

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Методические указания к лабораторным работам и практическим
занятиям по дисциплине «Организация и управление научными
исследованиями в области информационных систем»**

Направление подготовки 09.04.02 «Информационные системы и технологии»
Направленность (профиль) «Управление данными»

Ставрополь 2026

Содержание

	Методические указания к лабораторным работам	3
Лабораторная работа №1	Семантические сети	4
Лабораторная работа №2	Продукционная модель представления знаний	6
Лабораторная работа №3	Фреймовая модель представления знаний	10
Лабораторная работа №4	Модель, основанная на нечеткой логике	11
Лабораторная работа №5	Построение базы знаний экспертной системы	12
Лабораторная работа №6	Разработка специальных моделей представления знаний для БЗ и БД и правил для машины вывода	13
Лабораторная работа №7	Разработка БЗ и БД для Пролог – программ решения прикладных задач	22
Практическая работа 1	Разбиение на этапы жизненного цикла (ЖЦ). Диаграмма Ганта	28
Практическая работа 2	Проектирование базы данных	37
Практическая работа 3	Проектирование архитектуры информационной системы (ИС)	46
Практическая работа 4	Проектирование системной архитектуры ИС (состав обеспечивающих подсистем)	51
Практическая работа 5	Разработка технического проекта ИС.	60

Методические указания к лабораторным работам

Для того чтобы получить практические навыки применения систем организации интеллектуального анализа научиться различным подходам к построению базы знаний. Лабораторный практикум преследует достижение следующих целей обучения:

- получение практических навыков разработки баз знаний;
- изучение технологий инженерии знаний;
- получение практических навыков наполнения пустой оболочки экспертной системой знаниями в виде различных моделей представления знаний.

В качестве инструментального средства в методических указаниях используются пустые оболочки экспертных систем, которые ориентированы на различные модели представления знаний.

Методические указания содержат методические рекомендации по выполнению 7 лабораторных работ различной сложности.

Лабораторные работы №1-4 предназначены для получения практических навыков построения базы знаний интеллектуальной информационной системы для предложенной предметной области. Лабораторная работа №5 предполагает выбор студентом предметной области. Лабораторные работы №6 и №7 посвящены языкам логического программирования. Для каждой лабораторной работы указаны необходимый теоретический материал, ее цель, последовательность выполнения, предоставляемая студентом отчетность. Там, где это необходимо, приведены примеры и описана пошаговая последовательность действий.

Для выполнения лабораторного практикума студент должен получить номер варианта задания у преподавателя, если разделение по вариантам предусмотрено данной работой.

Защиты лабораторных работ предполагают определенную подготовку (прочтение соответствующих лекционных тем и выполнения соответствующих заданий на лабораторных занятиях). Отчет по лабораторной работе должен быть представлен в электронном и бумажном виде.

Лабораторная работа №1. Семантические сети

Цель работы: получение знаний, умений и навыков по созданию базы знаний, представляющей собой семантическую сеть.

Последовательность выполнения работы №1

1. Изучите раздел «Семантические сети» теоретического материала (Глава 1, с.18-20).
2. Выберите вариант лабораторной работы.
3. Ответьте на контрольные вопросы.
 - а. Что такое семантическая сеть?
 - б. Какие отношения обычно используются в семантических сетях?
 - с. Что такое бинарная семантическая сеть?
 - д. Какими достоинствами и недостатками обладает семантическая сеть?
4. Напишите отчет к лабораторной работе (приложение 1).

Варианты лабораторной работы №1

Вариант 1

Компания, в которой Вы работаете, получила задание на разработку справочной системы по журналам издательства «Издательство Мечты». Данная компания выпускает различные по целевой аудитории, ценовой категории и объему страниц журналы.

Вам необходимо построить модуль на основе семантической сети, позволяющий определить целевую аудиторию для различных журналов, а также для кого предназначено издание и его стоимость. Ваша задача построить семантическую сеть на основе информации, представленной в таблице 1.

Таблица 1

Название журнала	Основная целевая аудитория	Стоимость одного номера, руб.	Объем страниц журнала	Какая информация представлена в журнале	Возможна ли подписка на журнал
Тюнинг автомобилей	Мужчины	140	170	Современные технологии тюнинга автомобилей	нет
Мода	Женщины	90	90	Новейшие тенденции моды	да

Компьютерные и видео игры	Мужчины и женщины	65	60	Все о компьютерных и видео играх	нет
Рукоделие	Женщины	45	50	Эксклюзивные вещи своими руками	да
Фотография	Мужчины и женщины	100	95	Основы и секреты фотографии	да
Кино и музыка	Мужчины и женщины	30	30	Только актуальная информация и кино и музыке	нет

В построенной семантической сети определить:

1. Какой журнал предоставляет информацию о современных технологиях тюнинга автомобилей?
2. Какие журналы предназначены для мужчин?
3. Какие журналы стоят 100 рублей?
4. На какие журналы можно оформить подписку?

Вариант 2

Ваша задача состоит в создании экспертной системы АСУ предприятия, автоматизирующей контроль за выполнением задач коллективом предприятия. АСУ следует построить в виде семантической сети.

Система должна описывать структуру предприятия, в том числе руководство и структуру отделов.

Система так же должна описывать выполняемые предприятием задания, в том числе:

1. Наименование задания.
2. Сроки его выполнения.
3. Этапы выполнения задания и их очередность.

Для каждого этапа описывается:

1. Отдел, выполняющий этап.
2. Ответственное лицо, обычно – руководитель отдела или подразделения.
3. Сроки начала и окончания этапа.

Предприятие, для которого строится система – ООО «Созвездие»:

Директор: Иванов И.И.

Отдел разработки, нач. отдела – Перов П.П.

В составе отдела разработки:

Бюро постановки задач, нач. бюро – Сидоров С.С.

Бюро программирования, нач. бюро – Брайан Керниган

Бюро сопровождения, нач. бюро – Билл Гейтс

Отдел маркетинга, нач. отдела – Тошико Ямада

Задания в работе:

1. Разработка текстового редактора «Созвездие», этапы – постановка задачи, программирование, продвижение на рынок, поддержка.
2. Разработка Интернет - браузера «Созвездие», этапы – постановка задачи, программирование, продвижение на рынок, поддержка.

В построенной семантической сети определить:

1. Кто является начальником отдела маркетинга?
2. Какие задания выполняет ООО «Созвездие»?
3. Чем занимается Иванов И.И.?
4. Какие сроки выполнения заданы для разработки Интернет - браузера?

Лабораторная работа №2. Продукционная модель представления знаний

Цель работы: получение знаний, умений и навыков по созданию базы знаний, представляющей собой продукционную модель представления знаний.

Последовательность выполнения работы №2

1. Изучите раздел «Продукционная модель представления знаний» теоретического материала (Глава 1, с.16-18).
2. Изучите задание к лабораторной работе.
3. Ответьте на контрольные вопросы.
 - а. Как представлены знания в продукционной модели представления знаний?
 - б. Что такое консеквент?
 - с. Какие части имеет продукционная система?
 - д. Для чего нужна рабочая память?
 - е. Какими достоинствами и недостатками обладает продукционная модель представления знаний?
4. Напишите отчет к лабораторной работе (приложение 1).

Варианты лабораторной работы №2

Вариант 1

На рынке фотоаппаратов существует огромное множество различных фотокамер, способных удовлетворить почти любого потребителя. Выбор камеры

- непростое дело, и не всякий покупатель способен сам выбрать себе подходящий фотоаппарат. В фирменных салонах по продаже фотокамер не хватает консультантов, которые могли бы квалифицированно помочь клиенту. Ваша задача - разработать продукционную систему, помогающую покупателю в его непростом выборе.

Таким образом, задание на лабораторную работу состоит в наполнении пустой оболочки экспертной системы знаниями из таблицы 2.

Большинство покупателей склонны пере- или недооценивать свои запросы (и способности), поэтому вовсе необязательно прямо спрашивать у человека, любитель он или профессионал. Такие вещи лучше выяснять, спросив у человека, например, собирается ли он покупать аппарат впервые (новичок), снимает для семейного альбома (любитель), печатается ли он в каких-нибудь изданиях (профессионал).

Новичок редко может сказать, нужна ли ему сменная оптика в камере. Необходимо узнать, собирается ли он снимать портреты и пейзажи – в этом случае она ему, возможно, понадобится, или только фотографии вечеринок и.т.д. (в этом случае сменная оптика не нужна).

Мало кто так же способен внятно сказать, нужны ли ему художественные режимы (если он не занимается фотографией профессионально). Это лучше всего выяснить, спросив, например, что такое глубина резкости (предложите клиенту несколько вариантов на выбор, правильный ответ – диапазон расстояний, в котором все объекты выходят резкими).

Цена определяется исходя из максимальной суммы, которую готов потратить клиент.

Новые правила следует внести в часть программного кода, отвечающего за правила, то есть в часть, которая начинается со служебных слов:

topic camera.

set_number_of_values (camera,1).

Пример правила со служебным синтаксисом приведен ниже. В данном правиле класс пользователя - новичок, сменная оптика не нужна, художественные режимы отсутствуют, и низкая цена, этим характеристикам отвечает камера Riva Zoom 100. Этому правилу соответствует первая строка таблицы 2.

if ?class is new and ?opt is no and ?hud_res is no and ?price is low
then camera is 'Riva Zoom 100'.

Таблица 2

Класс	Сменная оптика	Художественные режимы	Цена	Камера
Новичок	нет	нет	Небольшая	Riva Zoom100
Новичок	нет	нет	Средняя	Riva Zoom500
Новичок	нет	нет	Высокая	Riva Zoom550
Новичок	есть	нет	Небольшая	Dynax 404si
Новичок	есть	нет	Средняя	Dynax 406si
Новичок	есть	нет	Высокая	Dynax 408si
Новичок	нет	есть	Небольшая	Nikon 1200

Новичок	нет	есть	Средняя	Kodak 835AF
Новичок	нет	есть	Высокая	Nikon 1201
Новичок	есть	есть	Небольшая	Rekam DH300
Новичок	есть	есть	Средняя	Rekam Max 3
Новичок	есть	есть	Высокая	Rekam Mega 110
Любитель	нет	нет	Небольшая	Konica POP MINI
Любитель	нет	нет	Средняя	Konica POP EFP-8
Любитель	нет	нет	Высокая	Konica Centuria-10
Любитель	нет	есть	Небольшая	Konica POP BF-8
Любитель	нет	есть	Средняя	Konica Centuria-20
Любитель	нет	есть	Высокая	Konica POP ST
Любитель	есть	нет	Небольшая	Konica POP ST 80
Любитель	есть	нет	Средняя	Samsung FINO 15 SE
Любитель	есть	нет	Высокая	Samsung FINO 20 SE
Любитель	есть	есть	Небольшая	Dynax 505si
Любитель	есть	есть	Средняя	Samsung VEGA 170
Любитель	есть	есть	Высокая	Pentax ESPIO 200
Профессионал	нет	нет	Небольшая	нет
Профессионал	нет	нет	Средняя	нет
Профессионал	нет	нет	Высокая	нет
Профессионал	нет	есть	Небольшая	нет
Профессионал	нет	есть	Средняя	нет
Профессионал	нет	есть	Высокая	нет
Профессионал	есть	нет	Небольшая	Konica Centuria-60Z
Профессионал	есть	нет	Средняя	Konica Centuria-70Z
Профессионал	есть	нет	Высокая	Olympus ZOOM 80
Профессионал	есть	есть	Небольшая	Olympus is-300
Профессионал	есть	есть	Средняя	Dynax 7
Профессионал	есть	есть	Высокая	Dynax 9

Вариант 2

Вы работаете в крупной компании производящей спецодежду. Ваша компания производит более 10 000 наименований различной спецодежды. Отдел программирования, в котором Вы работаете, получил задание на разработку системы, помогающей покупателю определиться с выбором. Покупатель может выбрать защищающую одежду для головы, рук или корпуса. Материал, из которого изготовлена одежда, может быть как натуральным, так и искусственным. Также в наличии имеются товары различной ценовой категории. Ваша задача: наполнить пустую оболочку экспертной системы знаниями из таблицы 3. Цена определяется исходя из максимальной суммы, которую готов потратить клиент.

Таблица 3

Защита	Материал	Серия	Цена	наименование
голова	натуральный	«Зима»	Небольшая	Каска «Труд»
голова	искусственный	«Омон»	Средняя	Каска «Байкал»

голова	искусственный	«Зима»	Высокая	Каска «СОМ 3-53 Люкс
голова	искусственный	«Рабочий»	Небольшая	Каска «СуперБосс»
голова	натуральный	«Комфорт»	Средняя	Каска «Эйрвинг»
голова	искусственный	«Зима»	Высокая	Каска «Вигард»
голова	натуральный	«Зима»	Небольшая	Каска «Металлург»
голова	искусственный	«Омон»	Средняя	Каска «Сварщик»
голова	натуральный	«Омон»	Высокая	Каска «Шахтер»
голова	натуральный	«Комфорт»	Небольшая	Каска «Термолюкс»
голова	искусственный	«Рабочий»	Средняя	Каска «Супер Вигард»
голова	искусственный	«Рабочий»	Высокая	Каска «Люкс»
руки	натуральный	«Зима»	Небольшая	Рукавицы утепленные гладкокрашенные
руки	искусственный	«Зима»	Средняя	Рукавицы нагольные
руки	натуральный	«Зима»	Высокая	Рукавицы меховые
руки	натуральный	«Омон»	Небольшая	Рукавицы крытые
руки	натуральный	«Омон»	Средняя	Рукавицы морозостойкие
руки	искусственный	«Рабочий»	Высокая	Рукавицы с двойным наладонником
руки	искусственный	«Рабочий»	Небольшая	Рукавицы джинсовые
руки	искусственный	«Зима»	Средняя	Рукавицы брезентовые
руки	натуральный	«Комфорт»	Высокая	Рукавицы крагиспилковые
руки	искусственный	«Рабочий»	Небольшая	Рукавицы суконные
руки	искусственный	«Комфорт»	Средняя	Вачеги для металлурга
руки	искусственный	«Комфорт»	Высокая	Вачеги цельноспилковые
корпус	искусственный	«Омон»	Небольшая	Костюм мужской «Вектор»
корпус	искусственный	«Омон»	Средняя	Костюм мужской «Вектор+»
корпус	искусственный	«Омон»	Высокая	Костюм мужской «Амулет»
корпус	искусственный	«Зима»	Небольшая	Костюм мужской «Диксон»
корпус	искусственный	«Зима»	Средняя	Костюм мужской «Зимник»
корпус	искусственный	«Рабочий»	Высокая	Костюм мужской «Модуль»
корпус	натуральный	«Комфорт»	Небольшая	Костюм мужской «Стим»
корпус	натуральный	«Комфорт»	Средняя	Костюм мужской «Рейсер»
корпус	натуральный	«Зима»	Высокая	Костюм мужской «Тайшет»
корпус	натуральный	«Зима»	Небольшая	Костюм мужской «Буран КМФ»
корпус	натуральный	«Комфорт»	Средняя	Костюм мужской «Аляска»

корпус	натуральный	«Рабочий»	Высокая	Костюм мужской «Легенда»
--------	-------------	-----------	---------	--------------------------

При наполнении базы правил обратите внимание на синтаксис. Не поставленная точка грозит неправильной работой программой.

