МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное общеобразовательное учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Алгоритмы и структуры данных»

на тему «Синтез структур данных на основе анализа инфологической модели предметной области»

Выполнил: студент группы Б03-191-3 Р.А. Гумметов

Принял: д.т.н., профессор М.А. Сенилов

Ижевск 2019

1. Цель работы

Цель работы: приобретение навыков анализа, структурирования и формализации информационного пространства, ограниченного предметной областью; применения ER-моделей и методик их анализа для синтеза необходимых структур данных.

2. Задание

Для заданной предметной области, используя методику построения ER-моделей (модель "сущность-связь"), разработать инфологическую модель (схему), на основе анализа которой синтезировать необходимые структуры данных и описать их на языке программирования.

3. Построение инфологической модели предметной области

3.1 Описание предметной области

У ГИБДД есть три наиболее важные функциональные задачи:

* регистрация автотранспортных средств при совершении сделки купли-продажи;
* разработка мер, повышающих безопасность дорожного движения и выполнение всех мер при совершении ДТП (дорожно-транспортное происшествие) на улицах города (регистрация, разбор, выявление виновных, автоэкспертиза и т.п.);
* борьба с угоном автотранспортных средств, оперативный поиск угнанных машин и задержание преступников.

ГИБДД занимается выделением и учетом номерных знаков на автотранспорт. К автотранспортным средствам относятся легковые, грузовые автомобили, прицепы, полуприцепы, мотоциклы, тракторы, автобусы, микроавтобусы. На разные виды транспорта выдаются разные виды номеров и в базу данных заносятся разные характеристики. Номера могут выделяться как частным владельцам, так и организациям. В справочнике номеров, выданных частным владельцам, фиксируется: номер, ФИО владельца, его адрес, марка автомобиля, дата выпуска, объем двигателя, номера двигателя, шасси и кузова, цвет и т.п. В справочнике номеров, выданных организации, дополнительно фиксируется: название организации, район, адрес, руководитель. Существует справочник свободных номеров (серия, диапазон номеров). ГИБДД периодически проводит технический осмотр (ТО) машин. Для прохождения техосмотра необходима квитанция об оплате налогов, сумма оплаты зависит от объема двигателя. Периодичность прохождения зависит от года выпуска и вида транспортного средства. Технические характеристики, проверяемые на ТО и допуски также зависят от вида транспортного средства.

ГИБДД занимается учетом и анализом ДТП (дорожно-транспортное происшествие). При регистрации ДТП фиксируется: дата, тип происшествия (наезд на пешехода, наезд на ограждение либо столб, лобовое столкновение, наезд на впереди стоящий транспорт, боковое столкновение на перекрестке и т.п.), место происшествия, марки пострадавших автомобилей, государственный номер, тип машины (легковая, грузовая, специальная), краткое содержание, число пострадавших, сумма ущерба, причина, дорожные условия и т.п. Анализ накопленной по ДТП статистике поможет правильно расставить запрещающие и предупреждающие знаки на улицах города, а так же спланировать местонахождение постов патрульных.

Угон либо исчезновение виновника ДТП с места происшествия требует оперативного вмешательства всех постов ГИБДД и патрульных машин. Для информирования о разыскиваемой машине ее данные (включая номера двигателя и кузова) извлекаются из базы зарегистрированных номеров и передаются по рации всем постам. Ведение статистики угонов, ее анализ и опубликование результатов в СМИ поможет снизить количество угонов, а хозяевам машин принять необходимые меры (самые угоняемые марки, самый популярный способ вскрытия, самые надежные сигнализации и т.п.).

3.1.1 Описание объектов предметной области

Анализ предметной области позволяет выделить объекты и их атрибуты. Для описания атрибутов объектов используются характеристики, представленные в табл. 2.1.

