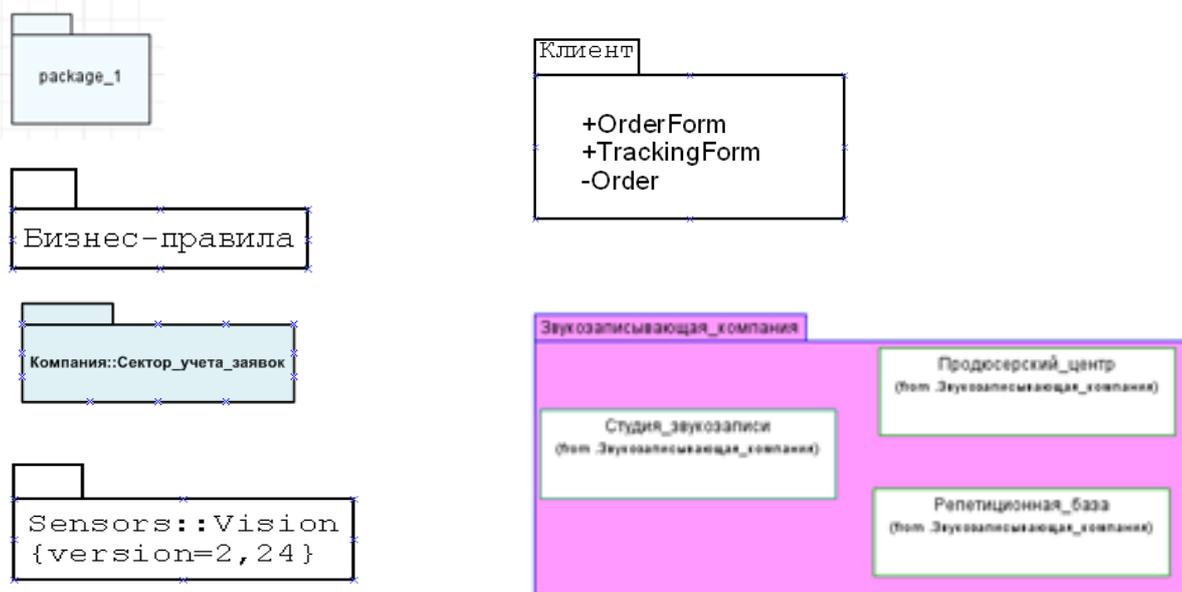


Рис.7. Обозначение пакета

Один пакет может содержать другие пакеты, а, следовательно, допускается иерархическая декомпозиция имени, например пакет АРМ_менеджера (Автоматизированное рабочее место менеджера) содержит класс Компьютер; при этом пакет АРМ_менеджера в свою очередь содержится в пакете АРМ. Полное имя класса: АРМ::АРМ_менеджера::Компьютер.

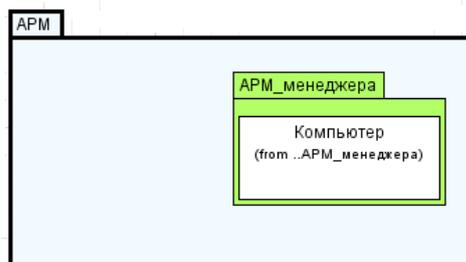


Рис.8. Обозначение составного пакета

Такая организация элементов позволяет визуализировать группы элементов, с которыми можно обращаться как с единым целым, контролируя при этом видимость и возможность доступа к отдельным элементам.

1.2. Атрибуты классов

Содержательной характеристикой класса является атрибут, содержащий множество значений, которые могут принимать отдельные объекты этого класса. При этом, класс может иметь любое число атрибутов или не иметь ни одного. Так, например, атрибутами класса «Усилитель» являются частотный диапазон, выходная мощность, коэффициент нелинейных искажений, уровень шума и т. д.

Запись атрибута также представляет собой отдельную строку текста, содержащую обязательное имя, в котором обычно каждое слово пишется с заглавной буквы, за исключением первого, например, name (имя) или birth_Date (дата_Рождения), квантор видимости атрибута, кратность, тип значений атрибута и, возможно, его исходное значение.

Например на рис.9 указаны атрибуты класса Контейнер, в качестве которых выступают атрибут тип_Контейнера, атрибут регистрационный_Номер_Контейнера, регистрационный_номер_после_перерегистрации, рабочее_состояние.

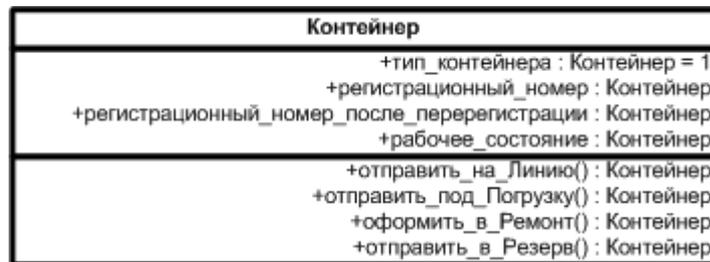


Рис.9. Указание атрибутов класса Контейнер

Качественной характеристикой описания элементов класса является квантор видимости атрибута – потенциальная возможность других объектов модели оказывать влияние на отдельные аспекты поведения данного класса.

Эта характеристика может принимать одно из трех возможных значений и, соответственно, отображается при помощи специальных символов:

символ "+" (public) обозначает атрибут с областью видимости типа общедоступный; атрибут с этой областью видимости доступен или виден из любого другого класса пакета, в котором определена диаграмма; например, для класса Class_1 указан атрибут общедоступного типа (рис.10,а);

символ "#" (protected) обозначает атрибут с областью видимости типа защищенный; атрибут с этой областью видимости недоступен или невиден для всех классов,

за исключением подклассов данного класса; например, для класса Class_2 указан атрибут защищенного типа (рис.10, б);

символ "-" (private) обозначает атрибут с областью видимости типа закрытый; атрибут с этой областью видимости недоступен или невиден для всех классов без исключения; например, для класса Class_3 указан атрибут защищенного типа (рис.10,в);

Квантор видимости при описании атрибутов может быть опущен, что будет означать тот факт, что видимость атрибута не указывается.

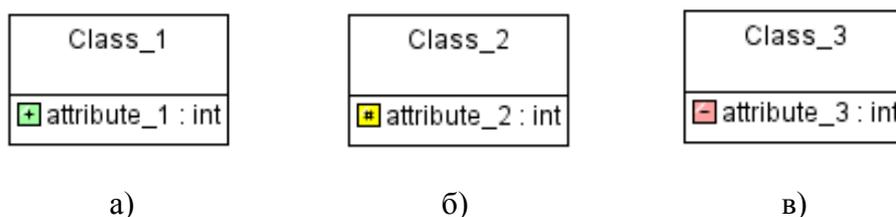


Рис.10. а), б), в). Обозначение квантора видимости атрибутов класса

Спецификацией области значений атрибутов конкретного типа является кратность (multiplicity) атрибута.

В общем случае кратность атрибута записывается в форме строки текста из цифр в квадратных скобках после имени соответствующего атрибута в виде: [нижняя граница ... верхняя граница] (рис.11).