Лабораторная работа №3. Фреймовая модель представления знаний

Цель работы: получение знаний, умений и навыков по созданию базы знаний, представляющей собой фреймовую модель представления знаний.

Последовательность выполнения работы №3

1. Изучите раздел «Фреймовая модель представления знаний» теоретического материала (Глава 1, с.21-25).
2. Выберите вариант лабораторной работы.
3. Ответьте на контрольные вопросы.
 - а. Что такое фрейм, слот?
 - б. Что такое механизм наследования? Для чего он нужен?
 - с. Какие указатели наследования Вы знаете?
 - д. Какие свойства фреймов Вы знаете?
 - е. Что такое демон?
 - ф. Какими достоинствами и недостатками обладают фреймы?
4. Напишите отчет к лабораторной работе (приложение 1).

Варианты лабораторной работы №3

Вариант 1

Вы – офицер штаба армии. Вам поступило задание разработать экспертную систему, предназначенную для выбора десантного средства для тактических операций.

В наличии имеются следующие боевые транспортные аппараты:

Таблица 4

Наименование	Способ передвижения	Вместимость, десант	Вооружение	Защита
БМП-3	Гусеницы	7 чел	пушка 100 мм	броня
Ми-24	Вертолет	6 чел	пушка 30 мм	сталь
БТР-50Б	Колеса	8 чел	пулемет 14.5 мм	броня
Ми-8	Вертолет	12 чел	нет	нет

Необходимо создать систему, занимающую наименьший объем памяти, позволяющую выбирать боевые средства тактической доставки десанта на основе заданных требований к ним.

Таким образом, задачей выполнения лабораторной работы является создание системы фреймов из знаний, представленных в таблице 4.

В созданной системе найти десантные средства, удовлетворяющие следующим требованиям:

1. вертолет, способный перевозить не менее 6 человек десанта над полем боя без поддержки наземных средств;
2. наземное средство доставки не менее 6 десантников в тыл противника при активном противодействии бронетанковых войск противника;
3. средство доставки не менее 6 человек десанта в тыл противника при наличии противодействия противника.

Вариант 2

В лабораторной работе №1 «Семантические сети», вариант 1 представлена задача разработки семантической сети для небольшого издательства. Ваша задача представить знания, описанные в таблице 1 в виде фреймов.

В построенной системе фреймов определить:

1. Какой журнал предоставляет информацию о современных технологиях тюнинга автомобилей?
2. Какие журналы предназначены для мужчин?
3. Какие журналы стоят 100 рублей?
4. На какие журналы можно оформить подписку?
5. Найти все журналы издательства.
6. Найти все журналы, которые стоят не более 150 рублей.

Лабораторная работа №4. Модель, основанная на нечеткой логике

Цель работы: получение знаний, умений и навыков по созданию базы нечетких знаний.

Последовательность выполнения работы №4

1. Изучите раздел «Модель, основанная на нечетких знаниях» теоретического материала (Глава 1, с. 30-45).
2. Выберите вариант лабораторной работы.
3. Ответьте на контрольные вопросы.
 - a. Что такое лингвистическая переменная, нечеткое множество?
 - b. Какое нечеткое число называется треугольным, трапецеидальным, в чем их различие?
 - c. Какие операции над нечеткими множествами Вы знаете?
 - d. Как работает нечеткий логический контроллер?
4. Напишите отчет к лабораторной работе (приложение 1).

Варианты лабораторной работы №4

Вариант 1

Компания, в которой Вы работаете, получила задание на разработку нечеткого управляющего контроллера для бытового кондиционера. Контроллер должен управлять работой главного вентилятора.

На вход контроллера поступают сигналы от датчиков влажности и температуры. На основе показаний этих датчиков Вам необходимо построить набор правил и нечеткие множества, определяющие лексические термы для этих правил.

Вам необходимо так же выбрать метод импликации и дефаззификации полученного в результате работы системы нечеткого множества.

Вариант 2

Вы сотрудник бюро автоматизации коммерческой поликлиники. Вам необходимо разработать нечеткий логический контроллер для выбора подходящего рациона питания. В качестве входных характеристик выступают рост человека и его вес, в качестве выходных параметров – калорийность диеты.

Вам необходимо так же выбрать метод импликации и дефаззификации полученного в результате работы системы нечеткого множества.

Лабораторная работа №5. Построение базы знаний экспертной системы

Цель работы: получение знаний, умений и навыков по созданию базы знаний экспертной системы для выбранной предметной области.

Последовательность выполнения работы №5

1. Изучите Главу 2 теоретического материала.
2. Выберите вариант лабораторной работы.
3. Составьте базу знаний для выбранной предметной области. Внесите полученную базу знаний в программную оболочку экспертной системы. Изучите компонент объяснения вывода на знаниях.
4. Ответьте на контрольные вопросы.
 - a. Что такое экспертная система?
 - b. Какие этапы разработки экспертной системы Вы знаете?
 - c. Какие специалисты принимают участие в разработке экспертной системы?
 - d. Какими практическими способами знания могут быть получены?
5. Напишите отчет к лабораторной работе (приложение 1).

Варианты лабораторной работы №5

Вариант 1

Построение базы знаний экспертной системы «Подбор тарифного плана оператора мобильной связи Мегафон-Сибирь».

Вариант 2

Построение базы знаний экспертной системы «Расчет приблизительной продолжительности жизни человека».

Вариант 3

Построение базы знаний экспертной системы «Выбор автосигнализации».

Вариант для выполнения лабораторной работы может быть предложен студентом и должен быть утвержден преподавателем.

Лабораторная работа №6. Разработка специальных моделей представления знаний для БЗ и БД и правил для машины вывода

Порядок работ и их виды

Введение с указанием предметной области.

В разделе **“I. Идентификация”** постановка (формулировка) проблемы, цели и задачи.

В разделе **“II.. Концептуализация”** нужно представить следующие результаты разработки Содержательной и Концептуальной моделей.

Выделение основных сущностей предметной области

Выявление связей (соответствий, в частности, отношений) между сущностями.

Классификация отношений. Описание сущностей. Разбиение задачи на подзадачи.

Выделение основных сущностей предметной области.

Результатом является полный список сущностей (объектов, концептов). В списке необходимо выделить основные сущности.

Оформление результатов: текст (список).

Выявление связей между сущностями

Результатом является граф, вершины которого обозначают сущности, а дуги - отношения между ними (в случае функциональной семантической сети - двудольного графа - отношения заносятся в соответствующие вершины). Возможно исключение некоторых сущностей, включенных на этапе разработки Содержательной модели.

Оформление результатов: текст, рисунки с пояснительным текстом (при необходимости).

Классификация отношений

Основные типы отношений: элемент-класс, род-вид; часть-целое; объект-атрибут (свойство, признак) объекта; объект-объект и т.д. Расширение и/или детализация данного списка разрешается и выражается в более высокой оценке работы.

Результатом является:

а) список отношений и их классификация (элемент-класс, род-вид; часть-целое; и т.д.);

б) классификация дуг (род-вид; часть-целое; и т.д.) и граф с помеченными дугами.

Оформление результатов: текст, содержащий список отношений и их классификацию. В тексте на рисунках графов помечены дуги.

Описание сущностей.

Каждой из сущностей устанавливается в соответствие ее краткое определение и описание (1-3 предложения), с указанием ссылок на литературу.

Результатом является совокупность описаний сущностей и соответствующих вершин графа, полученного на предыдущем этапе.

Оформление результатов: текст на бумажном носителе с рисунками, аккуратно выполненный от руки, или текстовый файл в формате Word.

Разбиение задачи на подзадачи

Результат: дерево подзадач, дерево решений, дерево состояний, И-ИЛИ-граф, и т.п.

Оформление результатов. Соответствующие графы (деревья и т.п.)

В разделе “**III. Формализация**” нужно представить следующие результаты.

Формальные модели представления (как знаний или данных, фактов, правил) тех сущностей и отношений, которые выделены на этапе Концептуализации в свете сформулированных целей и задач на этапе Идентификация, а именно:

логические МПЗ (предикатные и т.д.), логико-математические МПЗ (функциональные и т.д.), продукционные МПЗ, логико - лингвистические МПЗ,

реляционные МПЗ, сетевые (в частности, иерархические) МПЗ, семантические МПЗ (семантические сети, в том числе функциональные),

фреймовые МПЗ, информационные МПЗ (фрагменты интерфейса или диалога пользователя с ИИС как с ПС - программной системой), онтологические МПЗ (и МПДанных),

алгоритмические МПЗ (алгоритм, блок-схема), элементы (фрагменты) Пролог-программы (программный код).

Разработка элементов Машины Выводов и Механизма Выводов (стратегий вывода, принятия решений и т.п.) на основе результатов, полученных при выполнении предыдущих этапов и их подэтапов.

Результаты. Перечисленные выше МПЗ, “встроенные” в соответствующие части ИИС (и/или в Пролог-программу): БД, БП, БФ, МВ, Интерфейс или в ВС (вычислительную подсистему разрабатываемой Вами ИИС).

Оформление результатов: текст с рисунками на бумажном носителе, аккуратно выполненный от руки, или текстовый файл (в формате Word) с рисунками.

Порядок оформления лабораторных работ №6,7

Результаты (включая рисунки) оформляются на листах формата А4, можно от руки, разборчиво (соблюдение ГОСТа на чертеж не требуется).

Отчет по лабораторной работе

должен содержать следующие разделы:

1. Введение, предметная область, постановка задачи
2. Разделы I, II, III.

Для защиты ЛР необходимо:

- иметь оформленный текст с выполненной работой и/или файлы с выполненной работой;
- разместить файлы на носителе (дискете и т.д.) и сдать эти файлы (а не дискеты и т.п.).

Последовательность выполнения работы № 6

- 1) Выбрать свой вариант лабораторной работе 1 и разработать МПЗ.
- 2) Представить полученные Вами модели знаний, фактов, правил, данных в соответствующих частях Пролог - программы.

При необходимости обратиться к Пояснениям к лабораторной работе.

Варианты лабораторной работы №6

№ 1. На модели ромба методом прямого поиска получить план решения следующей задачи: зная периметр и угол γ , найти площадь ромба. Проделать то же самое методом обратного поиска.

№ 2. На модели ромба методом прямого поиска получить план решения следующей задачи: зная периметр и одну из диагоналей, найти площадь ромба и угол γ . Прodelать то же самое методом обратного поиска.

№ 3. На модели прямоугольного треугольника методом прямого поиска получить план решения следующей задачи: зная радиус описанной окружности и один из углов, прилежающих к гипотенузе, найти площадь треугольника. Прodelать то же самое методом обратного поиска,

№ 4. На модели прямоугольного треугольника методом прямого поиска получить план решения следующей задачи: зная площадь треугольника и один из катетов, найти гипотенузу и углы треугольника. Прodelать то же самое методом обратного поиска.

№ 5. На модели треугольника методом прямого поиска получить план решения следующей задачи: зная стороны треугольника, найти его углы. Прodelать то же самое методом обратного поиска.

№ 6. На модели треугольника методом прямого поиска получить план решения следующей задачи: зная две стороны треугольника и угол между ними, найти его третью сторону и два других угла. Прodelать то же самое методом обратного поиска.

№ 7. На модели треугольника методом прямого поиска получить план решения следующей задачи: зная сторону треугольника и два прилежающих к ней угла, найти две его другие стороны и третий угол. Прodelать то же самое методом обратного поиска.

№ 8. На модели квадрата методом прямого поиска получить план решения следующей задачи: зная периметр, найти площадь квадрата. Прodelать то же самое методом обратного поиска.

№ 9. На модели квадрата методом прямого поиска получить план решения следующей задачи: зная периметр, найти площадь квадрата. Прodelать то же самое методом обратного поиска.

№ 10. На модели трапеции методом прямого поиска получить план решения следующей задачи: зная длины параллельных сторон и высоту между ними, найти площадь трапеции. Прodelать то же самое методом обратного поиска.

№ 11. На модели параллелограмма методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 12. На модели прямоугольника методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 13. На модели окружности методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 14. На модели прямоугольника методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 15. На модели равнобедренной трапеции методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 16. На модели прямоугольного треугольника методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 17. На модели равностороннего треугольника методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 18. На модели равнобедренного треугольника методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 19. На модели эллипса методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

№ 20. На модели четырехугольника методом прямого поиска получить план решения следующей задачи: зная, найти Прodelать то же самое методом обратного поиска.

В задачах 11-20 следует самостоятельно указать те сущности, которые позволят найти (вычислить) указанные Вами искомые величины.

Пояснения к лабораторной работе №6

1. Механизм вывода на графах

Модели знаний являются основой компьютерных системах искусственного интеллекта. Назначение таких систем состоит в поиске решений задач, ответов на вопросы (запросы) пользователя, консультациях и т.п. В каждом таком случае требуется получать новые знания на основе уже имеющихся в БЗ. Способ получения новых знаний называется механизмом вывода. То есть, механизм вывода - это способ (в частном случае, алгоритм) поиска решения задачи на основе исходных данных и моделей знаний, представленных в БЗ.

Пример 1. Пусть нужно получить функциональную семантическую сеть - ФСС, т.е. получить семантическую модель знаний о ромбе в форме двудольного графа.

Ромб имеет 7 характеристик: длину стороны a , острый угол α , тупой угол γ , площадь S , периметр P и диагонали d_1 и d_2 .

Эти величины связаны следующими формулами:

$$\alpha + \gamma = 180, P = 4a, S = a^2 \sin \alpha, S = d_1 d_2 / 2, d_1^2 + d_2^2 = 4a^2.$$

Двудольный граф - разновидность графа, в котором присутствуют вершины только двух типов (двух долей), назовем их «черными» и «белыми». «Черным» вершинам ставятся в соответствие объекты (понятия), «белым» - связи между ними. Любая дуга на таком графе проходит между двумя вершинами только разных цветов (рисунок 1).

Построим двудольный граф с семью «черными» вершинами ($a, \alpha, \gamma, S, P, d_1, d_2$) и пятью «белыми» (для каждой из пяти формул).

Отметим, что связи в ФСС - функциональной семантической сети заданы формулами. Формула определяет способ вычисления входящей в нее величины через остальные. Например, из формулы F3: $S = a^2 \sin \alpha$

можно вычислить S , если заданы a и α ;

можно вычислить a , если заданы α и S ;

можно вычислить α , если заданы a и S .