Таблица 2.1

Характеристики атрибутов объектов предметной области

|  |  |
| --- | --- |
| Код характеристики | Название характеристики |
| А1 | Шаблон значений атрибутов |
| А2 | Процент наличия значений атрибута  в экземплярах объекта |
| А3 | Ограничения на доступ к значениям атрибута |
| А4 | Частота использования атрибута |
| А5 | Область допустимых значений |
| А6 | Признак выводимости значений |
| А7 | Признак дублирования значений |

Таблица 2.2

Описание объекта СОТРУДНИК

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя атрибута | А1 | А2 | А3 | А4 | А5 | А6 | А7 | Роль атрибута |
| ФИО | X(30) | - | - | - | - | - | Да | Фамилия, имя , отчество |
| Дата рождения | 9(6) | - | - | - | - | - | Да | Дата рождения сотрудника |
| Должность | X(30) | - | - | - | - | - | Да | Название должности |

Таблица 2.3

Описание объекта УЧАСТОК

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя атрибута | А1 | А2 | А3 | А4 | А5 | А6 | А7 | Роль атрибута |
| Номер | 9(3) | - | - | - | - | - | - | Номер участка |
| Территориальное расположение | X(30) | - | - | - | - | - | - | Территориальное расположение участка |

Таблица 2.4

Описание объекта ГОРОД

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя атрибута | А1 | А2 | А3 | А4 | А5 | А6 | А7 | Роль атрибута |
| Название | X(30) | - | - | - | - | - | - | Название города |

Таблица 2.5

Описание объекта ВОДИТЕЛЬ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя атрибута | А1 | А2 | А3 | А4 | А5 | А6 | А7 | Роль атрибута |
| ФИО | X(30) | - | - | - | - | - | Да | Фамилия, имя , отчество |
| Дата рождения | 9(6) | - | - | - | - | - | Да | Дата рождения водителя |

Таблица 2.6

Описание объекта АВТОМОБИЛЬНЫЙ НОМЕР

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя атрибута | А1 | А2 | А3 | А4 | А5 | А6 | А7 | Роль атрибута |
| Дата получения | 9(6) | - | - | - | - | - | Да | Дата получения автомобильного номера |
| Код региона регистрации | 9(3) | - | - | - | - | - | Да | Код региона регистрации автомобильного номера |
| Страна | X(30) | - | - | - | - | - | Да | Страна автомобильного номера |

Перечень характеристик объекта приведен в табл. 2.7. Для выделенных объектов их характеристики приведены в табл. 2.8.

Таблица 2.6

Характеристики объекта

|  |  |
| --- | --- |
| Код характеристики | Название характеристики |
| В1 | Способ обращения  к экземплярам объекта |
| В2 | Структурная активность объекта |
| В3 | Ограничения на доступ  к экземплярам объекта |
| В4 | Частота использования |
| В5 | Количество экземпляров объекта |
| В6 | Изменчивость состава  экземпляров объекта |

Таблица 2.8

Характеристики выделенных объектов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя объекта | В1 | В2 | В3 | В4 | В5 | Примечание |
| СОТРУДНИК | K(ФИО) | - | - | - | - | - |
| УЧАСТОК | K(НОМЕР) | - | - | - | - | - |
| ГОРОД | K(НАЗВАНИЕ) | - | - | - | - | - |
| ВОДИТЕЛЬ | K(ФИО) | - | - | - | - | - |
| АВТОМОБИЛЬНЫЙ НОМЕР | K(НОМЕР) | - | - | - | - | - |

3.1.2. Описание процессов предметной области

Процессы предметной области определяются запросами к информационной базе:

1) Выдать список НОМЕРОВ АВТОМОБИЛЕЙ, остановленных данным СОТРУДНИКОМ.

2) Выдать список НОМЕРОВ АВТОМОБИЛЕЙ, нарушивших на данном УЧАСТКЕ.

3) Выдать список СОТРУДНИКОВ, работающих на данном УЧАСТКЕ в определенный день. В определенный день каждый сотрудник работает только на одном участке.

