

Лабораторная работа №9

Тема: построение диаграммы классов

Цель: получить навыки построения диаграммы классов

Программное обеспечение: MS Windows, MS Visio, StarUML

Оборудование: IBM PC

Содержание работы

Задание.....	3
Теоретические сведения	4
Методические рекомендации по выполнению работы	5
Пример выполнения задания.....	7
Варианты заданий.....	10
Контрольные вопросы.....	13

Задание

1. Ознакомиться с теоретическими сведениями по теме лабораторной работы.
2. Ознакомиться с методическими рекомендациями и выполнить **задание в соответствии со своим вариантом.**

Номер варианта задания соответствует номеру по списку в журнале.

3. Построить диаграмму классов для системы, проектируемой в предыдущих лабораторных работах.
4. Для каждого класса необходимо задать атрибуты и операции. Каждый класс должен быть *подробно задокументирован* - необходимо задать *текстовое описание самого класса (см. пример) имя самого класса, описания его атрибутов, типов их значений и операций*. Также должны быть определены кванторы видимости для атрибутов и операций.
5. Ответить на контрольные вопросы. Продемонстрировать работу преподавателю.
6. Написать отчет. В отчете должно быть размещено:
 - Титульный лист.
 - Содержание.
 - Тема, цель, ПО, Оборудование к лабораторной работе.
 - Построенная диаграмма классов.
 - Ответы на контрольные вопросы.
 - Вывод по результатам проделанной работы.

Теоретические сведения

Диаграммы классов являются центральным звеном методологии объектно-ориентированных анализа и проектирования.

Диаграмма классов показывает классы и их отношения, тем самым представляя логический аспект проекта. Отдельная диаграмма классов представляет определенный ракурс структуры классов. На стадии анализа диаграммы классов используются, чтобы выделить общие роли и обязанности сущностей, обеспечивающих требуемое поведение системы. На стадии проектирования диаграммы классов используются, чтобы передать структуру классов, формирующих архитектуру системы.

Каждый класс должен иметь имя; если имя слишком длинно, его можно сократить или увеличить сам значок на диаграмме. Имя каждого класса должно быть уникально в содержащем его проекте.

На диаграммах классов изображаются также атрибуты классов, операции и ограничения, которые накладываются на связи между объектами.

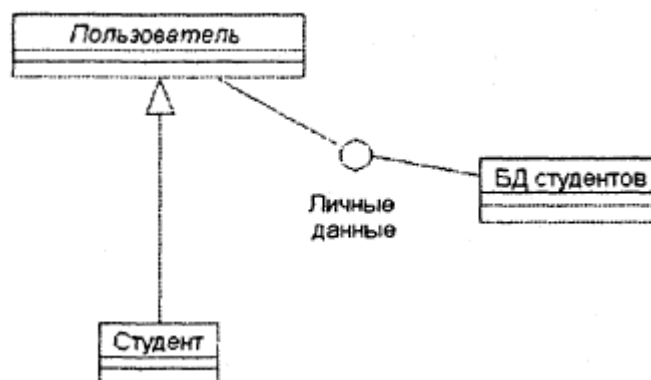
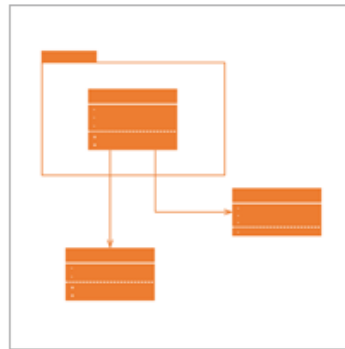


Рис.1. Пример диаграммы классов

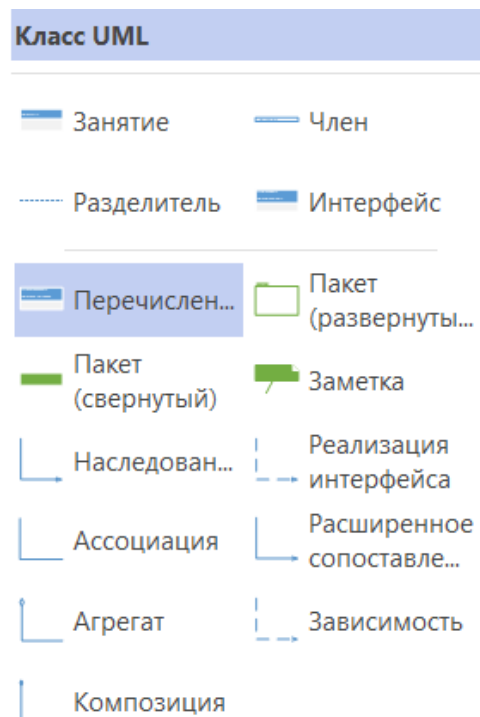
Методические рекомендации по выполнению работы

1. Запустить редактор MS Visio. Выбрать категорию шаблонов «Программы и базы данных», шаблон «Схема модели UML» - «Статическая структура UML», нажать на кнопку «Создать».