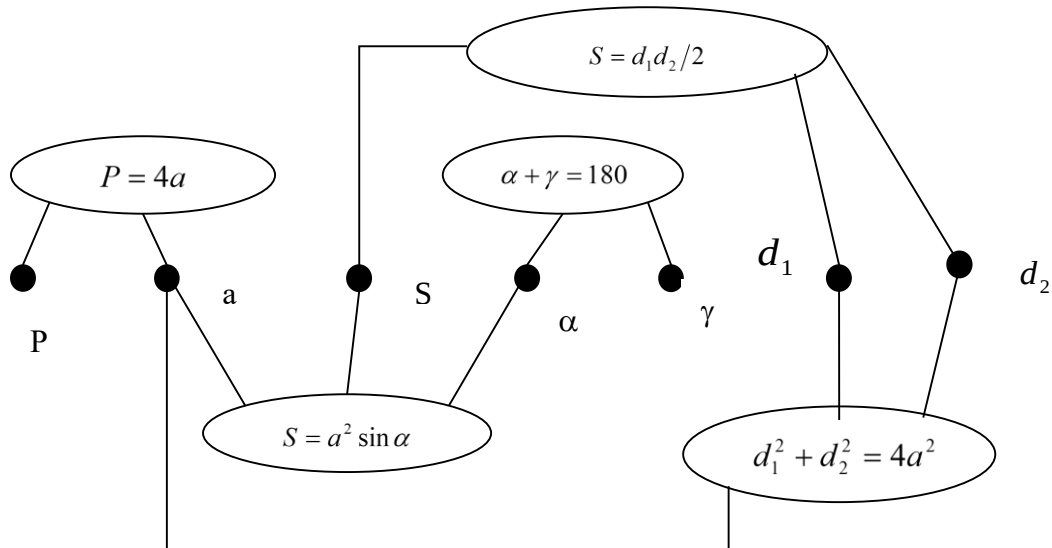


Рис. 6.1. Функциональная семантическая сеть

Пример 2. Рассмотрим процесс решения задачи на основе двудольного графа на рис. 6.1, представляющего модель знаний о ромбе в форме ФСС. Введем обозначения для формул, связывающих характеристики ромба:

F1: $\alpha + \gamma = 180$,

F2: $P = 4a$,

F3: $S = a^2 \sin \alpha$,

F4: $S = d_1 d_2 / 2$

F5: $d_1^2 + d_2^2 = 4a^2$.

Дано (входные данные): длины диагоналей d_1 и d_2 .

Требуется определить (выходные искомые величины): углы ромба α и γ .

Решение с помощью функциональной семантической сети сводится к поиску путей на графе, позволяющих от вершин с исходными данными добраться до вершин с искомыми величинами, и наоборот. Такой поиск можно запрограммировать.

Стратегии вывода. Вывод новых знаний идет в 2-х направлениях:

от известных данных к цели (результатам) и от цели к известным данным. Первый способ также называется прямой волной, прямым поиском, прямой стратегией вывода, второй - обратной волной, обратным поиском, обратной стратегией вывода.

Для данного примера прямой поиск пойдет от известных величин - d_1 и

d_2 - к искомым - α и γ . Обратный поиск - от искоемых α и γ к заданным d_1 и d_2 .

Ход и путь вывода. Последовательность шагов прямого поиска отобразим в виде таблицы 5.

Таблица 5

Прямой поиск					
№ шага	Что известно	Что требуется	Какие связи можно применить	Какую связь применим	Что найдем
1	d_1, d_2	α, γ	F5, F4	F5	a
2	d_1, d_2, a	α, γ	F4, F2	F4	S
3	d_1, d_2, a, S	α, γ	F2, F3	F2	P
4	d_1, d_2, a, S, P	α, γ	F3	F3	α
5	$d_1, d_2, a, S, P, \alpha$	γ	F1	F1	γ
6	$d_1, d_2, a, S, P, \alpha, \gamma$				

1) Для начала проверим, не принадлежат ли искомые данные к множеству уже известных. В данном случае это не так: известны d_1 и d_2 , требуется найти a и γ . Начнем поиск. Зная диагонали, имеем достаточно данных для того, чтобы применить связи F5 и F4. Выберем одну из них случайным образом, например F5. Преобразовав соответствующее математическое выражение, найдем a - длину стороны ромба.

2) Множество известных данных расширилось, поэтому в начале второго шага снова проверим, а не входят ли искомые величины в множество уже известных? Нет. Продолжим поиск. Известные величины - d_1, d_2, a - позволяют применить три связи: F5, F4, F2. Применять связь F5 нет смысла: ее уже использовали и ничего нового она не даст. Остаются F4 и F2. Выберем случайным образом одну из них, например F4. Применение этой связи позволит отыскать площадь ромба S .

3) Третий шаг опять начнем с проверки, не получен ли результат? Поскольку это не так, оценим достигнутое. Четыре известные характеристики ромба позволяют применить две связи F2 и F3 (не считая уже отработанных F4 и F5). Выберем случайным образом одну из них, скажем F2. Применив ее, найдем длину периметра P . Замечание. Здесь описан формальный алгоритм, пригодный для программирования. Человек не стал бы делать этот шаг, для него совершенно лишний. А вот программа его сделать вполне может, поскольку здесь в ней работает механизм случайного выбора.

4) Искомые величины все еще не найдены, поэтому продолжим поиск. Единственная связь, которой до сих пор не воспользовались и которая готова к применению, это связь F3. Воспользовавшись ей, найдем угол α .

5) Одно из искомых значений найдено. Осталась одна неопределенная величина и единственная неиспользованная связь - F1. Применим F1 и найдем с ее помощью второй угол - угол γ .

6) Сравнение множества известных данных и множества искомых данных показывает, что все требуемые данные найдены. Значит, задача решена.

Поиск закончился удачно. Но могло быть и не так: возможна ситуация, когда на очередном шаге не окажется ни одной связи, способной вычислить

что-то новое и готовой к использованию, а цель еще не достигнута. Это значит, что для решения задачи недостаточно исходных данных.

Теперь рассмотрим механизм обратного поиска. Если прямой поиск пути решения задачи можно совместить с самим процессом решения, то при обратном поиске сначала строится план, а затем по этому плану выстраивается решение.

Ход и путь вывода. Последовательность шагов обратного поиска отобразим в виде таблицы 6.

Таблица 6

Обратный поиск						
№ шага	Что известно	Что требуется	Применение каких связей даст искомые величины	Что будем искать	Какую связь применим	Что нужно для применения этой связи
1	d1, d2	α, γ	F1, F3	γ	F1	α
2	d1, d2	α	F3	α	F3	a, S
3	d1, d2	a, S	F2, F4, F5	a	F5	d1, d2
4	d1, d2	S, d1, d2	F4	S	F4	d1, d2
5	d1, d2	d1, d2				

1) Начинаем поиск от вершин двудольного графа, содержащих углы ромба (цели). Вычислить их можно с помощью связей F1 и F3. Случайным образом выберем одну из искомых переменных и одну из связей, которая способна ее вычислить. Пусть это будет γ и F1. Если удастся использовать эту связь, то сможем найти целевую величину γ . Но сделать это можно только в том случае, если будет известна величина α . Запланируем применение связи F1. В множестве искомых данных оставим α .

2) Проверим, все ли искомые данные есть в множестве известных? Не все. В множестве целевых данных один единственный элемент - α , и вычисление его возможно с помощью единственной связи - F3. Для применения этой связи нужны a и S. Планируем применение связи F3, не удаляя из целевого множества a, добавляем в него S.

3) Все ли искомые данные известны? Нет. Выберем случайно одну из целевых переменных, например a, и одну из связей для ее вычисления, например F5. Для ее применения нужны d1 и d2. Планируем применение связи F5, вычеркиваем a из множества искомых переменных, дописываем в это множество d1 и d2.

4) В множестве искомых данных три переменные, но две из них входят в множество исходных данных. Искать надо только одну переменную - S. Для этого есть связь F4. Планируем применение этой связи. Для ее использования нужно знать значения переменных d1 и d2. Запишем их в целевое множество, удалив оттуда переменную S.

5) Все требуемые данные известны. Значит, если начать с этих данных и последовательно применять запланированные связи, то в конце получим решение поставленной задачи. Планирование закончено.

Полученный план определяет следующий порядок применения формул

для решения задачи: $F4 \rightarrow F5 \rightarrow F3 \rightarrow F1$. Сравните этот план с тем, что получен методом прямого поиска. Отсюда очевидна неоднозначность пути решения задачи.

Следует разработать логические и продукционные модели, фреймовая модель (модель нужно представить в форме таблицы), логико - лингвистическая модель, реляционная модель, семантическая сеть.

Функциональная семантическая сеть (ФСС). Вычислительную модель нужно представлять графически с помощью графического языка в форме двудольного графа с использованием соответствующих формул.

Вычислительная модель любой вычислительной задачи имеет следующую форму:

ЗНАЯ М ВЫЧИСЛИТЬ $Y_1, Y_2, \dots, Y_N$ ПО $X_1, X_2, \dots, X_K$.

Здесь $M, Y_1, Y_2, \dots, Y_N, X_1, X_2, \dots, X_K$ - величины, которые имеют смысл, определяемый их вхождением в задачу - в ее постановку и условия; значит $X_1, X_2, \dots, X_K$ - входные данные, $Y_1, Y_2, \dots, Y_N$ - выходные искомые величины, значения их требуется вычислить, найти, определить. M - величина, которая выражает условие задачи. Идентификаторы ЗНАЯ, ВЫЧИСЛИТЬ и ПО имеют фиксированный смысл и служат для разделения известных данных $X_1, X_2, \dots, X_K$ от искомых величин $Y_1, Y_2, \dots, Y_N$.

Как представить полученные выше модели (рис. 2): знаний, фактов, правил и данных в должной форме и в соответствующих частях Пролог - программы, см. пример ниже, в котором предметной областью является треугольник (triangle).

Пролог-программа triangle.pro

triangle.pro /* triangle - имя программы, pro - расширение, указывающее,*/

/* что программа написана на Прологе*/

Predicates /* раздел описания предикатов */

point(4,2). /* point(4,2) - факт, полученный из предиката point(X,Y) */

/*подстановкой $X=4, Y=2$; ниже точка point(4,2) обозначена как P3 и т.п.*/

point(6,4).

point(7,1).

seg1(P3, P4).

seg1(point(4,2), point(6,4)).

...

Clauses /* раздел клауз: правил $A \leftarrow B$, имеющих в Прологе вид $A :- B$ */

/* раздел утверждений, условий и т.п.*/

area(X,Y,Z):- Z is X*Y/2. /* здесь area(X,Y,Z) - площадь, ее числовое */

/* значение присваивается переменной Z*/

/* $X = h_a$ - высота на $Y=a$ - основание или */

/* $X = h_c$ и $Y=c$, или же $X = h_b$ и $Y=b$ */

...

zapros:- makewindow("Введите координаты точки").

Goal

/* раздел целей*/

zapro.

В программе triangle.pro отсутствуют следующие разделы:

Constans /* раздел констант, не обязательный */

Domens /* характеристика типов переменных и т.д.*/

/* (например) name = symbol - тип переменной symbol */.

Разделы

Clauses /* раздел правил, условия, предложения, правила, утверждения

*/

Goal /* раздел целей */

представлены лишь указанными выше фрагментами.

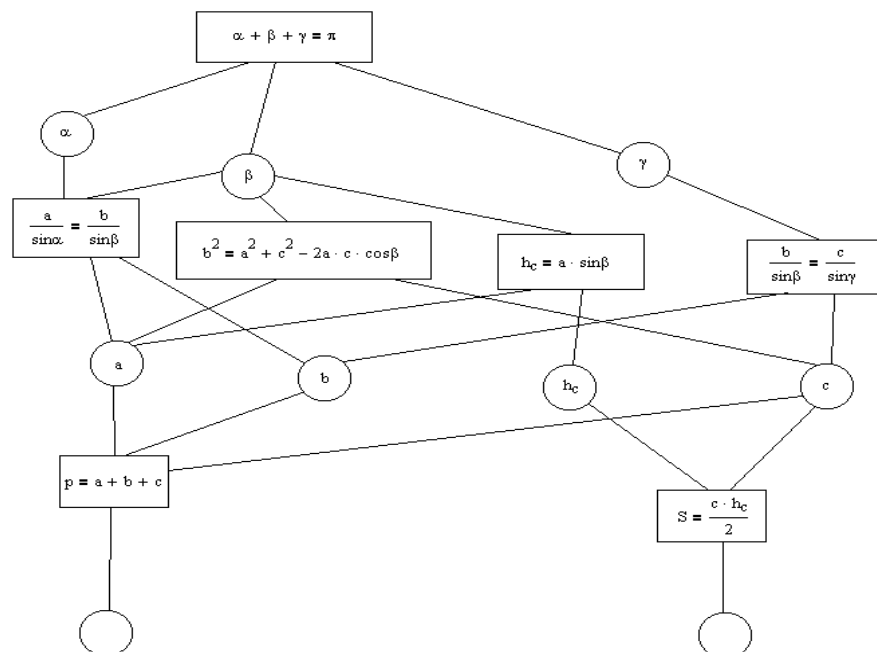
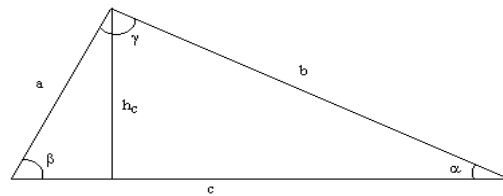


Рис. 6.2 Представление знаний, фактов, правил и данных с треугольником (предметная область).

Лабораторная работа 7. Разработка БЗ и БД для Пролог - программ решения прикладных задач

Ход выполнения работы:

- 1) Выбрать свой вариант задания.
- 2) Ознакомиться с заданием и выполнить его.

При необходимости обратиться к Пояснениям к лабораторной работе, которые приведены после пункта «Варианты лабораторной работы №7».

Варианты лабораторной работы №7

Разработать МПЗ для Базы Знаний и написать Базу Данных, Базу Фактов и Базу Правил для программы (Пролог - программы), решающую соответственно:

- 1) задачу о волке, козе и капусте;
- 2) задачу о трех каннибалах и трех миссионерах;
- 3) задачу о четырех ферзях на поле размером 4×4 клетки;
- 4) задачу о ферзях на поле размером 8×8 ;
- 5) задачу о шахматных ладьях на поле размером 8×8 ;
- 6) задачу о шахматных конях на поле размером 8×8 ;
- 7) задачу о лжецах и правдолюбцах;
- 8) задачу - игру в «крестики и нулики»;
- 9) задачу о коммивояжере;
- 10) задачу о Ханойских башнях.

1) Задача о “волке, козе и капусте”.

Человеку нужно переправиться через реку и перевезти с собой волка, козу и капусту. Имеется лодка, которой управляет человек. На лодке одновременно можно перевезти не более 2-х пассажиров, включая человека (он же лодочник). Если на одном и том же берегу с волком с козой нет человека, то он ее съест. Аналогично, если коза останется на берегу с капустой, то она съест капусту. Все остальные ситуации безопасны. Требуется организовать переправу через реку так, чтобы все пассажиры были доставлены в сохранности на другой берег реки.

2) Задача о “трех каннибалах и трех миссионерах”.