4) Выдать список ВОДИТЕЛЕЙ, остановленных данным СОТРУДНИКОМ на данном УЧАСТКЕ в определенный день.

5) Выдать список УЧАСТКОВ, на которых закреплен данный СОТРУДНИК, отправленный на повышение квалификации в данный ГОРОД. Сотрудник может быть закреплен только на одном участке. На повышение квалификации сотрудник может быть отправлен только в один город.

6) Выдать список ВОДИТЕЛЕЙ, задержанных на данном УЧАСТКЕ в данном ГОРОДЕ в определенный момент.

7) Выдать список УЧАСТКОВ, находящихся в данном ГОРОДЕ.

8) Выдать список УЧАСТКОВ, на которых останавливали данного ВОДИТЕЛЯ в данном ГОРОДЕ.

9) Выдать список СОТРУДНИКОВ , задерживавших автомобиль с данным АВТОМОБИЛЬНЫМ НОМЕРОМ в данном ГОРОДЕ.

10) Выдать список ГОРОДОВ, в которых данного ВОДИТЕЛЯ задерживали на данном УЧАСТКЕ.

3.2 Формализация процессов предметной области с помощью функциональных связей (ФС); приведение ФС к каноническому виду

Формализация запросов (процессов) осуществляется путем их описания функциональными связями (ФС).

Многомерная ФС между исходными объектами *X1, X2,…, Xn* и конечным объектом *Y* обозначается следующим образом:.

Полученные функциональные связи приводятся к каноническому виду cпомощью преобразований 1 и 2, нумеруются и заносятся в таблицу «Перечень функциональных связей».

В дальнейшем через *Т(А, В)* будем обозначать тип соответствия между объектами *А* и *В*. Он может принимать значения: 1:1 («один к одному»), 1:*М* («один ко многим»), *М*:1 («многие к одному»), *М:М* («многие ко многим»).

Ниже представлены описания запросов функциональными связями и показано приведение их к каноническому виду.

1) Запрос представлен следующей одномерной ФС:

T(СОТРУДНИК, АВТОМОБИЛЬНЫЙ НОМЕР) = M:M (определенный сотрудник может остановить несколько автомобилей, определенный автомобиль могут остановить несколько сотрудников).

2) Запрос представлен следующей одномерной ФС:

T(УЧАСТОК, АВТОМОБИЛЬНЫЙ НОМЕР) = M:M (определенный автомобиль может нарушить на нескольких участках, на определенном участке могут нарушить несколько автомобилей).

3) Запрос представлен следующей одномерной ФС:

T(УЧАСТОК, СОТРУДНИК) = 1:M (на определенном участке в определенный день может работать несколько сотрудников, определенный сотрудник в определенный день может работать только на одном участке).

4) Запрос представлен следующей многомерной ФС:

T(УЧАСТОК, СОТРУДНИК) = 1:M (на определенном участке в определенный день может работать несколько сотрудников, определенный сотрудник в определенный день может работать только на одном участке).

Применим преобразование 1:

T(СОТРУДНИК, ВОДИТЕЛЬ) = M:M (сотрудник в определенный день может остановить нескольких водителей, водителя в определенный день может остановить несколько сотрудников).

5) Запрос представлен следующей многомерной ФС:

T(ГОРОД, СОТРУДНИК) = 1:M (на повышение квалификации сотрудник может быть отправлен только в один город, в один город на повышение квалификации может быть отправлено несколько сотрудников).

Применим преобразование 1:

T(УЧАСТОК, СОТРУДНИК) = 1:M (сотрудник может быть закреплен только на одном участке, на одном участке может быть закреплено несколько сотрудников).

6) Запрос представлен следующей многомерной ФС:

T(ГОРОД, УЧАСТОК) = 1:M (в определенном городе может быть несколько участков, определенный участок может находиться только в одном городе).