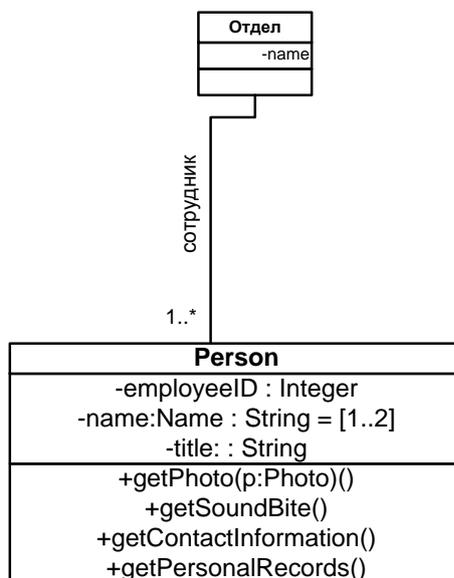


Рис.11. Обозначение кратности атрибута Name класса Person

В качестве верхней границы может использоваться специальный символ "*" (звездочка), который означает произвольное положительное целое число, т.е. неограниченное сверху значение кратности соответствующего атрибута.

Если в качестве кратности указывается единственное число, то кратность атрибута принимается равной данному числу. Если кратность атрибута не указана, то по умолчанию принимается ее значение равное [1..1], т. е. в точности 1.

Различают значения кратности: 0..1 (ноль или один); 1 (только один), 0..* (ноль или много); 1..* (один или много); n (только n (где $n > 1$)); 0..n (ноль к n (где $n > 1$)); 1...n (один к n (где $n > 1$)).

Тип атрибута представляет собой выражение, семантика которого определяется некоторым типом данных, определенным в пакете Типы данных языка UML или самим разработчиком.

Например, для атрибутов классов Class_1, Class_2, Class_3 указан целочисленный тип данных Int (Integer) языков C#, C++, VBA (в нотации UML тип атрибута иногда определяется в зависимости от языка программирования, который предполагается использовать для реализации данной модели).

Исходное значение служит для задания начального значения соответствующего атрибута в момент создания отдельного экземпляра класса. Если исходное значение не указано, то значение соответствующего атрибута не определено на момент создания нового экземпляра класса.

При задании атрибутов могут быть использованы дополнительные синтаксические конструкции — это подчеркивание строки атрибута, пояснительный текст в фигурных скобках и косая черта перед именем атрибута.

1.3. Операции классов

Операция (operation) класса – это реализация услуги, которая может быть запрошена у любого объекта данного класса, чтобы вызвать определенное его поведение. Класс может иметь любое число операций либо не иметь ни одной. Так автомобиль может перемещаться по грунту, корабль – перемещаться по воде, компьютер – производить вычисления.

Представление полного синтаксиса записи операций класса также подчиняется определенным синтаксическим правилам: каждой операции класса соответствует отдельная строка, которая состоит из квантора видимости операции, обяза-

тельного имени операции, выражения типа возвращаемого операцией значения и, возможно, строки-свойства данной операции:

<квантор видимости><имя операции>(список параметров): <выражение типа возвращаемого значения>{строка-свойство}

Квантор видимости, как и в случае атрибутов класса, может принимать одно из трех возможных значений и, соответственно, также отображается при помощи специального символа.

Для именованной операции используются короткие глагольные конструкции, описывающие некоторое поведение класса, которому принадлежит операция. Обычно каждое слово в имени операции пишется с заглавной буквы, за исключением первого.

Например, запись +создать() – может обозначать абстрактную операцию по созданию отдельного объекта класса, которая является общедоступной и не содержит формальных параметров, запись +нарисовать(форма: Многоугольник = прямоугольник, цвет_заливки: Color = (0, 0, 255)) – может обозначать операцию по изображению на экране монитора прямоугольной области синего цвета, если не указываются другие значения в качестве аргументов данной операции.

(список–параметров) содержит необязательные аргументы, синтаксис которых совпадает с синтаксисом атрибутов;

< выражение типа возвращаемого значения > является необязательной спецификацией и зависит от конкретного языка программирования;

{строка-свойство} показывает значения свойств, которые применяются к данной операции.

Например, запись запросить_Счет_Клиента(номер_счета:Integer) – обозначает операцию по установлению наличия средств на текущем счете клиента банка. При этом аргументом данной операции является номер счета клиента, который записывается в виде целого числа (например, «123456»).

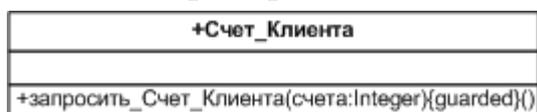


Рис.12.Обозначение операции «запросить_Счет_Клиента»

Квантор видимости для операции может быть опущен. В этом случае его отсутствие означает, что видимость операции не указывается.

1.4. Отношения между классами

Классы на диаграмме связываются различными типами отношений. При этом совокупность типов таких отношений фиксирована в языке UML и predeterminedена их семантикой.

1.4.1. Отношение зависимости

Отношением зависимости (dependency relationship) называют связь по использованию, когда изменение в спецификации одного класса может повлиять на поведение другого. Отношение зависимости используется в такой ситуации, когда некоторое изменение одного элемента модели может потребовать изменения другого зависимого от него элемента модели. Отношение зависимости графически изображается пунктирной линией между соответствующими элементами со стрелкой на одном из ее концов, направленной к той сущности, от которой зависит данная сущность. Например, некая сущность Класс-2 использует другую сущность Класс_1 (рис.13).

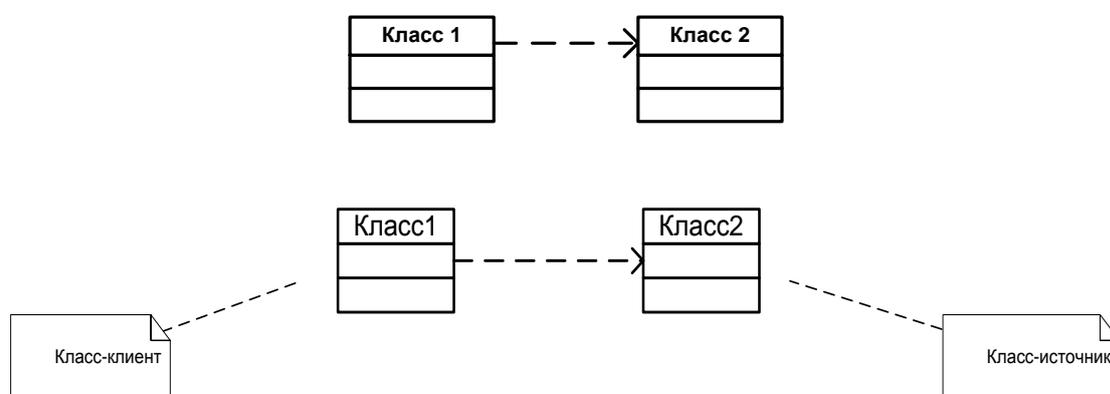


Рис.13. Обозначение отношения зависимости

В качестве класса-клиента и класса-источника зависимости может выступать множество элементов модели. В этом случае одна линия со стрелкой, выходящая от источника зависимости, расщепляется в некоторой точке на несколько отдельных линий, каждая из которых имеет отдельную стрелку для класса-клиента.