На правом берегу реки находятся три каннибала и три миссионера, которым нужно переправиться на левый берег на двухместной лодке, не считая 3-е место для лодочника. Никого другого на берегах нет, место лодочника занимает только он сам. Если на любом берегу каннибалов окажется больше, чем миссионеров, то первые съедят вторых. Как следует организовать переправу, чтобы все пассажиры оказались на правом берегу целыми и невредимыми, если к тому же лодка при ее курсировании с одного берега на другой обязательно должна перевозить хотя одного пассажира из 6-ти указанных (такое правило переправы на этой реке).

3) Задача о “4 ферзях” на поле размером 4×4 клетки.

Расставить как можно больше шахматных ферзей на доске размером 4×4 , чтобы они не «били» друг друга (конечно, каждый из ферзей находился в отдельной клетке: два ферзя не могут находиться в одной клетке, других фигур

и пешек на доске нет). Задача о размещении 4-х ферзей на шахматной доске состоит в определении того, можно ли расставить ровно 4-я ферзей на шахматной доске (4×4 клеток) таким образом, чтобы ни один ферзь не нападал на другого по шахматным правилам, а если можно, то каким числом способов. Правила ходов ферзя: ферзь со своей клетки может перемещаться на любую клетку по горизонтали, вертикали и диагонали, если на его пути не встречается другой ферзь. Если такой ферзь встречается, то первый ферзь должен либо срубить его и встать на его место, но не обходить ее (правило запрета), либо быть срубленным этим ферзем.

4) Задача о «8 ферзях».

Расставить 8 шахматных ферзей на доске размером 8×8, чтобы они не «били» друг друга (конечно, каждый из ферзей находился в отдельной клетке: два ферзя не могут находиться в одной клетке, других фигур и пешек на доске нет). Задача о размещении ферзей на шахматной доске состоит в определении того, можно ли расставить ровно восемь ферзей на шахматной доске (8×8 клеток) таким образом, чтобы ни один ферзь не нападал на другого по шахматным правилам, а если можно, то каким числом способов. Правила ходов ферзя: ферзь со своей клетки может перемещаться на любую клетку по горизонтали, вертикали и диагонали, если на его пути не встречается другой ферзь. Если такой другой ферзь встречается, то ферзь либо должен обязательно срубить его и встать на его место, но не обходить его (правило запрета), либо бить им срубленным.

5) Задача о «шахматных ладьях» на поле размером 8×8 клетки.

Расставить как можно больше шахматных ладей на доске размером 8×8, чтобы они не «били» друг друга (конечно, каждая из ладей находилась в отдельной клетке: две ладьи не могут находиться в одной клетке, других фигур и пешек на доске нет). Задача о размещении ладей на шахматной доске состоит в определении максимального числа ладей, которые можно расставить на шахматной доске (8×8 клеток) таким образом, чтобы ни одна ладья не нападала на другую, а если можно, то каким числом способов. Правила ходов ладьи: ладья со своей клетки может перемещаться на соответствующую клетку по известному шахматному правилу, если на ее пути не встречается другая ладья. Если такая другая ладья встречается, то ладья должна срубить ее и встать на ее место, но не обходить ее (правило запрета), либо быть ею срубленной.

6) Задача о «шахматных конях» на поле размером 8×8 клетки.

Расставить как можно больше шахматных коней на доске размером 8×8, чтобы они не «били» друг друга (конечно, каждый из коней находился в отдельной клетке: два коня не могут находиться в одной клетке, других фигур и пешек на доске нет). Задача о размещении коней на шахматной доске состоит в определении максимального количества коней, которых можно расставить

на шахматной доске (8×8 клеток) таким образом, чтобы ни один конь не напал на другого по шахматным правилам, а если можно, то каким числом способов. Правила ходов: конь со своей клетки может перемещаться на соответствующую клетку по шахматным правилам, если на этой клетке нет другого коня (правило запрета). В противном случае он должен срубить этого коня либо быть им срубленным.

7) Задача - логическая головоломка «Правдолюбцы и лжецы».

Население города состоит из правдолюбцев, всегда говорящих правду и только правду, и лжецов, всегда и только лгущих. В город приехал гость. Некоторые разговоры жителей этого города, которые он услышал, ему показались интересными.

Разговор 1. Житель А говорит жителю В: “либо я - лжец, либо ты правдолюбец”.

Разговор 2. Житель Д говорит жителям З и К: “либо я - лжец, либо З правдолюбец”.

Разговор 3. Один житель, подходя к жителям Р, П и Т спросил их: “а сколько среди вас правдолюбцев?”. Р ответил неопределенно, а П и говорит: “Р сказал, что среди нас есть один правдолюбец”. Но тут вмешался Т: “П лжет”.

Гость предлагает Вам выяснить, рассуждая логически, кто из жителей этого города - правдолюбец, а кто - лжец в каждом из 3-х случаев, исходя только из информации, приведенной в соответствующем разговоре.

8) Задача - игра в «крестики и нулики».

Исходным состоянием игры «Крестики-нолики» является пустая игровая доска размера 3×3, а конечным или целевым состоянием - доска на которой будет в одной строке, столбце или по диагонали располагается 3 крестика или нолика (в зависимости от того, чья победа предполагается). Путь из исходного состояния в конечное содержит последовательность ходов, ведущих к победе игрока, ставящего в клетке только х - крестик или только 0 - нолик.

9) Задача о коммивояжере.

Коммивояжер должен посетить каждый из пяти городов, не заходя более одного раза в каждый город. Между каждой парой городов имеется путь, длина которого (в километрах) указана на графе, см. рис. 1, а. Нужно, отправляясь из города, например, А, найти самый короткий путь, по которому коммивояжер по одному разу проходит через каждый город и в конце поездки возвращается в город А (рис. 3а).

Описание состояний, которым соответствуют вершины графа, осуществляется с помощью списка, элементами которого являются буквы. Последняя буква указывает город, которому соответствует данная вершина и в котором находится коммивояжер; предшествующая последовательность букв указывает путь, по которому коммивояжер попал в данный город. В простран-

стве состояний ищется решение. На рис. 3б это решение представлено последовательностью дуг (путем), ведущей из начальной вершины (А) в целевую вершину (ACDEBA), и составляет минимальный суммарный путь в 34 км.

Не следует смешивать исходный граф задачи, изображенный на рис. 7.1а, с графом, описывающим пространство состояний (рис.7.1.б).

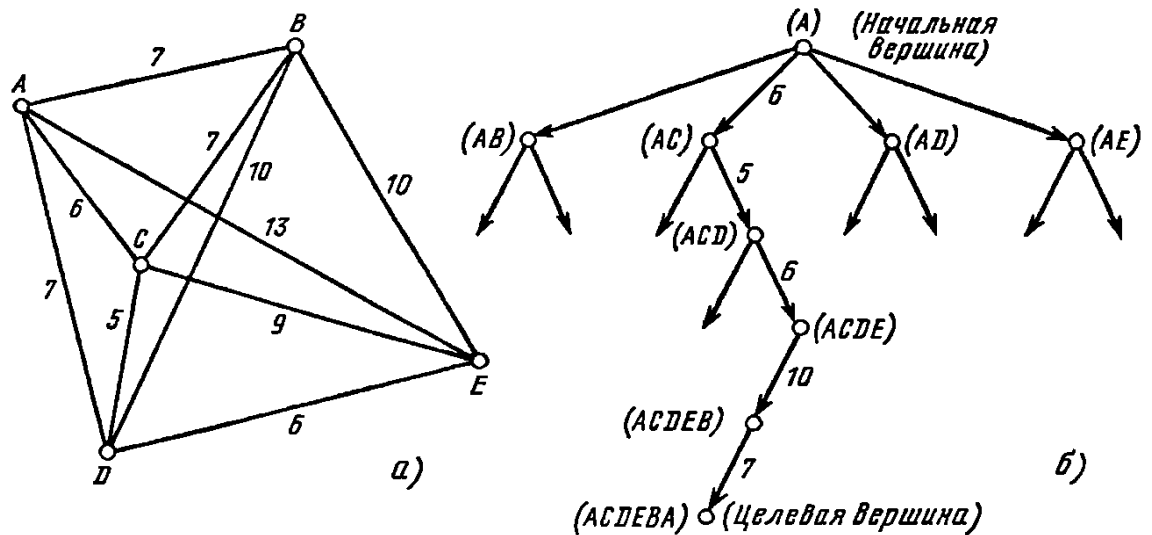


Рис. 7.1. Пояснение к задаче о коммивояжере.
а - граф дорог; б - пространство состояний.

10) Задача-головоломка «Ханойская башня» состоит в следующем. Имеется три стержня: а, б, с (рис. 7.2). В начале на стержне, а нанизано 64 диска уменьшающихся размеров (исходная башня). Все диски имеют разный диаметр, внизу находится самый большой диск, а наверху - самый маленький. Никакой диск большего диаметра не может находиться выше диска меньшего диаметра. В начале на стержнях б и с дисков нет. Задача: переложить все диски с а на б в том же порядке, в каком они были на а, пользуясь стержнем с как вспомогательным. При этом ни на одном из стержней никогда не должно возникать ситуации, при которой диск большего диаметра лежит на диске меньшего диаметра.

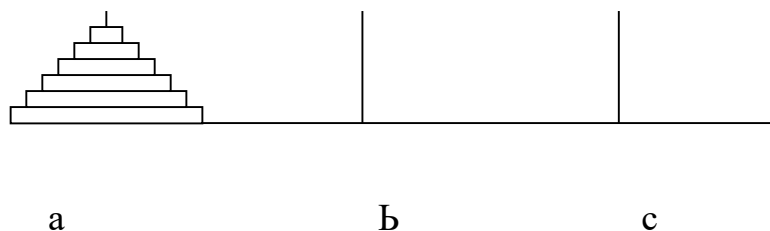


Рис. 7.2 - Ханойская башня

Пояснения к лабораторной работе №7

Целью предшествующего изложения на лекциях и работ практических занятиях была подготовка к решению на Прологе прикладных задач не вычислительного характера (логических задач, задач - головоломок, игр и т.д.), в которых ярко проявляются особенности Пролога и заложенной в него дескриптивной парадигмы программирования.

Пример представления фактов, правил и запросов в Пролог - программе.

Основными компонентами программы на Прологе являются разделы: предикаты (predicates), цель (goal) и условия (clauses).

Предикатами называются высказывания, составные части которых являются переменными (в примере ниже - символьными именами (symbol)).

Цель - это последовательность фактов, истинность которых должен доказать механизм логического вывода. В Прологе используется обратный способ вывода цели.

Условия могут быть либо фактами - предикатами с конкретными значениями символических имен, либо правилами. Заключение правила в Прологе пишется слева от знака `:-`, а условия, связанные логическими операциями "И", - через запятую справа.

```
Predicates
ruk(symbol,symbol)
nach(symbol,symbol)
print 1 (symbol,symbol)
Clauses
ruk("Петров","Иванов").
ruk("Петров","Смирнов").
ruk("Яшин","Петров").
nach(X,Y) :- ruk(X,Y).
nach(X,Y) :- ruk(X,Z),nach(Z,Y).
print1(Q,S):- nach(Q,S),write(Q),nl,fail.
Goal
write("Введите фамилию "), nl, readln(S), write("Список начальников"),
nl,
write("для ",S), nl, print 1(Q,S).
```

В этом примере введены предикаты:

`ruk(X,Y)` для представления в БД факта, что X является непосредственным руководителем - начальником сотрудника Y,

`nach(X,Y)` для представления в БД факта, что X является руководителем для Y,

`print1(Q,S)` для печати всех руководителей сотрудника S.

В разделе Clauses (условия) утверждения содержат факты в форме предикатов о конкретных непосредственных руководителях и правила для предикатов. В данной программе сведения о непосредственных руководителях хранятся в явном виде, а о других руководителях-начальниках не хранятся в явном виде, а являются результатом вывода получаемого по окончании работы данной Пролог - программы.

Практическая работа № 1

РАЗБИЕНИЕ НА ЭТАПЫ ЖИЗНЕННОГО ЦИКЛА (ЖЦ).

ДИАГРАММА ГАНТА

Цель работы: Научиться определять порядок выполнения, представлять и описывать этапы жизненного цикла информационной системы.

1. Теоретическая часть

1.1. Жизненный цикл АИС

Жизненный цикл информационных систем – это период их создания и использования, охватывающий различные состояния, начиная с момента возникновения необходимости в такой системе и заканчивая моментом ее полного выхода из употребления у пользователей.

Полный жизненный цикл информационной системы включает в себя, как правило: анализ, проектирование, кодирование (программирование), тестирование, внедрение и эксплуатацию.

Жизненный цикл образуется в соответствии с принципом нисходящего проектирования и, как правило, носит итерационный характер: реализованные этапы, начиная с самых ранних, циклически повторяются в соответствии с изменениями требований и внешних условий, введением ограничений и т.п. На каждом этапе жизненного цикла порождается определенный набор документов и технических решений, при этом для каждого этапа исходными являются документы и решения, полученные на предыдущем этапе. Каждый этап завершается верификацией порожденных документов и решений с целью проверки их соответствия исходными.

Сведем данные по каждому этапу в итоговую таблицу (Таблица 2.1).

Таблица 2.1. **Жизненный цикл АИС.**

№	Наименование этапа	Основные характеристики
---	--------------------	-------------------------

1	Разработка и анализ бизнес - модели	Определяются основные задачи АИС, проводится декомпозиция задач по модулям и определяются функции с помощью которых решаются эти задачи. Описание функций осуществляется на языке производственных (описание процессов предметной области), функциональных (описание форм обрабатываемых документов) и технических требований (аппаратное, программное, лингвистическое обеспечение АИС). Метод решения: Функциональное моделирование. Результат: 1.Концептуальная модель АИС, состоящая из описания предметной области, ресурсов и потоков данных, перечень требований и ограничений к технической реализации АИС. 2.Аппаратно-технический состав создаваемой АИС.
2	Формализация бизнес - модели, разработка логической модели бизнес-процессов.	Разработанная концептуальная модель формализуется, т.е. воплощается в виде логической модели АИС. Метод решения: Разработка диаграммы "сущность-связь" (ER (Entity-Relationship) – CASE- диаграммы). Результат: Разработанное информационное обеспечение АИС: схемы и структуры данных для всех уровней модульности АИС, документация по логической структуре АИС, сгенерированные скрипты для создания объектов БД.