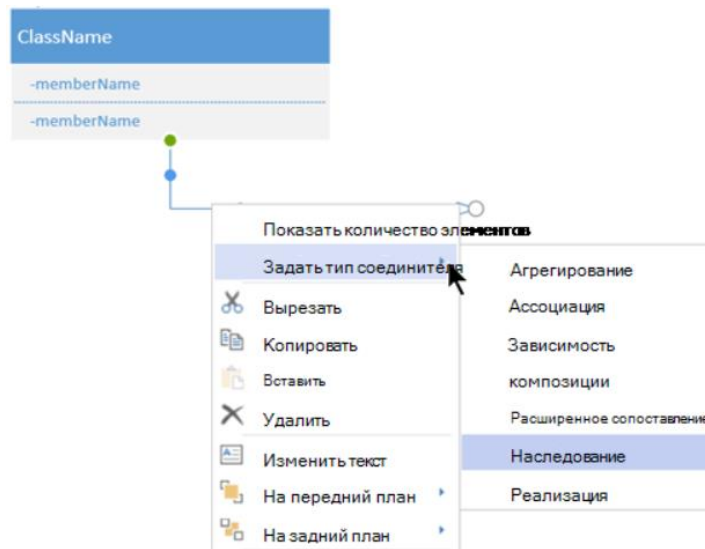
Статическая структура
UML

1. Дать осмысленное имя созданной модели.
2. Ознакомиться с интерфейсом программы и набором команд для построения диаграммы последовательности UML.



3. На вкладке **Вид** установите флажок **Точки соединения**. Этот параметр создает точки соединения, когда вы начинаете подключаться к фигурам.
4. Затем перетащите фигуры, которые вы хотите включить в схему, из окна Фигуры на страницу. Чтобы изменить подписи, дважды щелкните их.
5. Чтобы добавить дополнительные члены в форму класс, перечисление или интерфейс, перетащите фигуру участника с панели фигур на соответствующую фигуру.

6. Соединение двух фигур с помощью соединителей UML для представления связей между фигурами. Чтобы изменить типы отношений, щелкните соединительную линию правой кнопкой мыши. Выберите нужную связь в меню " задать тип соединителя ".



7. Разработать диаграмму классов по своей предметной области. Дать осмысленное имя созданной модели (например: «Система обработки заказов»).
8. Разместить на рабочем листе в необходимом количестве элементы «Класс» и дать им соответствующие названия. Для этого нужно дважды щелкнуть по классу и в появившемся окне свойств в категории «Класс» в поле «Имя» ввести название класса.
9. Задать для классов атрибуты и операции. Для этого также нужно вызвать окно свойств класса и выбрать в нем справа соответствующие категории «Атрибуты» и «Операции». Для атрибутов желательно задать типы данных. Параметры «Видимость», «Кратность» и «Начальное значение» можно не задавать. Для операций также можно не задавать параметры.
10. Указать отношения между классами диаграммы.

Пример выполнения задания

1. Создание диаграммы классов для сценария "Добавить новый заказ" прецедента "Работа с заказом".
2. Диаграммы классов будем рассматривать с концептуальной точки зрения.
3. Создадим новую диаграмму классов и назовем ее "Add New Order". В поле документации запишем для нее следующий текст: "Диаграмма классов для сценария "Добавить новый заказ" прецедента "Работа с заказом"".
4. Заполнение диаграммы начнем с определения классов-сущностей. Рассматриваемый сценарий состоит из:

- самого заказа;
- клиента, который делает заказ;
- комплектующих изделий, которые входят в заказ.

Создадим классы-сущности Order (Заказ), Client (Клиент) и ComponentPart (Комплектующее изделие). Поскольку в один заказ может входить много разных комплектующих изделий, и одно комплектующее изделие может входить во много заказов, то введем еще один класс-сущность OrderItem (Состав заказа). Опишем каждый класс.