Например, если функционирование сущности Класс_С зависит от особенностей реализации сущностей Класс_А и Класс_Б, то данная зависимость может быть изображена следующим образом (рис. 14).

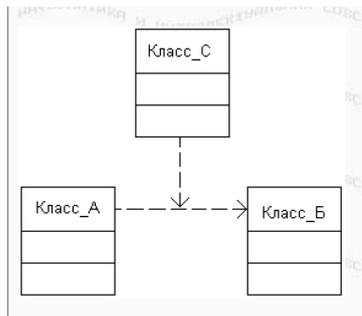


Рис. 14. Обозначение зависимости между классом–клиентом (Класс_С) и классами–источниками (Класс_А и Класс_Б)

Для отношения зависимости предусмотрены дополнительные возможности, позволяющие описывать некоторые смысловые оттенки, специфические для конкретной решаемой задачи. UML предоставляет для этой цели набор стереотипов: `derive`, `permit`, `instanceOf`, `instantiate` и др.

Например, стереотип `derive` для отношения зависимости показывает, что источник может быть вычислен по цели (рис.15):

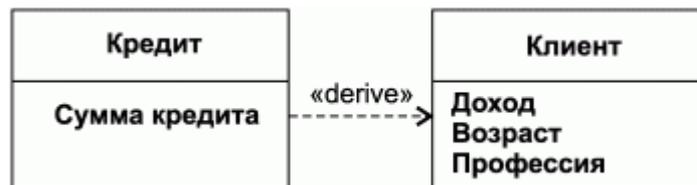


Рис.15. Указание стереотипа «derive» для отношения зависимости

Использовать стереотип `derive` следует при моделировании связи между двумя атрибутами или двумя ассоциациями, одна из которых конкретна, а вторая концептуальна. Например, класс `Человек` может иметь конкретный атрибут `дата_Рождения`, а также атрибут `возраст`, который происходит от атрибута `дата_Рождения`, и может быть выведен из даты рождения человека.

1.4.2. Отношение ассоциации

Ассоциацией (*association relationship*) называется структурная связь, показывающая, что объекты одного класса некоторым образом связаны с объектами другого или того же самого класса.

Ассоциация может отображаться графически линией со стрелкой (маркером в виде треугольника), показывающей направление следования классов и кратность - количество объектов, связанных отношением. Отсутствие стрелки рядом с именем ассоциации означает, что порядок следования классов в рассматриваемом отношении не определен (рис.16).

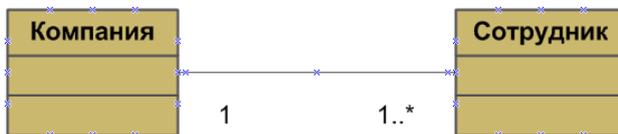


Рис.16. Обозначение отношения ассоциации

Так, примера на рис.16 кратность «1» для класса «Компания» означает, что каждый сотрудник может работать только в одной компании. Кратность «1..*» для класса «Сотрудник» означает, что в каждой компании могут работать несколько сотрудников, общее число которых заранее неизвестно и ничем не ограничено.

Кроме направления ассоциации, могут быть указаны на диаграмме роли, которые каждый класс играет в данном отношении, например, роль «ведет учет заявок» (рис.17).

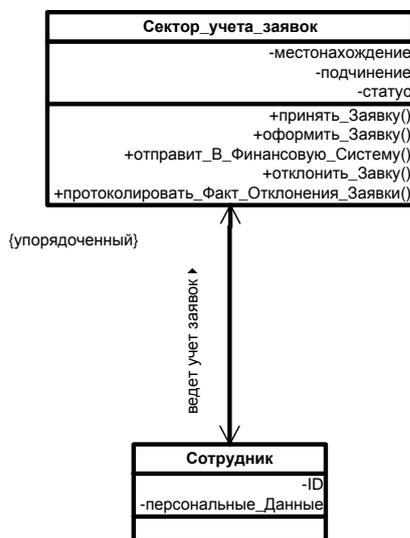


Рис. 17. Обозначение роли «Ведет учет заявок» отношения классов

Допускается создание ассоциаций, связывающих сразу три и более класса классов (они называются N-арными ассоциациями).

В этом случае, ассоциация изображается ромбом на пересечении линий, как показано на этой диаграмме, (рис.18)¹:

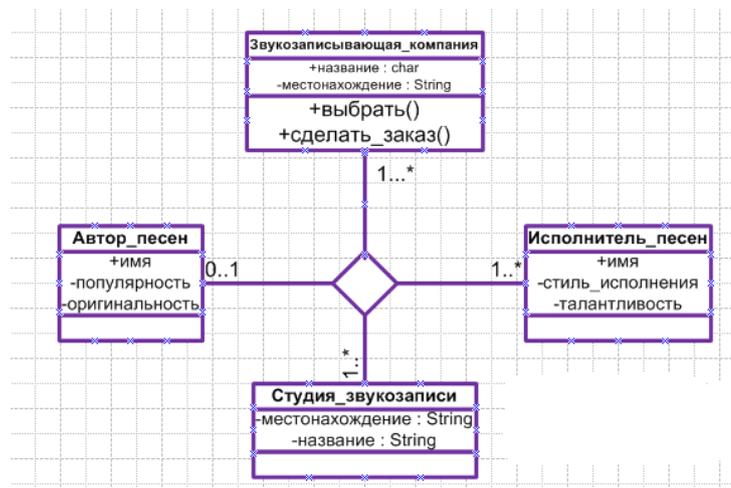


Рис. 18. Обозначение N-арной ассоциации

Частным случаем отношения ассоциации является так называемая исключаящая ассоциация (Xor-association).

На диаграмме классов исключаящая ассоциация изображается пунктирной линией, соединяющей две и более ассоциации, рядом с которой записывается строка - ограничение "{хор}".

Например, счет в банке может быть открыт для клиента, в качестве которого может выступать физическое лицо (индивидуум) или компания, что изображается с помощью исключаящей ассоциации (рис. 19).

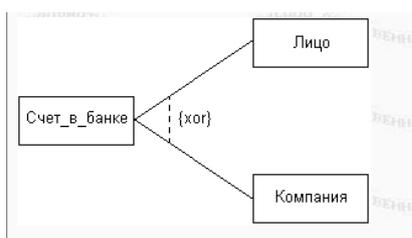


Рис. 19. Обозначение исключаящей ассоциации между тремя классами²

Специальной формой или частным случаем отношения ассоциации является отношение агрегации, которое, в свою очередь, тоже имеет специальную форму – отношение композиции.

¹ Рисунок создан на основе источника

< http://www.intuit.ru/department/se/intuml/class/free/3/intuml_3.html#image.3.13 >

² Источник рисунка < http://www.intuit.ru/department/se/intuml/class/free/3/intuml_3.html#image.3.13 >

1.4.3. Отношение агрегации

Отношение агрегации (generalization relationship) имеет место между несколькими классами в том случае, если один из классов представляет собой некоторую сущность, включающую в себя в качестве составных частей другие сущности.