3	Выбор лингвистического обеспечения, разработка программного обеспечения АИС.	Разработка АИС: выбирается лингвистическое обеспечение (среда разработки – инструментарий), проводится разработка программного и методического обеспечения. Разработанная на втором этапе логическая схема воплощается в реальные объекты, при этом логические схемы реализуются в виде объектов базы данных, а функциональные схемы – в пользовательские формы и приложения. Метод решения: Разработка программного кода с использованием выбранного инструментария. Результат: Работоспособная АИС.
4	Тестирование и отладка АИС	На данном этапе осуществляется корректировка информационного, аппаратного, программного обеспечения, проводится разработка методического обеспечения (документации разработчика, пользователя) и т.п. Результат: Оптимальный состав и эффективное функционирование АИС. Комплект документации: разработчика, администратора, пользователя.
5	Эксплуатация и контроль версий	Особенность АИС созданных по архитектуре клиент-сервер является их многоуровневость и многомодульность, поэтому при их эксплуатации и развитии на первое место выходят вопросы контроля версий, т.е. добавление новых и развитие старых модулей с выводом из эксплуатации старых. Например,

		если ежедневный контроль версий не ведется, то в как показала практика, БД АИС за год эксплуатации может насчитывать более 1000 таблиц, из которых эффективно использоваться будет лишь 20-30%. <u>Результат: Наращиваемость и безизбыточный состав гибкой, масштабируемой АИС</u>
--	--	--

1.2. Диаграмма ГАНТА

Диагра́мма Га́нта (англ. Gantt chart, также ленточная диаграмма, график Ганта) — это популярный тип столбчатых диаграмм (гистограмм), который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Используется в приложениях по управлению проектами.

По сути, диаграмма Ганта состоит из полос, ориентированных вдоль оси времени. Каждая полоса на диаграмме представляет отдельную задачу в составе проекта (вид работы), её концы — моменты начала и завершения работы, её протяженность — длительность работы. Вертикальной осью диаграммы служит перечень задач. Кроме того, на диаграмме могут быть отмечены совокупные задачи, проценты завершения, указатели последовательности и зависимости работ, метки ключевых моментов (вехи), метка текущего момента времени «Сегодня» и др.

Диаграммы Ганта позволяет:

1. Визуально оценить последовательность задач, их относительную длительность и протяженность проекта в целом;
2. Сравнить планируемый и реальный ход выполнения задач;
3. Детально проанализировать реальный ход выполнения задач.

Так как в силу разных причин иногда под рукой нет специализированных программ для создания Диаграммы Ганта, но необходимость в построении данной диаграммы возникает довольно часто, то рассмотрим построение

данной диаграммы в популярной программе Excel. Теперь рассмотрим построение Диаграммы Ганта более подробно [4].

1) Ввод начальных данных.

рассмотрим отображение процесса приготовления супа на диаграмме Ганта. Приготовление супа состоит из нескольких процессов (Таблица. 2.2). Некоторые процессы выполняются последовательно, а некоторые параллельно, соответственно каждый процесс имеет время начала, длительность и время конца.

Откроем новый Лист в Excel и введем данные из Таблицы 1.2.

	A	B	C	D
1	Приготовить суп			
2	Процесс	Время начала, час	Продолжительность, час	Время конца, час
3	1. Отварить мясо	18:00	1:00	19:00
4	2. Накрошить капусту	18:00	0:20	18:20
5	3. Накрошить морковь	18:20	0:10	18:30
6	4. Накрошить картофель	18:30	0:10	18:40
7	5. Отварить капусту	19:00	0:15	19:15
8	6. Отварить морковь	19:15	0:20	19:35
9	7. Добавить зелень	19:35	0:05	19:40
10	8. Довести до кипения	19:40	0:10	19:50
11	Итого:	18:00	1:50	19:50

Таблица 1.2. Список процессов для приготовления супа.

Диаграмму Ганта можно построить несколькими способами:

1. Самый простой, но менее наглядный способ состоит в использовании стандартных текстовых функций Excel - "ПОВТОР". Функция ПОВТОР (REPT) позволяет заполнить ячейку строкой символов, повторенной заданное количество раз. Синтаксис:

=ПОВТОР(текст;число_повторений),

где текст - размноженная строка символов, заключенная в кавычки; число_повторений указывает, сколько раз нужно повторить текст. Если число_повторений равен 0, функция ПОВТОР оставляет ячейку пустой, а если

он не является целым числом, эта функция отбрасывает десятичные знаки после запятой.

Соответственно в качестве текста выступает символ "0", а число_повторений - колонка D для каждого из процесса. Если время начала процесса зависит от времени окончания предыдущего, в этом случае используется две функции "ПОВТОР", разделенных символом "&". В первой функции в качестве текста участвует пробел - " ", а число_повторений - суммарная продолжительность процессов, от которой зависит время начала текущего. Во второй функции в качестве текста участвует символ - "0", а число_повторений - продолжительность текущего процесса.

Данный способ отображен в Таблице 1.3.

	A	B	C	D	E	F
1	Приготовить суп					
2	Процесс	Время начала, час	Продолжительность, час	Время конца, час	Диаграмма Ганта	формула
3	1. Отварить мясо	18:00	1:00	19:00		=ПОВТОР("0";D3)
4	2. Накрошить капусту	18:00	0:20	18:20		=ПОВТОР("0";D4)
5	3. Накрошить морковь	18:20	0:10	18:30		=ПОВТОР(" ";D4)&ПОВТОР("0";D5)
6	4. Накрошить картофель	18:30	0:10	18:40		=ПОВТОР(" ";D5+D4)&ПОВТОР("0";D6)
7	5. Отварить капусту	19:00	0:15	19:15		=ПОВТОР(" ";D3)&ПОВТОР("0";D7)
8	6. Отварить морковь	19:15	0:20	19:35		=ПОВТОР(" ";D3+D7)&ПОВТОР("0";D8)
9	7. Добавить зелень	19:35	0:05	19:40		=ПОВТОР(" ";D3+D7+D8)&ПОВТОР("0";D9)
10	8. Довести до кипения	19:40	0:10	19:50		=ПОВТОР(" ";D3+D8+D9+D7)&ПОВТОР("0";D10)
11	Итого:	18:00	1:50	19:50		

Таблица 1.3. Диаграмма Ганта вариант 1.

2. С помощью средства Мастер диаграмм.

1). Построим линейчатую диаграмму с накоплением основе данных диапозона A3:C10 (таблица 1.4.).

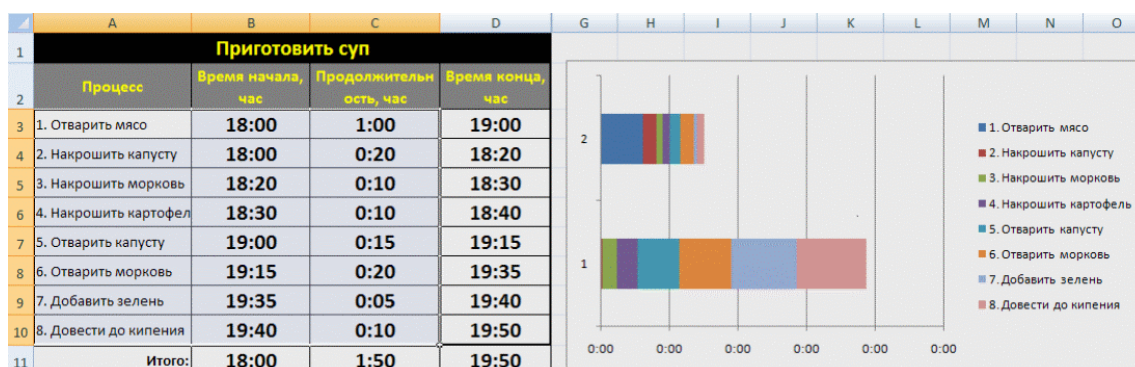


Таблица. 1.4. Линейчатая диаграмма с накоплением

2). Приведем диаграмму к виду диаграммы Ганта.

Щелкните правой кнопкой в область диаграммы, выберите *Выбрать данные*. В открывшемся окне щелкните *Строка/столбец*. Нажмите *ОК*.

Щелкните правой кнопкой первые ряды на диаграмме. Это ряды для *Время* начала. Если в программе Microsoft Excel используются цвета, установленные по умолчанию, то эти ряды имеют голубой цвет. Выберите *Формат ряда данных*. Во вкладке *Заливка* установите *Нет заливки*, во вкладке *Цвет границ* установите *Нет линий*. Нажмите *Заккрыть*.

Дважды щелкните ось категорий (x), которая на линейчатой диаграмме расположена вертикально. На линейчатой диаграмме оси x и y поменяны местами. Выберите *Формат оси*. Перейдите на вкладку *Шкала* и установите флажок *обратный порядок категорий*.

Щелкните правой кнопкой на легенду диаграммы, выберите *Удалить*.

Дважды щелкните ось категорий (y), которая на линейчатой диаграмме расположена горизонтально. Выберите *Формат оси*. Перейдите на вкладку *Шкала* и установите в поле *Цена промежуточных делений* флажок *фиксированное* и значение равное 5,0. В поле *Промежуточные* выберите *Наружу*. Перейдите на вкладку *Число* и установите в поле *Числовые форматы* значение *Общий*.

Дополнительно можете добавить на диаграмму линии сетки с промежуточными значениями, изменить заливку (таблица. 2.5).

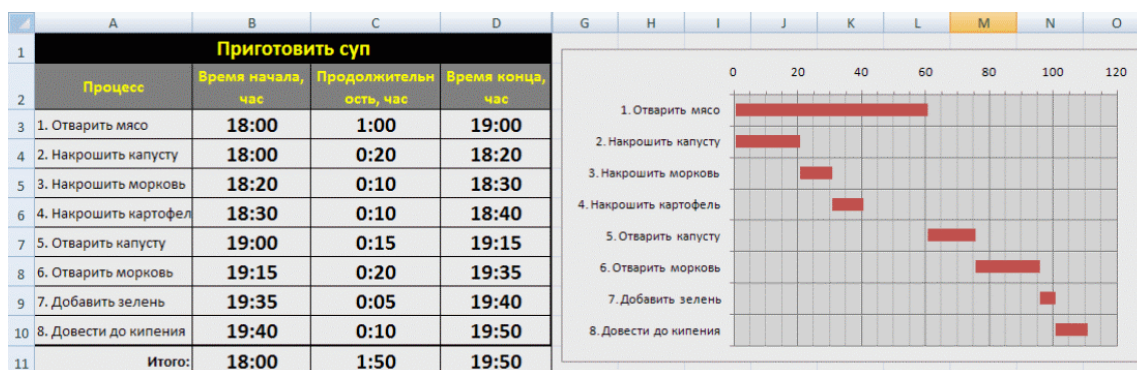


Таблица. 1.5. Диаграмма Ганта вариант 2.

В дальнейшем можно заняться «украшательством» своей диаграммы – добавить условное форматирование, чтобы автоматически выделять наиболее «затратные» задачи цветом, можно добавить фильтры.

2. Порядок выполнения работы

1. Изучить теоретическую часть по приведенным выше данным и дополнительной литературе.
2. Просмотреть демонстрационный пример.
3. Получить у преподавателя вариант задания для выполнения.
4. Проанализировать полученное задание.
5. Определить характер решаемой задачи.
6. Построить диаграмму Ганта и описать порядок выполнения этапов жизненного цикла АИС.

3. Варианты заданий

Построить диаграмму Ганта и описать порядок выполнения этапов жизненного цикла для следующих автоматизированных информационных систем:

По выбору: система по отысканию рыночных ниш; системы, ускоряющие потоки товаров; системы по снижению издержек производства, системы автоматизации технологии ("менеджмент уступок"); управление производственным процессом; системы документооборота; оперативное управление предприятием, информационная интеллектуальная система, обучающая система, информационная библиотечная система и т.д.

4. Контрольные вопросы

1. Перечислите основные этапы жизненного цикла автоматизированных информационных систем?
2. Что представляет диаграмма Ганта?
3. Опишите процесс построения диаграммы Ганта?
4. Возможности диаграммы Ганта?

5. Литература

1. Петров В.Н.. Информационные системы. Санкт-Петербург, Москва-Харьков-Минск, 2004
2. <http://www.fms.gov.ru/upload/iblock/669/gismu.pdf>
3. <http://delovoymir.biz/ru/columns/1349/>
4. http://easyprog.ru/index.php?option=com_content&task=view&id=251

Практическая работа № 2

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Цель работы: Научиться строить модели, создаваемые в процессе проектирования баз данных: инфологическая, даталогическая, физическая

1. Теоретическая часть

I. Инфологическое моделирование как результат разработки модели программного обеспечения (ПО)

Инфологическая модель применяется на втором этапе проектирования БД, то есть после словесного описания предметной области. Инфологическая модель представляет собой описание будущей базы данных, представленное с помощью естественного языка, формул, графиков, диаграмм, таблиц и других средств, понятных как разработчикам базы данных (БД), так и обычным пользователям. Назначение такой модели состоит в адекватном описании процессов, информационных потоков, функций системы с помощью общедоступного и понятного языка, что делает возможным привлечение экспертов предметной области, консультантов, пользователей для обсуждения модели и внесения исправлений. В данном случае под созданием инфологической модели будем понимать именно ее создание для БД. В общем случае, инфологическая модель может создаваться для любой проектируемой системы и представляет ее описание (в общем случае в произвольной форме).

Создание инфологической модели является естественным продолжением исследований предметной области, но в отличие от него является представлением БД с точки зрения проектировщика (разработчика). Наглядность представления такой модели позволяет экспертам предметной области оценить ее точность и внести исправления. От правильности модели зависит успех дальнейшей разработки.

Инфологическую модель можно представить в виде словесного описания, однако наиболее наглядным является использование специальных графических нотаций, разработанных для проведения подобного рода моделирования.

Важно отметить, что создаваемая на этом этапе модель полностью не зависит от физической реализации будущей системы. В случае с БД это означает, что совершенно не важно где физически будут храниться данные: на бумаге, в памяти компьютера и кто или что эти данные будет обрабатывать. В этом случае, когда структуры данных не зависят от их физической реализации, а моделируются исходя из их смыслового назначения, моделирование называется семантическим.

Существует несколько способов описания инфологической модели, однако, в настоящее время одним из наиболее широко распространенных подходов, применяемых при инфологическом моделировании, является подход, основанный на применении диаграмм «сущность-связь» (ER – Entity Relationship). При рассмотрении последующих примеров будем использовать одну из самых распространенных в рамках ER моделей нотацию IDEF1X. Данный стандарт был разработан в 1993 г. Национальным институтом стандартизации и технологий и является федеральным стандартом обработки информации (США), описывающим семантику и синтаксис языка, правила и технологии для разработки логической модели данных.

Для построения инфологической модели важно знать элементы этой модели. Базовыми элементами модели сущность-связь являются сущности.

Сущность по форме представляет собой только некоторое реальное описание объекта, точнее набор описаний его значимых признаков-атрибутов. Конкретный набор значений атрибутов объекта будет называться экземпляром сущности.

II Даталогическое проектирование

Даталогическая модель (ДЛМ) строится на основе ИЛМ. ДЛМ БД является концептуальной моделью БД и отражает логические связи между информационными элементами ДЛМ. В ДЛМ фиксируются данные и связи данных между ними.

ДЛМ строится в терминах информационных единиц, допустимых в той конкретной СУБД, в среде которой проектируется БД. ДЛМ зависит от выбора СУБД для разработки БД или информационной модели.

Схема БД - описание ДЛМ на языке выбранной СУБД.