Класс *Client*:

Параметр	Значение
Комментарий	Класс, представляющий собой клиента фирмы
Атрибуты	name : String - наименование клиента address : String - адрес клиента phone : String - телефон клиента Все атрибуты имеют модификатор доступа - private
Операции	AddClient() - добавление нового клиента RemoveClient() - удаление существующего клиента GetInfo() - получить информацию о клиенте Все операции имеют модификатор доступа - public

Класс *Order*:

Параметр	Значение
Комментарий	Класс, представляющий собой заказ, который делает клиент
Атрибуты	orderNumber : Integer - номер заказа orderDate : Date - дата оформления заказа orderComplete : Date - дата выполнения заказа Все атрибуты имеют модификатор доступа - private
Операции	Create() - создание нового заказа SetInfo() - занести информацию о заказе GetInfo() - получить информацию о заказе Все операции имеют модификатор доступа - public

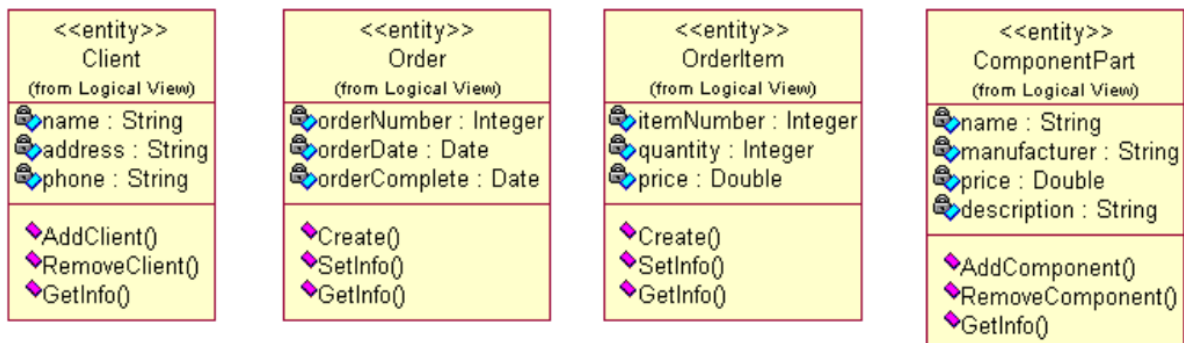
Класс *OrderItem*:

Параметр	Значение
Комментарий	Класс, представляющий собой пункт заказа, который делает клиент
Атрибуты	itemNumber : Integer - номер пункта заказа quantity : Integer - количество комплектующих изделий price : Double - цена за единицу Все атрибуты имеют модификатор доступа - private
Операции	Create() - создание новой строки заказа SetInfo() - занести информацию о строке заказа GetInfo() - получить информацию о строке заказа Все операции имеют модификатор доступа - public

Класс *ComponentPart*:

Параметр	Значение
Комментарий	Класс, представляющий собой комплектующие изделия
Атрибуты	name : String - наименование manufacturer : String - производитель price : Double - цена за единицу description - описание Все атрибуты имеют модификатор доступа - private
Операции	AddComponent() - добавление нового комплектующего изделия RemoveComponent() - удаление комплектующего изделия GetInfo() - получить информацию о комплектующем изделии Все операции имеют модификатор доступа - public