Данное отношение имеет фундаментальное значение для описания структуры сложных систем, поскольку применяется для представления системных взаимосвязей типа «часть – целое».

Это отношение по своей сути описывает декомпозицию или разбиение сложной системы на более простые составные части, которые также могут быть подвергнуты декомпозиции, если в этом возникнет необходимость в последующем.

Так, автомобиль состоит из кузова, двигателя, трансмиссии и т.п., а в состав приемопередающего устройства входят передатчик, приемник и антенно-фидерное устройство.

Графически отношение агрегации изображается сплошной линией, один из концов которой представляет собой геометрическую фигуру – ромб. Этот ромб указывает на тот из классов, который представляет собой «целое» (рис. 20).

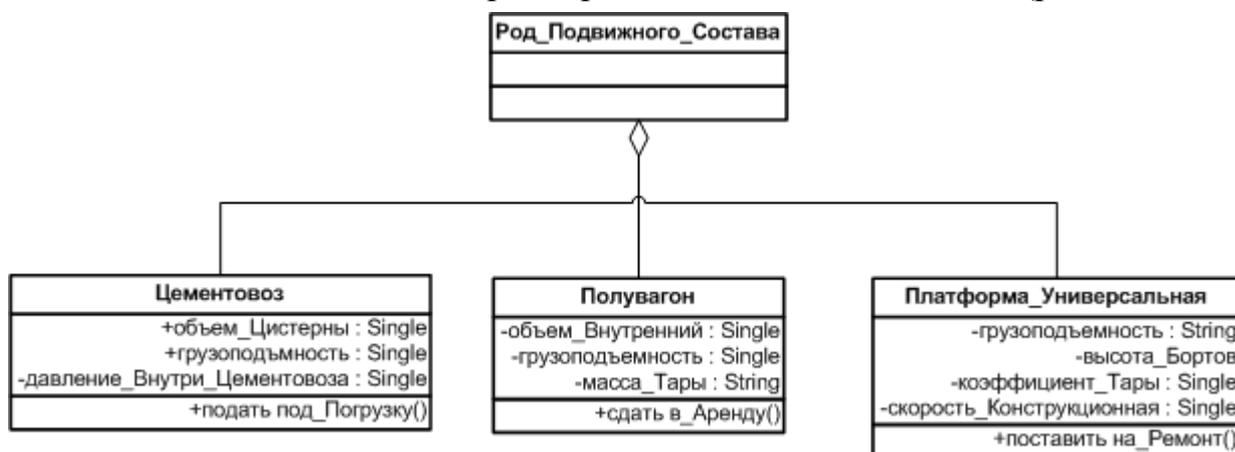


Рис. 20. Обозначение отношения агрегации

Примером отношения агрегации может служить деление класса Аналитическая_информация на составные части: Отчет_по_грузу, Отчет_по_контейнерам, Отчет_по_тарифам (рис.21).



Рис. 21. Обозначение на диаграмме классов отношения агрегации

Отношение агрегации обладает кратностью. Так, класс Система_обеспечения_безопасности_объектов может содержать содержит одну подсистему Сопровождение_грузов, которая в свою очередь может содержать, например, четыре класса Охрана_вооруженная, каждый из которых может принадлежать лишь одному классу Сопровождение_грузов (рис.22).



Рис.22. Обозначение кратности отношения агрегации

1.4.4.Отношение композиции

Отношение композиции (realization relationship) служит для выделения специальной формы отношения «часть-целое», при которой составляющие части в некотором смысле находятся внутри целого.

Специфика взаимосвязи между ними заключается в том, что части не могут выступать в отрыве от целого, т. е. с уничтожением целого уничтожаются и все его составные части.

Графически отношение композиции изображается сплошной линией, один из концов которой представляет собой закрашенный внутри ромб. Этот ромб указывает на тот из классов, который представляет собой класс-композицию или «целое» (рис. 23).



Рис.23. Обозначение отношения композиции

В качестве дополнительных обозначений для отношений композиции и агрегации могут использоваться дополнительные элементы, применяемые для отношения ассоциации. А именно, указание кратности класса ассоциации и имени данной ассоциации, которые не являются обязательными.

Применительно к классу `Заказ_на_перевозку_грузов` отношение композиции может иметь следующий вид (рис. 24).

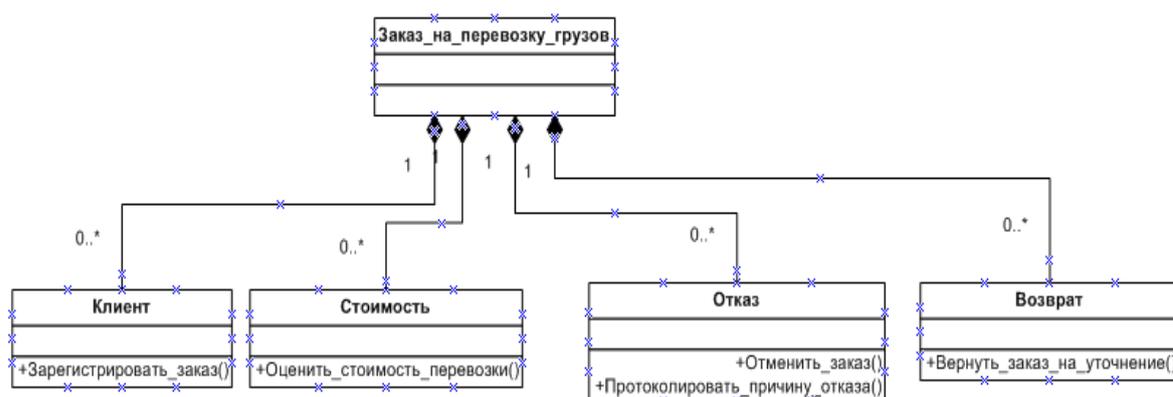


Рис. 24. Обозначение на диаграмме отношения композиции

1.4.5. Отношение обобщения

Отношение обобщения (генерализация) является обычным таксономическим отношением между более общим элементом (класс-предок) и более частным или специальным элементом (класс-потомок).

Применительно к диаграмме классов данное отношение описывает иерархическое строение классов и наследование их свойств и поведения. При этом предполагается, что класс-потомок обладает всеми свойствами и поведением класса-предка, а также имеет свои собственные свойства и поведение, которые отсутствуют у класса-предка.

На диаграммах отношение обобщения обозначается сплошной линией с треугольной стрелкой на одном из концов, направленной на более общий класс (класс-предок или суперкласс) от более специального класса (класса-потомка или под-класса) (рис.25).

Как правило, на диаграмме может указываться несколько линий для одного отношения обобщения, что отражает его таксономический характер. В этом случае более общий класс разбивается на подклассы одним отношением обобщения, например, так, как показано на рис. 25.

В этом случае данные отдельные линии изображаются сходящимися к единственной стрелке, имеющей с ними общую точку пересечения. Родительский Класс Отчет_По_Заказам_На_Перевозку имеет три потомка Отчет_По_Количеству_Заказов, Отчет_По_Клиентам, Отчет_За_Период, которые наследуют структуру и поведение родительского класса.