III Физическое проектирование

Для привязки ДЛМ к среде хранения используется модель данных физического уровня (физическая модель, внутренняя модель). Эта модель БД определяется используемыми ЗУ, способами физической организации данных в среде хранения.

Физическая модель, также как и ДЛМ, строится с учетом особенностей выбранной СУБД.

Физическое проектирование — описание физической структуры БД.

Рассмотрим проектирование базы данных на примере модели данных «Библиотека».

К стержневым сущностям можно отнести:

1. Создатели (Код создателя, Создатель).

Эта сущность отводится для хранения сведений об основных людях, принимавших участие в подготовке рукописи издания (авторах, составителях, титульных редакторах, переводчиках и художниках). Такое объединение допустимо, так как данные о разных создателях выбираются из одного домена (фамилия и имена) и исключает дублирование данных (один и тот же человек

может играть разные роли в подготовке разных изданий). Например, С.Я.Маршак писал стихи (Сказка о глупом мышонке) и пьесы (Двенадцать месяцев), переводил Дж.Байрона, Р.Бернса, Г.Гейне и составлял сборники стихов.

Так как фамилия и имена (инициалы) создателя могут быть достаточно громоздкими (М.Е. Салтыков-Щедрин, Франсуа Рене де Шатобриан, Остен Жюль Жан-Батист Ипполит и т.п.) и будут многократно встречаться в разных изданиях, то их целесообразно нумеровать и ссылаться на эти номера. Для этого вводится целочисленный атрибут "Код_создателя", который будет автоматически наращиваться на единицу при вводе в базу данных нового автора, переводчика или другого создателя.

Аналогично создаются: Код_издательства, Код_заглавия, Вид_издания, Код_характера, Код_языка, Номер_билета, Номер_переплета, Код_места и Код_издания, замещающие от одного до девяти атрибутов.

2. *Издательства (Код_издательства, Название, Город).*

3. *Заглавия (Код_заглавия, Заглавие).*

Выделение этой сущности позволит сократить объем данных и снизить вероятность возникновения противоречивости (исключается необходимость ввода длинных текстовых названий для различных томов собраний сочинений, повторных изданий, учебников и т.п.).

4. *Вид_издания (Вид_издания, Название_вида).*

5. *Характеры (Код_характера, Характер_переиздания).*

6. *Языки (Код_языка, Язык, Сокращение).*

Кроме названия языка хранится его общепринятое сокращение (англ., исп., нем., фр.), если оно существует.

7. *Места (Код_места, Номер_комнаты, Номер_стеллажа, Номер_полки).*

Один из кодов этой сущности (например, "-1") отведен для описания обобщенного места, находящегося за стенами хранилища книг (издание выдано читателю, временно передано другой библиотеке или организации).

8. Читатели (Номер билета, Фамилия, Имя, Отчество, Адрес, Телефон).

Две ключевые сущности, описывающие издание и его конкретные экземпляры, оказываются зависимыми от других сущностей и попадают в класс обозначений:

1. Издание (Код издания, Код заглавия, Вид издания, Номер тома, Авторский знак, Библиотечн_шифр, Повторность, Код издательства, Год издания, Аннотация) [Заглавия, Вид издания, Издательства];
2. Переплеты (Номер переплета, Код издания, Цена, Дата приобретения)[Издания];

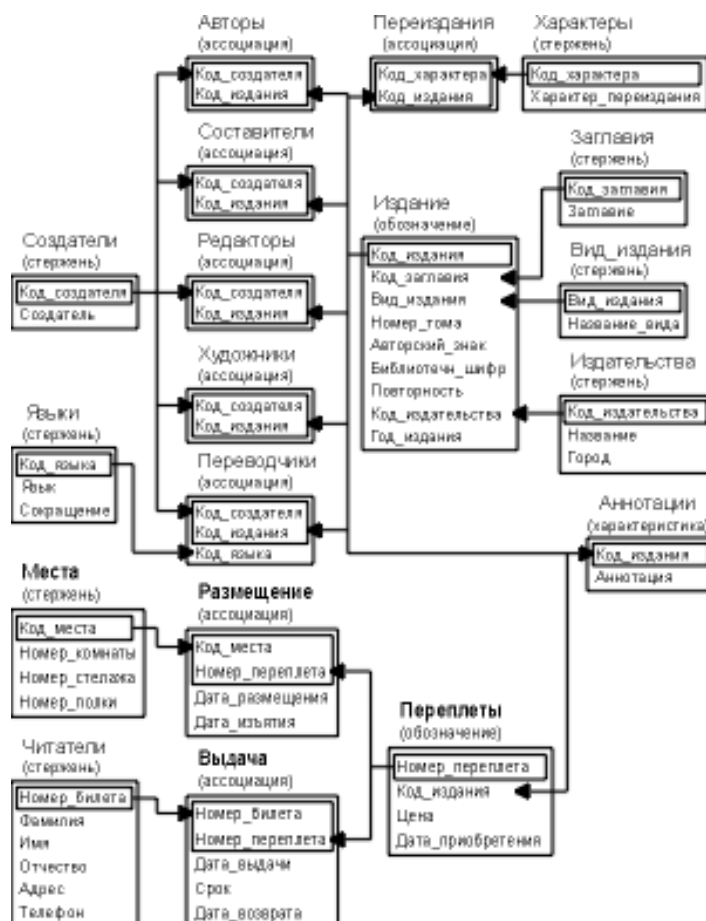


Рис. 2.1. Инфологическая модель базы данных "Библиотека", построенная с помощью языка "Таблицы-связи"

Стержневые сущности и обозначения связаны между собой ассоциациями:

1. Авторы [Создатели М, Издание N] (Код_создателя, Код_издания).
2. Составители [Создатели М, Издания N] (Код_создателя, Код_издания).
3. Редакторы [Создатели М, Издания N] (Код_создателя, Код_издания).
4. Художники [Создатели М, Издания N] (Код_создателя, Код_издания).
5. Переводчики [Создатели М, Издания N] (Код_создателя, Код_издания, Язык).
6. Переиздания [Характеры М, Издания N] (Код_характера, Код_издания).
7. Размещение [Места М, Переплеты N] (Код_места, Номер_переплета, Дата_размещения, Дата_изъятия).
8. Выдача [Читатели М, Переплеты N] (Номер_билета, Номер_переплета, Дата_выдачи, Срок, Дата_возврата).

И, наконец, для уменьшения объема часто используемого обозначения "Издания" из него выделена характеристика:

1. Аннотации (Код_издания, Аннотация) {Издание}.

В соответствии с процедурой проектирования каждая из полученных сущностей должна быть представлена базовой таблицей. Первый вариант этих таблиц описывается так:

```
СОЗДАТЬ ТАБЛИЦУ Создатели *( Стержневая сущность )
    ПЕРВИЧНЫЙ КЛЮЧ ( Код_создат )
    ПОЛЯ ( Код_создат Целое, Фам_ИО Текст 30 );
СОЗДАТЬ ТАБЛИЦУ Издательства *( Стержневая сущность )
    ПЕРВИЧНЫЙ КЛЮЧ ( Код_издательства )
    ПОЛЯ ( Код_издательства Целое, Название
        Текст 40, Город Текст 25 );
СОЗДАТЬ ТАБЛИЦУ Заглавия *( Стержневая сущность )
    ПЕРВИЧНЫЙ КЛЮЧ ( Код_заглавия )
    ПОЛЯ ( Код_заглавия Целое, Заглавие Запись );
СОЗДАТЬ ТАБЛИЦУ Вид_издания *( Стержневая сущность )
    ПЕРВИЧНЫЙ КЛЮЧ ( Вид_издания )
    ПОЛЯ ( Вид_издания Целое, Название_вида Текст 16);
СОЗДАТЬ ТАБЛИЦУ Характеры *( Стержневая сущность )
    ПЕРВИЧНЫЙ КЛЮЧ ( Код_характера )
    ПОЛЯ ( Код_характера Целое, Характер_переиздания Текст 16 );
СОЗДАТЬ ТАБЛИЦУ Языки *( Стержневая сущность )
    ПЕРВИЧНЫЙ КЛЮЧ ( Код_языка )
    ПОЛЯ ( Код_языка Целое, Язык Текст 16, Сокращение Текст 6 );
СОЗДАТЬ ТАБЛИЦУ Места *( Стержневая сущность )
    ПЕРВИЧНЫЙ КЛЮЧ ( Код_места )
    ПОЛЯ ( Код_места Целое, Номер_комнаты Целое,
        Номер_стелажа Целое, Номер_полки Целое );
```

СОЗДАТЬ ТАБЛИЦУ Читатели *(Стержневая сущность)
 ПЕРВИЧНЫЙ КЛЮЧ (Ном_билета)
 ПОЛЯ (Ном_билета Целое, Фамилия Текст 20, Имя Текст 16,
 Отчество Текст 20, Адрес Текст 60, Телефон Текст 9);

СОЗДАТЬ ТАБЛИЦУ Издание *(Обозначение)
 ПЕРВИЧНЫЙ КЛЮЧ (Код_издания)
 ВНЕШНИЙ КЛЮЧ (Код_заглавия ИЗ Заглавия
 NULL-значения НЕ ДОПУСТИМЫ
 УДАЛЕНИЕ ИЗ Заглавия ОГРАНИЧИВАЕТСЯ
 ОБНОВЛЕНИЕ Заглавия.Код_заглавия ОГРАНИЧИВАЕТСЯ)
 ВНЕШНИЙ КЛЮЧ (Вид_издания ИЗ Вид_издания
 NULL-значения ДОПУСТИМЫ
 УДАЛЕНИЕ ИЗ Вид_издания ОГРАНИЧИВАЕТСЯ
 ОБНОВЛЕНИЕ Вид_издания.Вид_издания КАСКАДИРУЕТСЯ)
 ВНЕШНИЙ КЛЮЧ (Код_издательства ИЗ Издательства
 NULL-значения НЕ ДОПУСТИМЫ
 УДАЛЕНИЕ ИЗ Издательства ОГРАНИЧИВАЕТСЯ
 ОБНОВЛЕНИЕ Издательства.Код_издательства КАСКАДИРУЕТСЯ)
 ПОЛЯ (Код_издания Целое, Код_заглавия Целое,
 Вид_издания Текст 16, Номер_тома Целое,
 Авторский_знак Текст 3, Библиотечн_шифр Текст 12,
 Повторность Целое, Код_издательств-ва Целое,
 Год_издания Целое)
 ОГРАНИЧЕНИЯ (1. Значения полей Код_заглавия, Вид_издания
 и Код_издательства должны принадлежать набору значений
 соответствующих полей таблиц Заглавия, Вид_издания
 и Издательства; при нарушении вывод сообщения "Такого
 заглавия нет", "Такого вида издания нет" или "Такого
 издательства нет".);

СОЗДАТЬ ТАБЛИЦУ Переплеты *(Обозначение)
 ПЕРВИЧНЫЙ КЛЮЧ (Номер_переплета)
 ВНЕШНИЙ КЛЮЧ (Код_издания ИЗ Издания
 NULL-значения НЕ ДОПУСТИМЫ
 УДАЛЕНИЕ ИЗ Издания ОГРАНИЧИВАЕТСЯ
 ОБНОВЛЕНИЕ Издания.Код_издания КАСКАДИРУЕТСЯ)
 ПОЛЯ (Номер_переплета Целое, Код_издания Целое, Цена Деньги,
 Дата_приобретения Дата)
 ОГРАНИЧЕНИЯ (Значения поля Код_издания должны принадлежать набору
 значений соответствующего поля таблицы Издания;
 при нарушении вывод сообщения "Такого издания нет");

СОЗДАТЬ ТАБЛИЦУ Аннотации *(Характеризует Издания)
 ПЕРВИЧНЫЙ КЛЮЧ (Код_издания)
 ВНЕШНИЙ КЛЮЧ (Код_издания ИЗ Издания
 NULL-значения ДОПУСТИМЫ
 УДАЛЕНИЕ ИЗ Издания ОГРАНИЧИВАЕТСЯ
 ОБНОВЛЕНИЕ Издания.Код_издания КАСКАДИРУЕТСЯ)
 ПОЛЯ (Код_издания Целое, Аннотация Запись)
 ОГРАНИЧЕНИЯ (Значения поля Код_издания должны принадлежать набору
 значений соответствующего поля таблицы Издания;
 при нарушении вывод сообщения "Такого издания нет");

СОЗДАТЬ ТАБЛИЦУ Авторы *(Связывает Создатели и Издания)
 ПЕРВИЧНЫЙ КЛЮЧ (Код_создателя, Код_издания)
 ВНЕШНИЙ КЛЮЧ (Код_создателя ИЗ Создатели
 NULL-значения НЕ ДОПУСТИМЫ
 УДАЛЕНИЕ ИЗ Создатели ОГРАНИЧИВАЕТСЯ
 ОБНОВЛЕНИЕ Создатели.Код_создателя КАСКАДИРУЕТСЯ)
 ВНЕШНИЙ КЛЮЧ (Код_издания ИЗ Издания
 NULL-значения НЕ ДОПУСТИМЫ
 УДАЛЕНИЕ ИЗ Издания ОГРАНИЧИВАЕТСЯ
 ОБНОВЛЕНИЕ Издания.Код_издания КАСКАДИРУЕТСЯ)
 ПОЛЯ (Код_создателя Целое, Код_издания Целое)
 ОГРАНИЧЕНИЯ (Значения полей Код_создателя и Код_издания должны
 принадлежать набору значений соответствующих полей
 таблиц Создатели и Издания; при нарушении вывод

сообщения "Такого автора нет" или "Такого издания нет");