5. Создадим классы-сущности

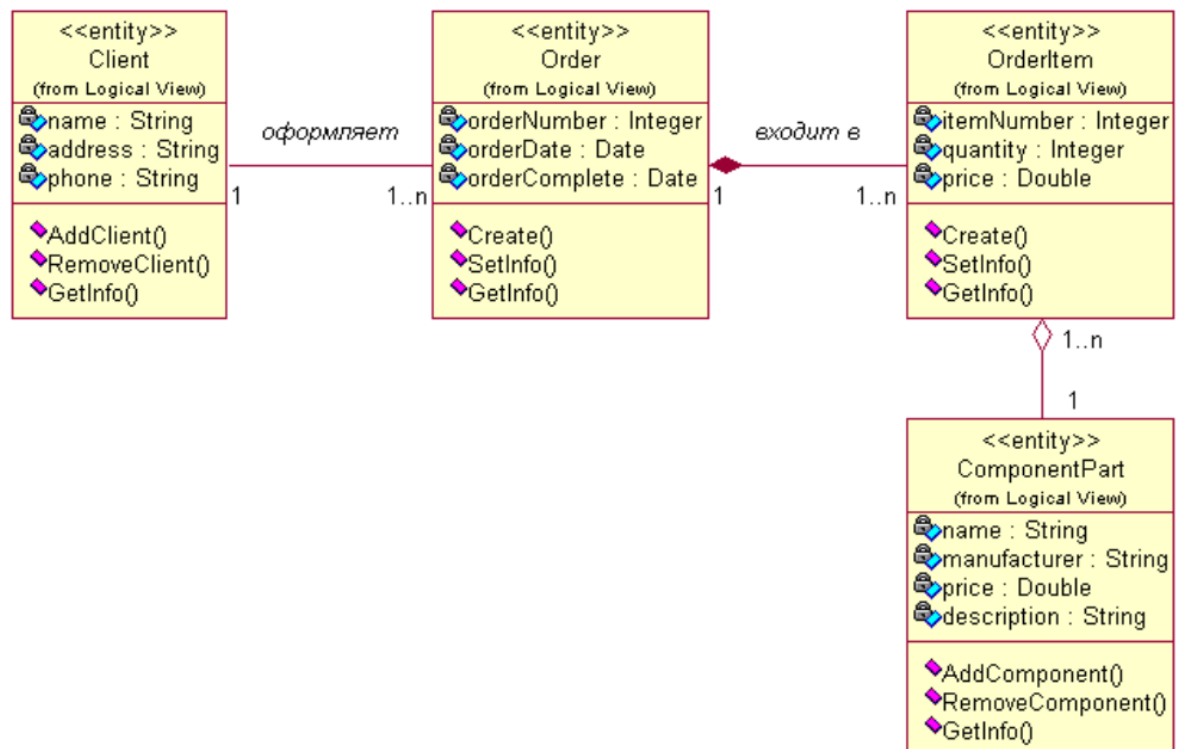


6. Добавим отношения между классами:

- ✓ класс Client и Order - отношение ассоциации, поскольку данные два класса просто связаны друг с другом и никакие другие типы связей здесь применить нельзя. Один клиент может сделать несколько заказов, каждый заказ поступает только от одного клиента, поэтому кратность связи со стороны класса Client - 1, со стороны Order - 1..n;
- ✓ класс Order и OrderItem - отношение композиции, поскольку строка заказа является частью заказа, и без него существовать не может. В один заказ

может входить несколько строк заказа, строка заказа относится только к одному заказу, поэтому кратность связи со стороны Order - 1, со стороны OrderItem - 1..n;

- ✓ класс OrderItem и ComponentPart - отношение агрегации, поскольку комплектующие изделия являются частями строки заказа, но и те, и другие, являются самостоятельными классами. Одно комплектующее изделие может входить во много строк заказа, в одну строку заказа входит только одно комплектующее изделие, поэтому кратность связи со стороны ComponentPart - 1, со стороны OrderItem - 1..n.



Варианты заданий

1. «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

2. «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.

3. «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.

4. «Калькулятор». Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и несколько математических функции.

5. «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.

6. «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.

7. «Автосервис». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, марка автомобиля, вид работы, дата приема заказа и стоимость ремонта. После выполнения работ распечатывается квитанция.

8. «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.

9. «Картотека агентства недвижимости», предназначенный для использования работниками агентства. В базе содержатся сведения о квартирах (количество комнат, этаж, метраж и др.). При поступлении заявки на обмен (куплю, продажу) производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.

10. «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.

11. «Авиакасса», содержащий сведения о наличии свободных мест на авиамаршруты. В базе должны содержаться сведения о номере рейса, экипаже, типе самолета, дате и времени вылета, а также стоимости авиабилетов (разного класса). При поступлении заявки на билеты программа производит поиск подходящего рейса.

12. «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.

13. «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.

14. «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

15. «Поликлиника. Учет больных», содержащий сведения о врачах, больных, их датах и времени приема, а также назначениях. Программный модуль предназначен для организации записи пациента к определенному врачу, конкретной специализации, в отведенное время.

16. «Гостиница. Размещение клиентов», содержащий информацию о гостинице, ее номерах и посетителях. Посетитель может забронировать номер в отеле, соответствующего класса на определенную дату и время.

17. «Страховая компания. Заключение договоров», содержащий сведения о страховой компании, ее страховых агентах, условиях страхования и заключенных договорах с определенным клиентом.

18. «Театр. Продажа билетов.», содержащий сведения о спектаклях, проводимых в театре, его составе, и позволяющей оформить покупку и бронь билета.

19. «Компьютерный сервисный центр.», позволяющий вести учет сервисных работ с компьютерной и оргтехникой.

20. «Салон красоты. Оказание услуг», позволяющий вести запись на оказание услуг и производить учет оказанных услуг.

21. «Химчистка. Учет заказов», позволяющий производить учет оказанных услуг.

22. «Аптека. Поступление и продажа лекарств», позволяющий производить учет поступивших в аптеку лекарств, а также получать сведения о их продажи.

23. «Фитнес-клуб», содержащий информацию о клубе и услугах, которые он предоставляет. А также вести учет клиентов.

24. «Оптовый склад. Заключение договоров с поставщиками».

25. «Мебельный салон. Учет заказов».

Контрольные вопросы

1. Назначение и понятие диаграммы классов?
2. Опишите категории связей между классами.
3. Понятие класса?
4. Что такое атрибуты класса?
5. Что означает видимость элементов класса и как она обозначается на диаграмме?