Рис. 25. Обозначение на диаграмме отношения обобщения

Для связей обобщения язык UML содержит ограничения. В большинстве случаев ограничение размещается рядом с элементом и заключается в фигурные скобки, например {complete}.

В качестве ограничений могут быть использованы следующие ключевые слова языка UML:

{complete} означает, что в данном отношении обобщения специфицированы все классы-потомки, и других классов-потомков у данного класса-предка быть не может.

Например, класс Клиент_банка является предком для двух классов: Физическое_лицо и Компания, и других классов-потомков он не имеет.

На соответствующей диаграмме классов это можно указать явно, записав рядом с линией обобщения данную строку-ограничение (рис.26).

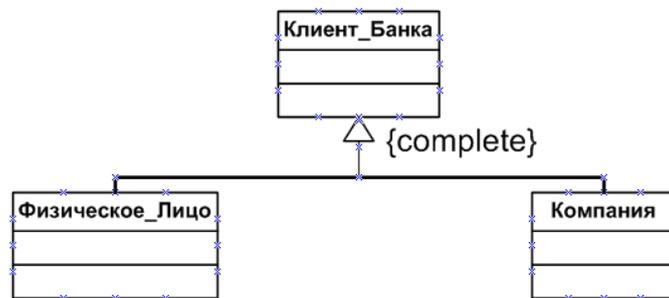


Рис.26. Обозначение ограничения {complete} отношения обобщения

{incomplete} означает тот факт, что на диаграмме указаны в обобщении не все классы-потомки. В последующем, возможно, восполнить их перечень не изменяя уже построенную диаграмму.

{disjoint} означает тот факт, что классы-потомки не могут содержать объектов, одновременно являющихся экземплярами двух или более классов.

В приведенном выше примере это условие также выполняется, поскольку предполагается, что никакое конкретное физическое лицо не может являться одновременно и конкретной компанией. В этом случае рядом с линией обобщения можно записать данную строку-ограничение.

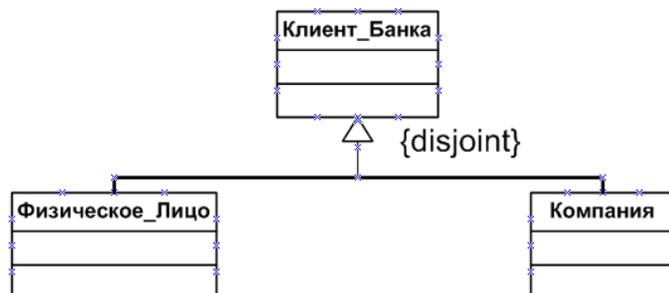


Рис.27. Обозначение ограничения {disjoint} отношения обобщения

{overlapping} означает, что отдельные экземпляры классов-потомков могут принадлежать одновременно нескольким классам.

Например, класс Транспорт может быть специализирован путем создания подклассов Наземный_Транспорт и Водный_Транспорт, автомобиль - амфибия относится к обоим классам.

На следующем рисунке представлен фрагмент диаграммы классов, содержащий разные отношения между классами, а также их отдельные параметры (рис.28).

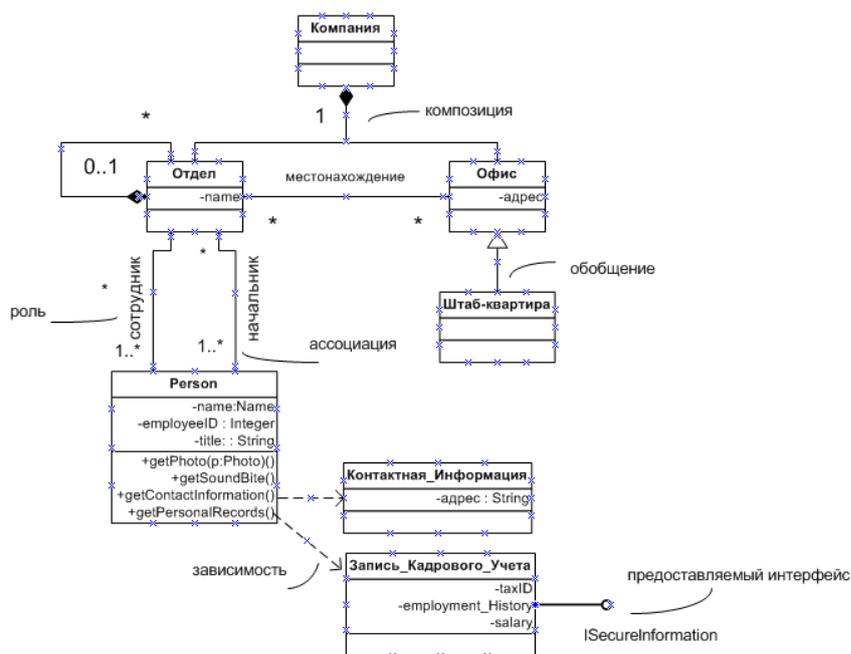


Рис.28. Фрагмент диаграммы классов

2. Практические задания

Создание диаграммы классов на основе шаблона Схема модели UML MS Office Visio 2007

Цель работы: получить навыки построения диаграмм классов, применить приобретенные навыки для построения объектно–ориентированных моделей на основе описания фрагмента определенной предметной области.

Требования к содержанию отчета: отчет содержит описание классов определенной предметной области, основных отношений между классами, создание диаграммы классов, оформление в виде текстового документа.

Задание:

На основании описания фрагмента *Аналитическая отчетность по различным информационным срезам перевозки грузов с использованием собственного и арендованного парка контейнеров* для предприятия N (перечень основных бизнес-процессов представлен в Приложении) выделить основные классы, установить отношения, создать диаграмму классов.

Порядок выполнения практических заданий

1. Откройте приложение **MS Office Visio** выберите в **Категории шаблонов** шаблон **Программное обеспечение и базы данных** (рис.35)

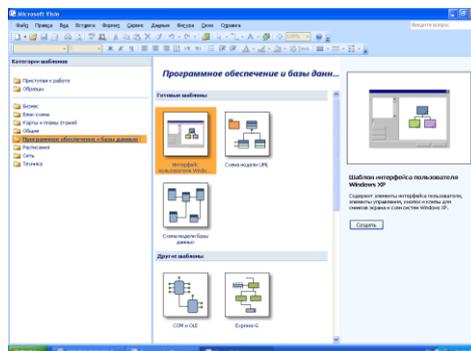


Рис.35. Выбор шаблона Программное обеспечение и базы данных

2. Выберите **Готовый шаблон – Схема модели UML** и выполните сохранение документа в личной папке с именем **Диаграмма классов. vsd**.
3. Ознакомьтесь с элементами графического интерфейса и найдите обязательные панели инструментов **Стандартная**, **Форматирование** (при необходимости отразите их с помощью команды горизонтального меню Вид/Панели инструментов), Область - **Фигуры**, содержащие категории **Деятельность UML**, **Взаимодействия UML**, **Компоненты UML**, **Топология UML**, **Последовательности UML**, **Схема состояний UML**, **Статическая структура UML**, **Сценарий выполнения UML**, **Проводник по моделям**, содержащий иерархическую структуру объектов Системы UML1, Рабочую область, ярлык Страница_1, горизонтальную и вертикальную линейки.