Аналогичное содержание имеют описания таблиц Составители, Редакторы, Художники и Переиздания. Остальные же таблицы проектируемой базы данных описываются так:

```
СОЗДАТЬ ТАБЛИЦУ Переводчики *( Связывает Создатели, Издания и Языки)
ПЕРВИЧНЫЙ КЛЮЧ ( Код_создателя, Код_издания )
ВНЕШНИЙ КЛЮЧ ( Код_создателя ИЗ Создатели
                NULL-значения НЕ ДОПУСТИМЫ
                УДАЛЕНИЕ ИЗ Создатели ОГРАНИЧИВАЕТСЯ
                ОБНОВЛЕНИЕ Создатели.Код_создателя КАСКАДИРУЕТСЯ)
ВНЕШНИЙ КЛЮЧ ( Код_издания ИЗ Издания
                NULL-значения НЕ ДОПУСТИМЫ
                УДАЛЕНИЕ ИЗ Издания ОГРАНИЧИВАЕТСЯ
                ОБНОВЛЕНИЕ Издания.Код_издания КАСКАДИРУЕТСЯ)
ВНЕШНИЙ КЛЮЧ ( Код_языка ИЗ Языки
                NULL-значения НЕ ДОПУСТИМЫ
                УДАЛЕНИЕ ИЗ Языки ОГРАНИЧИВАЕТСЯ
                ОБНОВЛЕНИЕ Языки.Код_языка КАСКАДИРУЕТСЯ)
ПОЛЯ ( Код_создателя Целое, Код_издания Целое )
ОГРАНИЧЕНИЯ ( Значения полей Код_создателя, Код_издания и
                Код_языка должны принадлежать набору значений
                соответствующих полей таблиц Создатели, Издания
                и Языки; при нарушении вывод сообщения "Такого
                автора нет" или "Такого издания нет" или "Такого
                языка нет" );

СОЗДАТЬ ТАБЛИЦУ Размещение *( Связывает Места и Переплеты )
ПЕРВИЧНЫЙ КЛЮЧ ( Код_места, Номер_переплета )
ВНЕШНИЙ КЛЮЧ ( Код_места ИЗ Места
                NULL-значения НЕ ДОПУСТИМЫ
                УДАЛЕНИЕ ИЗ Места ОГРАНИЧИВАЕТСЯ
                ОБНОВЛЕНИЕ Места.Код_места КАСКАДИРУЕТСЯ)
ВНЕШНИЙ КЛЮЧ ( Номер_переплета ИЗ Переплеты
                NULL-значения НЕ ДОПУСТИМЫ
                УДАЛЕНИЕ ИЗ Переплеты КАСКАДИРУЕТСЯ
                ОБНОВЛЕНИЕ Переплеты.Ном_переплета КАСКАДИРУЕТСЯ)
ПОЛЯ ( Код_места Целое, Номер_переплета Целое,
        Дата_размещения Дата, Дата_изъятия Дата )
ОГРАНИЧЕНИЯ ( Значения полей Код_места и Номер_переплета
                должны принадлежать набору значений соответствующих
                полей таблиц Переплеты и Места; при нарушении вывод
                сообщения "Такого переплета нет" или "Такого места нет" );

СОЗДАТЬ ТАБЛИЦУ Выдача *( Связывает Читатели и Переплеты )
ПЕРВИЧНЫЙ КЛЮЧ ( Ном_билета, Ном_переплета )
ВНЕШНИЙ КЛЮЧ ( Ном_билета ИЗ Читатели
                NULL-значения НЕ ДОПУСТИМЫ
                УДАЛЕНИЕ ИЗ Читатели КАСКАДИРУЕТСЯ
                ОБНОВЛЕНИЕ Читатели.Ном_билета КАСКАДИРУЕТСЯ)
ВНЕШНИЙ КЛЮЧ ( Ном_переплета ИЗ Переплеты
                NULL-значения НЕ ДОПУСТИМЫ
                УДАЛЕНИЕ ИЗ Переплеты КАСКАДИРУЕТСЯ
                ОБНОВЛЕНИЕ Переплеты.Ном_переплета КАСКАДИРУЕТСЯ)
ПОЛЯ ( Ном_билета Целое, Ном_переплета Целое, Дата_выдачи Дата,
        Срок Целое, Дата_возврата Дата )
ОГРАНИЧЕНИЯ ( Значения полей Ном_билета и Ном_переплета должны
                принадлежать набору значений соответствующих полей таблиц
                Читатели и Переплеты; при нарушении вывод сообщения
                "Такого читателя нет" или "Такого переплета нет" );
```

Теперь следует проверить, не нарушены ли в данном проекте какие-либо принципы нормализации, т.е. что любое неключевое поле каждой таблицы:

- функционально зависит от полного первичного ключа, а не от его части (если ключ составной);
- не имеет функциональной зависимости от другого неключевого поля.
- Сущности Авторы, Составители, Редакторы, Художники и Переиздания, не имеющие неключевых полей, безусловно нормализованы. Нормализованы и сущности Создатели, Характеры, Заглавия, Вид_издания и Аннотации, состоящие из несоставного ключа и единственного неключевого поля.

Анализ сущностей Переводчики, Размещение и Выдача, состоящих из составного ключа и неключевых полей, показал, что в них нет функциональных связей между неключевыми полями. Последние же не зависят функционально от какой-либо части составного ключа.

Наконец, анализ сущностей Издания, Переплеты, Места, Читатели и Языки, показал, что единственной "подозрительной" сущностью является стержень Языки, имеющий два функционально связанных неключевых поля: Язык и Сокращение.

Поле Язык стало неключевым из-за ввода цифрового первичного ключа Код_языка, заменяющего текстовый возможный ключ Язык. Это позволило уменьшить объем хранимых данных в таблице Переводчики, затраты труда на ввод множества текстовых значений и возможной противоречивости, которая часто возникает из-за ввода в разные поля ошибочных дубликатов (например, "Английский", "Англиский", "Анлийский", "Английский" и т.п.).

Процесс проектирования БД на основе принципов нормализации представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели.

Процесс проектирования длительный и требует обсуждений с заказчиком и со специалистами в предметной области. Наконец, при разработке серьезных

корпоративных информационных систем проект базы данных является тем фундаментом, на котором строится вся система в целом, и вопрос о возможном кредитовании часто решается экспертами банка на основании именно грамотно сделанного инфологического проекта БД. Следовательно, инфологическая модель должна включать такое формализованное описание предметной области, которое легко будет «читаться» не только специалистами по базам данных. И это описание должно быть настолько емким, чтобы можно было оценить глубину и корректность проработки проекта БД, и конечно, оно не должно быть привязано к конкретной СУБД. Выбор СУБД – это отдельная задача, для корректного ее решения необходимо иметь проект, который не привязан ни к какой конкретной СУБД.

2. Порядок выполнения работы

1. Изучить теоретическую часть по приведенным выше данным и дополнительной литературе.
2. Просмотреть демонстрационный пример.
3. Получить у преподавателя вариант задания для выполнения.
4. Проанализировать полученное задание.
5. Определить характер решаемой задачи.
6. Построить инфологическую модель на языке "Таблицы-связи".
7. Спроектировать базу данных и представить базовой таблицей.
8. Ввести в описания таблиц дополнительные сведения об ограничениях целостности (выше указан лишь минимальный их набор) и дать описание таблиц.

3. Варианты заданий

Спроектировать базу данных для следующих автоматизированных информационных систем:

По выбору: система по отысканию рыночных ниш; системы, ускоряющие потоки товаров; системы по снижению издержек производства, системы

автоматизации технологии ("менеджмент уступок"); управление производственным процессом; системы документооборота; оперативное управление предприятием, информационная интеллектуальная система, обучающая система, информационная библиотечная система и т.д.

4. Контрольные вопросы

1. Как можно представить инфологическую модель?
2. Что представляет собой сущности?
3. Каким образом представляются базовые таблицы?
4. Каким образом строятся даталогическая и физическая модели?
5. Опишите принципы нормализации?

Практическая работа № 3

ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ (ИС)

Цель работы: Научиться проектировать информационную систему на основе архитектуры «Файл-сервер», «Клиент-сервер», многоуровневой архитектуры или архитектуры " INTRANET.

1. Теоретическая часть

Архитектура ИС – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы. Архитектура информационной системы должна выбираться с учетом нужд бизнеса, а не личных пристрастий разработчиков. Архитектура системы должна строиться еще на стадии технико-экономического обоснования системы.

Конструктивно архитектура обычно определяется как набор ответов на следующие вопросы:

— что делает система;

- на какие части она разделяется;
- как эти части взаимодействуют;
- где эти части размещены.

Таким образом, архитектура ИС является логическим построением, или моделью, и влияет на совокупную стоимость владения через набор связанных с ней решений по выбору средств реализации, СУБД, операционной платформы, телекоммуникационных средств и т. п. — то есть через то, что мы называем инфраструктурой ИС.

Файл-серверная архитектура

Данная архитектура (рис.3.1) наиболее распространена по следующим причинам:

- сокращение автономности ПО клиента
- простота организации
- сохранение привычных условий для пользователя

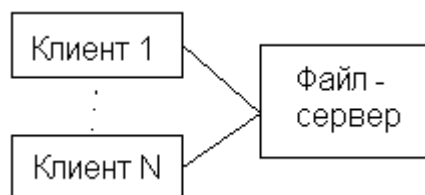


Рис.3.1

При всей простоте организации необходимо учитывать особенности:

- при работе с БД необходимо учитывать особенности каждой выбранной СУБД;
- необходимо учитывать особенности поддержания целостности и надежности хранимой БД. Для этого должно выполняться:

- Наличие транзакционного управления
- Хранение избыточных данных
- Возможность введения ограничений целостности и их проверку

В целом, в файл-серверной архитектуре мы имеем простое приложение, работающее с небольшими объемами данных и рассчитанное на однопользовательскую работу.

Клиент-серверная архитектура

Под клиент-серверной архитектурой (рис.3.2) понимается ИС, основанная на применении серверов БД и имеющая следующие характеристики:

- На стороне «клиент» выполняется ввод приложения, в который обязательно входят компоненты, поддерживающие интерфейс с пользователем;
- Клиентская часть приложения взаимодействует с клиентской частью СУБД, которая является представителем СУБД для приложения.

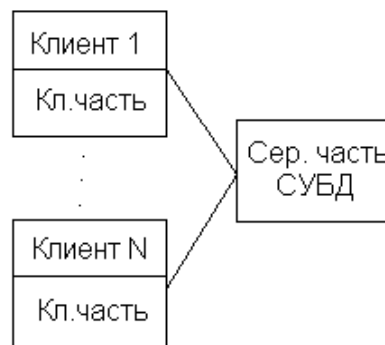


Рис.3.2

Интерфейс (ИФ) между клиентской частью приложения и клиентской частью сервера БД основан на использовании языка SQL. Функции предварительной обработки, предназначенные для запросов к БД или формирования конечных отчетов, выполняются в коде приложения. Клиентская часть сервера БД, используя средства сетевого доступа, обращается к серверу БД, передавая ему текст на языке SQL. На стороне сервера БД происходит компиляция полученных операторов на языке SQL, далее выполняются операторы, результаты обработки передаются клиентской части приложения и в конечном итоге – пользователю.

В клиент-серверной архитектуре клиенты могут быть достаточно «тонкими», а сервер должен быть настолько «толстым», чтобы удовлетворить запросы всех клиентов.

С другой стороны, пользователи данной архитектуры часто бывают не удовлетворены, в виду постоянно растущих сетевых расходов. Но на практике,

как правило, пользователю необходим какой-то кусок БД, что приводит к идее поддержки локального кэша общей БД.

Концепция локального кэширования является частным случаем концепции реплицирования данных. Для поддержки локального кэша необходим компонент СУБД, который должен обеспечивать однопользовательский режим.

Для поддержки локального кэша необходимо выполнять его синхронизацию с основной БД. Это достигается двумя путями:

-автоматическая синхронизация, путем использования триггеров (через определенные промежутки времени)

-в ручном режиме

Архитектура клиент-сервер является более дорогой по сравнению с файл-серверной, требуется более мощная аппаратура для сервера и более развитые СУБД.

Однако наряду с этим недостатком, существует основное достоинство данной архитектуры, заключающееся в масштабируемости системы и способности к развитию.

Internet – приложение

IntraInternet – архитектура базируется на протоколе http. Данная служба получила широкое распространение по ряду причин:

- Достаточно просто разработать ИС, а затем осуществить доступ по данному методу
- Наличие готовых клиентских частей (браузеры)

Наряду с этими достоинствами существует ряд ограничений:

- В данной ИС отсутствует прикладная обработка данных,
- Гипертекстовые структуры сложно модифицируются
- Пользователь имеет право только просматривать данные
- Недостаточно организован поиск информации. Однако все эти причины могут быть разрешены с использованием более развитых механизмов WEB-технологий.

На настоящий момент логика приложения реализуется с помощью двух подходов: CGI, API. Оба подхода основываются на наличии специальных конструкций, информирующих клиента браузера, что ему следует послать веб-серверу специальное сообщение. При получении сообщения сервер должен вызвать внешнюю процедуру, получить ее результаты и вернуть их клиенту в стандартном виде (рис.3.3).

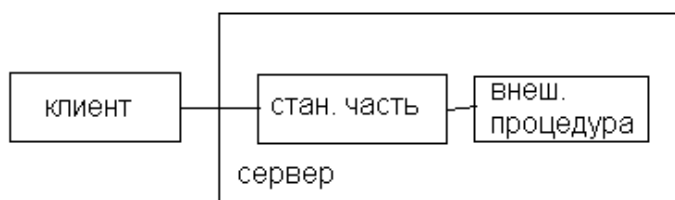


Рис.3.3

Данный метод доступа используется для обеспечения доступа к БД в Интернет системах. Язык HTML позволяет вставлять в документ необходимые формы. Когда браузер встречает форму, он предлагает пользователю заполнить её, а затем посылает серверу сообщение, которое содержит введенные параметры. Как правило, к сообщению приписывается внешняя процедура сервера. При получении сообщения от клиента сервер вызывает указанную процедуру и передает ей параметры, введенные пользователем. Таким образом, процедура играет роль шлюза между веб-сервером и сервером БД (рис.3.4).

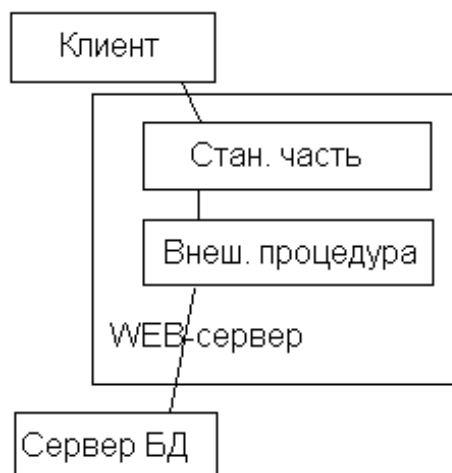


Рис.3.4

Системы оперативной обработки данных (OLAP - системы)

OLAP системы – аналитические системы, помогающие принимать бизнес-решения за счет динамически проводимых анализов моделирования или прогнозирования данных (рис.3.5).

Данные системы используют в случаях:

- 1 Если основным источником информации является деятельность какой-либо корпорации
- 2 Если для оперативной обработки требуются свежие данные
- 3 Если одновременно существуют несколько ОС
- 4 Если БД являются сильно изменчивыми
- 5 Если информация БД настолько критична для корпорации, что для ее защиты требуются более тонкие приемы защиты.

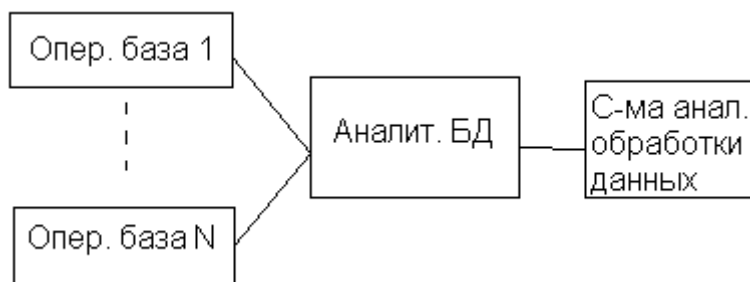


Рис.3.5

Склады данных (Data Ware Housing)

Хранилища такого типа наиболее актуальны. Они характеризуются большим объемом редко изменяемых данных. Хранилища могут обмениваться информацией с другими хранилищами (VLBD – очень большие хранилища данных)

При проектировании данных систем необходимо уделять внимание интерфейсам обмена данных, а также конвертациям данных. Внутренняя структура хранилищ рассчитана на выполнение сложных запросов. Для работы хранилища используется выделенный сервер.