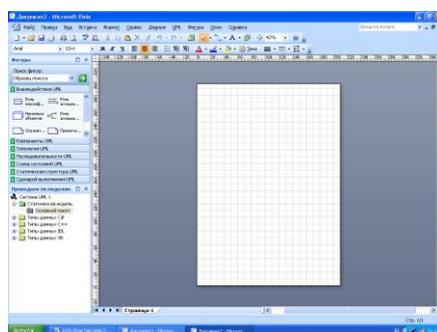


Рис.37. Рабочая область MS Office Visio

4. Установите с помощью команды **Файл/Параметры страницы** и диалогового окна **Параметры страницы** следующие опции: на вкладке **Свойства страни-**

цы - укажите имя – *Диаграмма классов формирования аналитической отчетности*, единицы измерения – миллиметры, тип – передняя, на вкладке **Настройка печати** – Бумага в принтере – Альбомная, Масштаб – 100%, разместить – на одном листе, на вкладке **Размер страницы** – как в принтере, на вкладке **Тени** – опции выберите по своему усмотрению, на вкладке **Макеты и маршруты** – направление – сверху вниз, установите опцию – сдвигать другие фигуры при размещении, интервал между фигурами примите – по умолчанию, примените параметры, нажав кнопку ОК.

5. Установите масштаб видимости достаточно удобный для работы с листом документа, например 100%.
6. Перейдите в категорию **Статическая структура UML**, ознакомьтесь с содержимым этой категории и найдите элементы: Класс, Пакет, Подсистема, Интерфейс, Метакласс, Двусторонняя ассоциация, Обобщение, Композиция, Примечание, Ограничение и др.
7. Отобразите дополнительные панели с помощью команды горизонтального меню Вид: **Панорама и масштаб, Размер и положение**, дополнительные панели инструментов с помощью команды горизонтального меню Вид /Панели инструментов: **Набор элементов, Разработчик, Привязать и приклеить** (кнопка Приклеивание к контуру фигуры, Приклеивание к маркерам фигуры, Приклеивание к вершинам фигуры, Привязка к пересечениям фигур).
8. Создайте поэтапно статическую структуру классов UML, с помощью которой может быть сформирована некоторая функциональная часть системы, например *Аналитическая отчетность по различным информационным срезам перевозки грузов с использованием собственного и арендованного парка контейнеров для предприятия N* (перечень основных бизнес-процессов представлен в Приложении). Для чего:
 - Выберите структурные элементы (идентифицируйте классы), участвующие в формировании аналитических отчетов, например, *Аналитическая информация, Аналитическая информация по грузу, Аналитическая информация по контейнерам, Аналитическая информация по тарифам, Аналитическая информация по счетам, Аналитический отчет по грузу, Аналитический отчет по контейнеру, Контейнер* и создайте предварительный вариант совокупности классов с указанием имен (один из возможных вариантов представлен на рис.38).

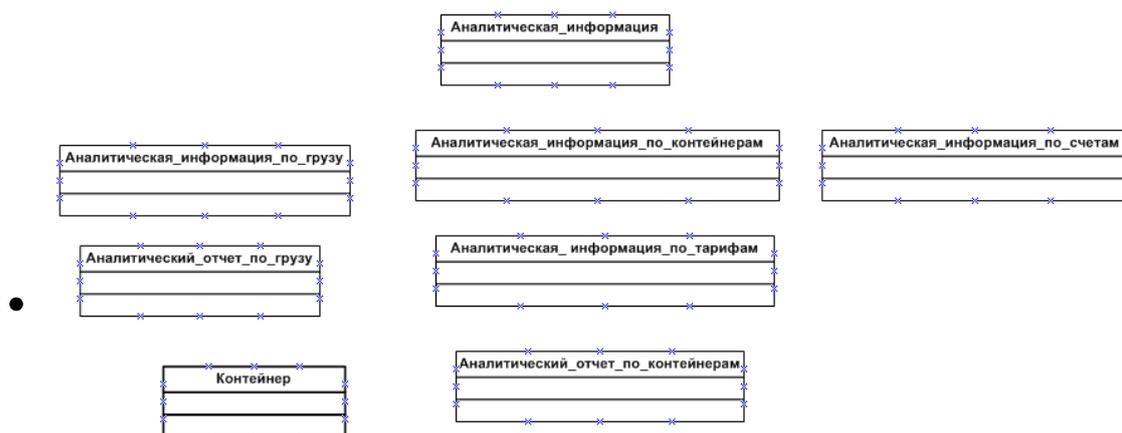


Рис.38. Формирование статической структуры объектов диаграммы

- Установите для каждого класса атрибуты в соответствии с перечнем и содержательным описанием бизнес-процессов:

например, для класса Контейнер в качестве атрибутов могут выступать данные: тип контейнера (универсальный, специализированный); форма собственности (собственный, арендованный); государственный регистрационный номер (возможна перерегистрация с сохранением всех эксплуатационных данных); состояние контейнера (на линии, в ремонте и т.п.). В данном случае все атрибуты видимы, принадлежат основному пакету Контейнер, для атрибута тип_контейнера установлено исходное значение – 1 (рис.39).

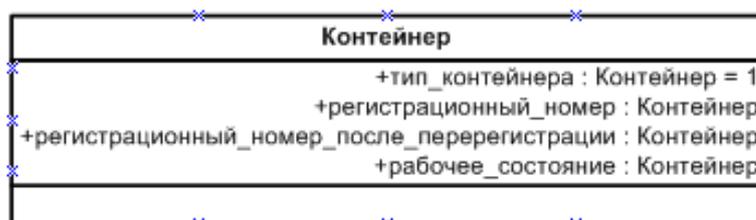


Рис.39.Описание атрибутов класса Контейнер

для класса Аналитический_отчет_по_грузу, в качестве атрибутов могут выступать данные: по сортировке, по отправлениям, по самопривозу/самовывозу, по транзитному грузу, по ценным грузам (рис.40).

Аналитический_отчет_по_грузу
-по_сортировке
-по_отправкам
-по_самопривозу/самовывозу
-по_транзитному_грузу
-по_ценным_грузам
+выпустить_Отчет_по_Запросу()
+выпустить_Отчет()

Рис.40. Описание атрибутов и операций класса Аналитический_отчет_по_грузу

для класса Аналитический_отчет_по_контейнерам, в качестве атрибутов могут выступать данные: по_рейсам, по_пробегу, по_пунктам_маршрута (рис.41)

Аналитический_отчет_по_контейнерам
-по_рейсам
-по_пробегу
-по_пунктам_маршрута
+выпустить_Отчет_по_Запросу()
+выпустить_Отчет()

Рис.41. Описание атрибутов и операций класса Аналитический_отчет_по_контейнерам

для класса Аналитическая_информация_по_счетам, в качестве атрибутов могут выступать данные: тип_счета, модель_счета, способ_выставления_счета (рис.42)

Аналитическая_информация_по_счетам
-тип_счета
-модель_счета
-способ_выставления_счета
+подготовить_Счет()
+проконтролировать_Доставку_Счета()
+проконтролировать_Оплату_Счета_Клиентом()

Рис.42. Описание атрибутов и операций класса Аналитическая_информация_по_счетам

- Установите для каждого класса операции:

для класса Контейнер в качестве операций могут выступать данные: отправить под погрузку, отправить на текущий ремонт, отправить на капитальный ремонт, отправить в резерв, отправить на линию и др. (рис.43). В данном примере операции видимы, принадлежат основному пакету Контейнер.

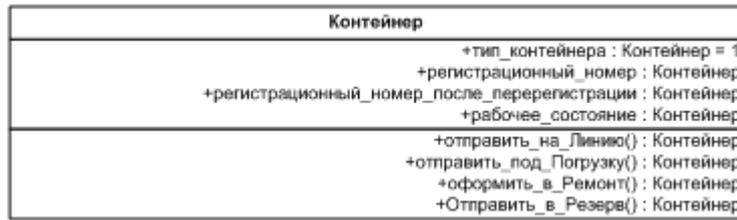


Рис.43.Описание операций класса Контейнер

для класса Аналитический_отчет_по_грузу в качестве операций могут выступать данные: выпустить_Отчет_по_запросу, выпустить_Отчет (рис.40);

для класса Аналитический_отчет_по_по_контейнерам в качестве операций могут выступать данные: выпустить_Отчет_по_запросу, выпустить_Отчет (рис.41);

для класса Аналитический_отчет_по_по_счетам в качестве операций могут выступать данные: подготовить_Счет, проконтролировать_Доставку_Счета, проконтролировать_Оплату_Счета_Клиентом (рис.42);

- Убедитесь, что все элементы наполнены адекватным содержанием и расположите все структурные элементы диаграммы наиболее оптимально на странице для установления отношений между ними.

В качестве примера на рис.44 показан набор классов, описывающих реализацию формирования аналитической отчетности по информационным срезам предприятия N. Акцент сделан на классе Аналитическая_информация, с которым связаны классы Аналитическая_информация_по_грузу, Аналитическая_информация_по_контейнерам, Аналитическая_информация_по_счетам, Аналитическая_информация_тарифам посредством отношения композиции, т.е.полного владения, для всех связей показано отношение «1...*», что будет означать наличие аналитической информации (отчетов), общее число которых заранее неизвестно и ничем не ограничено. Наличие отношения агрегации для классов Аналитический_отчет_по_грузу, Аналитический_отчет_по_контейнерам с соответствующими классами говорит о этих классах как о потомках, наследующих соответствующие операции, класс Контейнер связан двусторонней ассоциацией с классом Аналитический_отчет_по_контейнерам «многие ко многим».

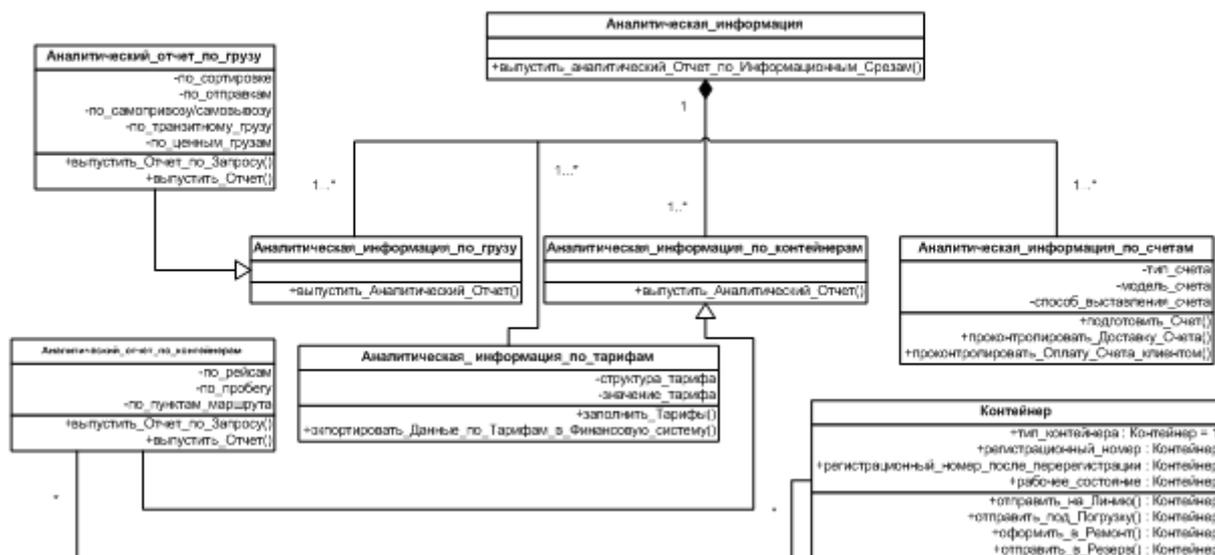


Рис.44. Фрагмент диаграммы классов, описывающей реализацию формирования аналитической отчетности по информационным срезам предприятия N

9. Создайте с помощью команды **Вставка/Создать страницу...** новую страницу с именем *Диаграмма классов учета персонала, обслуживающего перевозки* и установите следующие опции: единицы измерения – миллиметры, тип – передняя, на вкладке **Настройка печати** – Бумага в принтере – Альбомная, Масштаб – 100%, разместить – на одном листе, на вкладке **Размер страницы** – как в принтере, на вкладке **Тени** – опции выберите по своему усмотрению, на вкладке **Макеты и маршруты** – направление – сверху вниз, установите опцию – сдвигать другие фигуры при размещении, интервал между фигурами примите – по умолчанию, примените параметры, нажав кнопку ОК.
10. Установите с помощью кнопки **Масштаб** панели инструментов **Стандартная** масштаб видимости достаточно удобный для работы с листом документа, например 100%.
11. Идентифицируйте классы учета персонала, обслуживающего перевозки на основе перечня бизнес-процессов, описанных в Приложении и создайте диаграмму классов с указанием их имен, атрибутов, операций, например, так как показано на рис. 45.

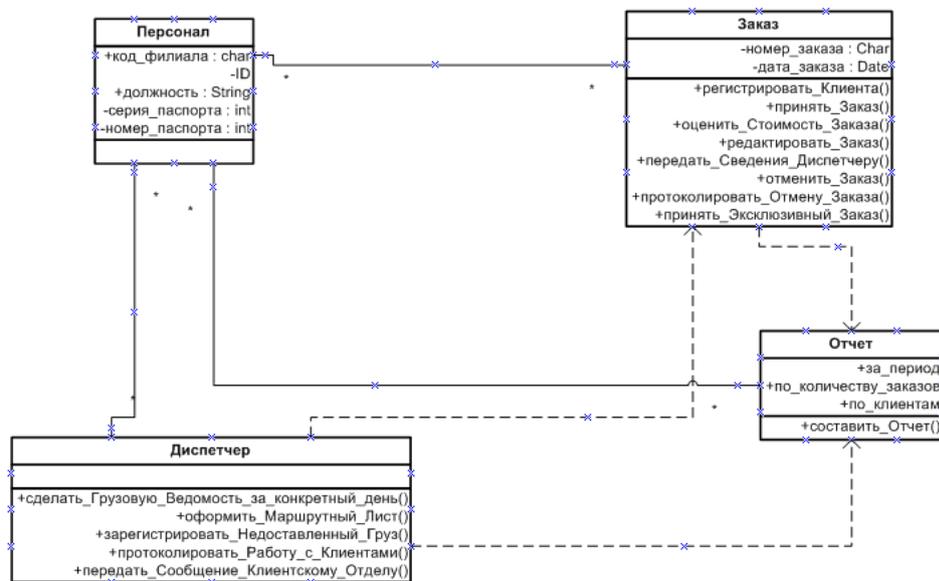


Рис.45.Фрагмент диаграммы классов, описывающей процесс формирования заказов и аналитической отчетности предприятия N