Хранилища используются для прогнозирования развития бизнеса (рис.3.6). Они оснащаются средствами аналитической обработки информации. Хранилища состоят из:

1. Программное обеспечение источников данных

2. Загрузочная секция (отвечает за трансформацию информации)
3. Программное обеспечение трансформации данных (пересылка, проверка на корректность)
4. Хранилища данных (несколько параллельных серверов БД, которые обеспечивают переработку информации и хранение)
5. Интерфейс клиента

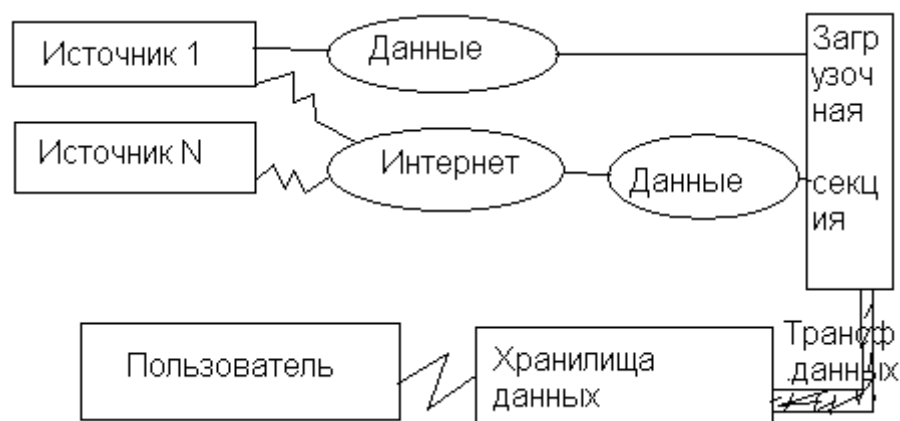


Рис.3.6

2. Порядок выполнения работы

1. Изучить теоретическую часть по приведенным выше данным и дополнительной литературе.
2. Просмотреть демонстрационный пример.
3. Получить у преподавателя вариант задания для выполнения.
4. Проанализировать полученное задание.
5. Определить характер решаемой задачи.
6. Спроектировать информационную систему на основе вышеописанных архитектур
7. Обосновать выбор архитектуры ИС.

3. Варианты заданий

Спроектировать информационную систему на основе выбранной архитектуры для следующих задач: По выбору: система по отысканию рыночных ниш; системы, ускоряющие потоки товаров; системы по снижению издержек производства, системы автоматизации технологии ("менеджмент

уступок"); управление производственным процессом; системы документооборота; оперативное управление предприятием, информационная интеллектуальная система, обучающая система, информационная библиотечная система и т.д.

4. Контрольные вопросы

1. Что предусматривает архитектура ИС?
2. Отличие «клиент-серверной» архитектуры от архитектуры «файл-сервер»
3. Каким преимуществом обладает архитектура "клиент-сервер"?
4. Перечислите недостатки клиент-серверного подхода?
5. Что предусматривает многоуровневая архитектура?
6. Достоинства и недостатки многоуровневой архитектуры?

5. Литература

1. Бекаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2002. – СПб.: БХВ-СПб., 2003. – 720 с.
2. Голицина О.Л., Максимов Н.В., Попов И.И. Базы данных: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2003. – 352 с.
3. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2002. – 304 с.
4. Петров В.Н. Информационные системы. – СПб.: Питер, 2003. – 688 с.
5. <http://www.osp.ru/cio/2002/06/172179/>

Практическая работа № 4

ПРОЕКТИРОВАНИЕ СИСТЕМНОЙ АРХИТЕКТУРЫ ИС (СОСТАВ ОБЕСПЕЧИВАЮЩИХ ПОДСИСТЕМ)

Цель работы: Научиться проектировать системную архитектуру информационной системы, выделить элементы и модули информационного, технического, программного обеспечения и других обеспечивающих подсистем.

1. Теоретическая часть

Параллельно с проектированием схемы базы данных выполняется проектирование процессов, чтобы получить спецификации (описания) всех модулей ИС. Оба эти процесса проектирования тесно связаны, поскольку часть бизнес-логики обычно реализуется в базе данных (ограничения, триггеры, хранимые процедуры). Главная цель проектирования процессов заключается в отображении функций, полученных на этапе анализа, в модули информационной системы. При проектировании модулей определяют интерфейсы программ: разметку меню, вид окон, горячие клавиши и связанные с ними вызовы.

Обеспечивающие подсистемы являются общими для всей ИС независимо от конкретных функциональных подсистем, в которых применяются те или иные виды обеспечения.



Рис. 4.1. Обеспечивающие подсистемы ИС

Состав обеспечивающих подсистем не зависит от выбранной предметной области и имеет (рис. 4.1.):

- функциональную структуру;
- информационное обеспечение;

- математическое (алгоритмическое и программное) обеспечение;
- техническое обеспечение;
- организационное обеспечение;
- кадровое обеспечение,

а на стадии разработки ИС дополнительные обеспечения:

- правовое;
- лингвистическое;
- технологическое;
- методологическое;
- интерфейсы с внешними ИС.

Функциональная структура представляет собой перечень реализуемых ею функций (задач) и отражает их соподчиненность. Под функцией ИС понимается круг действия ИС, направленных на достижение частной цели управления.

Подсистема информационного обеспечения (ИО) является одной из важнейших и включает в себя всю совокупность информации, циркулирующую на объекте, а также отражает процессы ее сбора, преобразования и использования и служит основой связи объекта с внешней средой. Информационное обеспечение – это совокупность средств и методов построения информационной базы. Оно определяет способы и формы отображения состояния объекта управления в виде данных внутри ИС, документов, графиков и сигналов вне ИС. Состав ИО:

- компоненты вне машинного ИО (классификаторы, документы);
- внутримашинное ИО – макеты экранных форм для ввода первичной информации и вывода результативной, структура информационной базы, базы данных.

Центральный компонент ИО – БД, через которую осуществляется обмен данными различных задач.

Подсистема математического обеспечения состоит из совокупности математических методов, моделей и алгоритмов обработки информации, используемых при создании системы.

Подсистема программного обеспечения включает в себя все многообразие типовых и стандартных программ и процедур, пакетов прикладных программ (ППП), реализующих решение задач на компьютере для всех функциональных подсистем ИС. ПО делится на два комплекса:

- общее – ОС, операционные оболочки, компиляторы, интерпретаторы, программные среды, СУБД, сетевые программы и т.п.;
- специальное – совокупность прикладных программ, разработанных для конкретных задач в рамках функциональных подсистем, контрольные примеры.

Технологическое обеспечение — это совокупность проектных решений, определяющих технологию обработки информации на всех технологических этапах:

- сбора и регистрации первичной информации;
- подготовки и контроля файлов и баз данных;
- передачи информации;
- арифметической и логической обработки;
- накопления и хранения;
- выпуска выходных документов.

Подсистема лингвистического обеспечения представляет собой совокупность научно-технических терминов и языковых средств, используемых в целях облегчения общения персонала с компьютерами и другими средствами вычислительной техники. Языковые средства делятся на 2 группы:

- традиционные языки – естественные, математические, алгоритмические, языки моделирования;
- языки, предназначенные для диалога с ЭВМ – информационно-поисковые языки СУБД, языки операционных сред, входные языки пакетов прикладных программ.

Организационное обеспечение – это совокупность средств и методов организации производства и управления ими в условиях внедрения ИС.

В неё входят:

1. Документы, регламентирующий процесс создания и функционирования АИС.
2. Совокупность средств, необходимых для эффективного проектирования и функционирования ЭИС (общесистемные и отраслевые классификаторы, унифицированные системы документов, типовые пакеты прикладных программ (Офис), типовые структуры управления предприятием.
3. Техническая документация, получаемая в процессе обследования, проектирования и внедрения системы:
 - технико-экономическое обоснование;
 - техническое задание;
 - технический и рабочий проекты и документы;
4. Состав пользователей АИС. (в первую очередь персонал, организационно-кадровая структура проекта, определяющая, в частности, состав главных конструкторов и специалистов).

Целью организационного обеспечения является: выбор и постановка задач управления, анализ системы управления и путей ее совершенствования, разработка решений по организации взаимодействия ИС и персонала, внедрение задач управления. Организационное обеспечение включает в себя методики проведения работ, требования к оформлению документов, должностные инструкции и т. д. Это обеспечение является одной из важнейших подсистем ИС, от которой зависит успешная реализация целей и функций системы.

Все обеспечивающие подсистемы связаны между собой и с функциональными подсистемами

2. Порядок выполнения работы

1. Изучить теоретическую часть по приведенным выше данным и дополнительной литературе.
2. Просмотреть демонстрационный пример.

3. Получить у преподавателя вариант задания для выполнения.
4. Проанализировать полученное задание.
5. Определить характер решаемой задачи.
6. Спроектировать системную архитектуру информационной системы.
7. Привести характеристику обеспечивающих подсистем ИС.
8. Привести интерфейсы программ: разметку меню, вид окон, горячие клавиши и связанные с ними вызовы

3. Варианты заданий

Спроектировать системную архитектуру информационной системы и описать обеспечивающие подсистемы для следующих задач: По выбору: система по отысканию рыночных ниш; системы, ускоряющие потоки товаров; системы по снижению издержек производства, системы автоматизации технологии ("менеджмент уступок"); управление производственным процессом; системы документооборота; оперативное управление предприятием, информационная интеллектуальная система, обучающая система, информационная библиотечная система и т.д.

4. Контрольные вопросы

1. Приведите основные положения (стадии) системного подхода как инструмента решения проблем?
2. Опишите функциональную структуру информационной системы организации?
3. Дайте характеристику обеспечивающих подсистем информационной системы?
4. Опишите состав обеспечивающих подсистем информационной системы?

Практическая работа № 5

РАЗРАБОТКА ТЕХНИЧЕСКОГО ПРОЕКТА ИС

Цель работы: Научиться разрабатывать технический проект информационной системы.

1. Теоретическая часть

Этап проектирования завершается разработкой технического проекта ИС.

Технический проект системы - это техническая документация, содержащая общесистемные проектные решения, алгоритмы решения задач, а также оценку экономической эффективности *автоматизированной системы управления* и перечень мероприятий по подготовке объекта к внедрению.

На этом этапе осуществляется комплекс научно-исследовательских и экспериментальных работ для выбора основных проектных решений и расчет экономической эффективности системы.

Состав и содержание *технического проекта* приведены в [таблице 5.1](#).

Таблица 5.1. Содержание технического проекта

№п\п	Раздел	Содержание
1	Пояснительная записка	• основания для разработки системы • перечень организаций разработчиков • краткая характеристика объекта с указанием основных <i>технико-экономических показателей</i> его функционирования и связей с другими объектами • краткие сведения об основных проектных решениях по функциональной и обеспечивающим частям системы
2	Функциональная и организационная структура системы	• обоснование выделяемых подсистем, их перечень и назначение • перечень задач, решаемых в каждой подсистеме, с краткой характеристикой их содержания • схема информационных связей между подсистемами и между задачами в рамках каждой подсистемы

3	Постановка задач и алгоритмы решения	• организационно-экономическая сущность задачи (наименование, цель решения, краткое содержание, метод, периодичность и время решения задачи, способы сбора и передачи данных, связь задачи с другими задачами, характер использования результатов решения, в которых они используются) • экономико-математическая модель задачи (структурная и развернутая форма представления) • входная оперативная информация (характеристика показателей, диапазон изменения, формы представления) • нормативно-справочная информация (НСИ) (содержание и формы представления) • информация, хранимая для связи с другими задачами • информация, накапливаемая для последующих решений данной задачи • информация по внесению изменений (система внесения изменений и перечень информации, подвергающейся изменениям) • алгоритм решения задачи (последовательность этапов расчета, схема, расчетные формулы) • контрольный пример (набор заполненных данными форм входных документов, условные документы с накапливаемой и хранимой информацией, формы выходных документов, заполненные по результатам решения экономико-технической задачи и в соответствии с разработанным алгоритмом расчета)
4	Организация информационной базы	• источники поступления информации и способы ее передачи • совокупность показателей, используемых в системе • состав документов, сроки и периодичность их поступления • основные проектные решения по организации фонда НСИ • состав НСИ, включая перечень реквизитов, их определение, диапазон изменения и перечень документов НСИ

		• перечень массивов НСИ, их объем, порядок и частота корректировки информации • структура фонда НСИ с описанием связи между его элементами; требования к технологии создания и ведения фонда • методы хранения, поиска, внесения изменений и контроля • определение объемов и потоков информации НСИ • контрольный пример по внесению изменений в НСИ • предложения по унификации документации
5	Альбом форм документов	
6	Система математического обеспечения	• обоснование структуры математического обеспечения • обоснование выбора системы программирования • перечень стандартных программ
7	Принцип построения комплекса технических средств	• описание и обоснование схемы технологического <i>процесса обработки данных</i> • обоснование и выбор структуры комплекса технических средств и его <i>функциональных групп</i> • обоснование требований к разработке нестандартного оборудования • комплекс мероприятий по обеспечению надежности функционирования технических средств
8	Расчет экономической эффективности системы	• сводная <i>смета</i> затрат, связанных с эксплуатацией систем • расчет годовой экономической эффективности, источниками которой являются оптимизация производственной структуры хозяйства (объединения), снижение себестоимости продукции за счет рационального использования производственных ресурсов и уменьшения потерь, улучшения принимаемых управленческих решений

9	Мероприятия по подготовке объекта к внедрению системы	• перечень организационных мероприятий по совершенствованию бизнес-процессов • перечень работ по внедрению системы, которые необходимо выполнить на стадии рабочего проектирования, с указанием сроков и ответственных лиц
1	Ведомость документов	

В завершение стадии технического проектирования производится разработка документации на поставку серийно выпускаемых изделий для комплектования ИС, а также определяются технические требования и составляются ТЗ на разработку изделий, не изготавливаемых серийно.

2. Порядок выполнения работы

1. Изучить теоретическую часть по приведенным выше данным и дополнительной литературе.
2. Просмотреть демонстрационный пример.
3. Получить у преподавателя вариант задания для выполнения.
4. Проанализировать полученное задание.
5. Разработать технический проект информационной системы

3. Варианты заданий

Разработать технический проект для следующих ИС: По выбору: система по отысканию рыночных ниш; системы, ускоряющие потоки товаров; системы по снижению издержек производства, системы автоматизации технологии ("менеджмент уступок"); управление производственным процессом; системы документооборота; оперативное управление предприятием, информационная интеллектуальная система, обучающая система, информационная библиотечная система и т.д.

4. Контрольные вопросы

1. Как можно представить технический проект?
2. Состав технического проекта?
3. Содержание технического проекта?
4. Что происходит в завершение стадии технического проектирования?