- 12.Создайте с помощью команды **Вставка/Создать страницу...** новую страницу с именем *Обработка груза* и создайте самостоятельно диаграмму классов, выполнив идентификацию классов на основе бизнес-процессов, описанных в Приложении (например, бизнес-процесс Обработка груза).
- 13.Создайте новую страницу с именем *Управление адресами и маршрутами* и создайте самостоятельно диаграмму классов, выполнив идентификацию классов на основе бизнес-процессов, описанных в Приложении (например, бизнес-процесс Управление маршрутом, Управление адресами и маршрутами, Комплексное формирование географических данных).
- 14.Сохраните изменения и завершите работу с приложением MS Office Visio.

Основные бизнес-процессы предприятия N, осуществляющего перевозки в контейнерах с использованием собственного и арендованного парка контейнеров:

Бизнес-процесс	Наполнение бизнес-процесса
Учет транспортных средств, стоящих на балансе предприятия	<p>Тип контейнера (универсальный, специализированный);</p> <p>Форма собственности (собственный, арендованный);</p> <p>Государственный регистрационный номер (возможна перерегистрация с сохранением всех эксплуатационных данных);</p> <p>Состояние контейнера (на линии, в ремонте и т.п.).</p>
Учет персонала, обслуживающего перевозки	<p>ФИО;</p> <p>Табельный номер;</p> <p>Код филиала (для учета распределения персонала по филиалам);</p> <p>Должность;</p> <p>Паспортные данные (серия, номер, дата выдачи, кем выдан);</p> <p>Номер мобильного телефона.</p>
Прием и оформление заказов на перевозку	<p>Регистрация клиентов, которым предоставляются услуги по перевозкам;</p> <p>Прием новых заказов на перевозку груза;</p> <p>Предварительная оценка стоимости перевозки;</p> <p>Передача информации диспетчеру;</p> <p>Поиск, просмотр, редактирование, копирование, возврат заказов на доработку при возникновении проблемных ситуаций, протоколирование процесса общения с клиентом;</p> <p>Отмена заказа и протоколирование причины отмены;</p> <p>Автоматическая генерация регулярных и эксклюзивных заказов.</p>
Отчеты по заказам на перевозку	<p>За период;</p> <p>По количеству заказов;</p> <p>По клиентам.</p>

Обработка груза	<p>Забор груза у отправителя по указанному адресу;</p> <p>Доставка груза получателю;</p> <p>Самостоятельный вывоз груза с терминала;</p> <p>Взвешивание, измерение габаритов, упаковка груза;</p> <p>Маркировка груза бирками для автоматической обработки на терминале;</p> <p>Страхование груза;</p> <p>Перевозка груза регулярными рейсами;</p> <p>Контейнерная перевозка груза, комплектация/раскомплектация, пломбирование контейнера;</p> <p>Оперативное управление проблемным грузом;</p> <p>Справочник типа упаковок;</p> <p>Справочник средств пакетирования;</p> <p>Справочник характера груза;</p> <p>Справочник типов оборудования.</p>
Управление адресами и маршрутами	<p>Справочник стран;</p> <p>Справочник областей/регионов</p> <p>Справочник населенных пунктов;</p> <p>Справочник транспортный узлов;</p> <p>Справочник почтовых индексов;</p> <p>Справочник улиц и строений.</p>
Формирование географических данных	<p>Логистические зоны;</p> <p>Тарифные зоны;</p> <p>Этапы транспортировки груза;</p> <p>Схемы транспортировки.</p>
Управление маршрутом	<p>Создание маршрута, автоматическое/ручное формирование груза на маршрут;</p> <p>Контроль загрузки маршрута и, в случае необходимости, добавление новых контейнеров;</p> <p>Выпуск манифеста и товарно-транспортной накладной для отправляемого груза;</p> <p>Формирование и отправка извещения об отправке груза;</p> <p>Переназначение разгрузки.</p>

<p>Диспетчеризация</p>	<p>Генерация грузовой ведомости на конкретный день; Автоматическое определение номера контейнера для оперативных заказов; Обмен сообщениями между диспетчером и клиентским отделом, протоколирование диалога; Оформление и печать маршрутного листа на перевозку груза; Отмена доставки с указанием причины и инициатора отмены; Регистрация груза, недоставленного получателю; Автоматическое ведение легенды работы с заказами.</p>
<p>Аналитическая информация:</p>	<p>Выпуск аналитических отчетов по различным информационным срезам</p>
<p>по грузу:</p>	<p>Отчет по сортировке; Отчет по отправкам; Отчет по самопривозу/самовывозу; Отчет по транзитному грузу; Отчет по ценным грузам.</p>
<p>по контейнерам:</p>	<p>Отчет по рейсам, пробегу и др.; Отчет по стопам (ежедневно по всем пунктам маршрута следования).</p>
<p>Управление услугами, тарифами и счетами</p>	<p>Справочник основных сервисов, дополнительных сервисов; Справочник семейства услуг, справочник услуг; Справочник структур тарифов и самих тарифов; Справочник налогов; Справочник типов счетов, способов выставления счетов, моделей счетов; заполнение тарифов; Подготовка счетов; Контроль доставки счетов и их оплаты клиентом; Счета-проформы Справочник курса валют; Экспорт данных в финансовую/бухгалтерскую систему (например, в 1С).</p>

Список литературы и ссылки на интернет-ресурсы

1. Буч Г. Язык UML. Руководство пользователя/Г. Буч, Д. Рамбо, Якобсон И. 2-е изд.: Пер. с англ. Мухин Н.- М.: ДМК Пресс, 2007. – 496 с.
2. Леоненко А. Самоучитель.UML. [Электронный ресурс]. Режим доступа: <http://khpi-iip.mipk.kharkiv.edu/library/case/leon/g15/g15.html#4>
3. Пашкевич А.П.Современные технологии программирования/А.П.Пашкевич, Чумаков О.А. Курс лекций. [Электронный ресурс]. Режим доступа: http://www.bsuir.by/m/12_100229_1_58444.pdf
4. Фастовский Э.Г. Лабораторные работы по курсу "Системы автоматизированного проектирования программного обеспечения" [Электронный ресурс]. Режим доступа: <http://khpi-iip.mipk.kharkiv.edu/library/case/lab/labs/lab04.html>
5. [Электронный ресурс]. Режим доступа: http://www.znannya.org/?view=Example_construction_diagram_sequence
6. [Электронный ресурс]. Режим доступа: <http://sun.tsu.ru/mminfo/000063105/inf/01/image/101-105.pdf>