Применим преобразование 1:

T(УЧАСТОК, ВОДИТЕЛЬ) = M:M (определенный водитель может быть остановлен на нескольких участках, на определенном участке может быть остановлено несколько водителей).

7) Запрос представлен следующей одномерной ФС:

T(ГОРОД, УЧАСТОК) = 1:M (в определенном городе может быть несколько участков, определенный участок может находиться только в одном городе).

8) Запрос представлен следующей многомерной ФС:

T(ВОДИТЕЛЬ, ГОРОД) = M:M (определенного водителя могут остановить в нескольких городах, в определенном городе могут остановить нескольких водителей).

T(ВОДИТЕЛЬ, УЧАСТОК) = M:M (определенного водителя могут остановить на нескольких участках, на определенном участке могут остановить нескольких водителей).

Применим преобразование 2:

T(ГОРОД, УЧАСТОК) = 1:M (в определенном городе может быть несколько участков, определенный участок может находиться только в одном городе).

9) Запрос представлен следующей многомерной ФС:

T(ГОРОД, АВТОМОБИЛЬНЫЙ НОМЕР) = 1:M (в определенном городе может быть несколько автомобильных номеров, определенный автомобильный номер может быть только в одном городе).

Применим преобразование 1:

T(АВТОМОБИЛЬНЫЙ НОМЕР, СОТРУДНИК) = M:M (автомобиль с определенным автомобильным номером могут остановить несколько сотрудников, один сотрудник может остановить несколько автомобилей с разными автомобильными номерами).

10) Запрос представлен следующей многомерной ФС:

T (ВОДИТЕЛЬ, УЧАСТОК) = M:M (определенного водителя могут задержать на нескольких участках, на определенном участке могут задержать несколько водителей).

Т (ГОРОД, УЧАСТОК) = 1:М (в определенном городе может быть несколько участков, определенный участок может находиться только в одном городе).

Т (ВОДИТЕЛЬ, ГОРОД) = М:М (определенного водителя могут задержать в нескольких городах, в определенном городе могут задержать несколько водителей).

Установленные функциональные связи сведены в табл. 2.9

Таблица 2.9

Перечень ФС

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № ФС | Исходные объекты | Конечные объекты | Тип соотн. | Параметры выборки | Частота | Огр. на время | Огр  На право |
| 1.1 | Сотрудник | Автомобильный номер | М:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 2.1 | Участок | Автомобильный номер | М:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 3.1 | Участок | Сотрудник | 1:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 4.1 | Участок | Сотрудник | 1:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 4.2 | Сотрудник | Водитель | М:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 5.1 | Город | Сотрудник | 1:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 5.2 | Сотрудник | Участок | М:1 | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 6.1 | Город | Участок | 1:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 6.2 | Участок | Водитель | М:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 7.1 | Город | Участок | 1:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 8.1 | Водитель | Участок | М:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 8.2 | Участок | Город | М:1 | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 9.1 | Город | Автомобильный номер | 1:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 9.2 | Автомобильный номер | Сотрудник | М:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 10.1 | Водитель | Город | М:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |
| 10.2 | Город | Участок | 1:М | Исх.: К(НАЗВАНИЕ)  Кон.: М | - | - | - |

3.3 Отображение функциональных связей в структурные. Построение инфологической схемы предметной области в виде ER-модели

Каждая из функциональных связей из таблицы 2.9 последовательно отображается в структурные связи. Совокупный результат таких отображений представляет собой инфологическую схему предметной области – см. рис. 2.1. Правила отображения функциональных связей в структурные описаны в [2].В соответствии с этими правилами отображения ФС в СС выполняются следующим образом.

1) Анализируется ФС 1.1. Так как тип соответствия T (СОТРУДНИК, АВТОМОБИЛЬНЫЙ НОМЕР) = M:M, то отображение ФС в СС должно выполняться по правилу 3. По этому правилу выделяется объект-связка 1, экземпляр объекта-связки 1 – ПРОТОКОЛ 1 (документ, фиксирующий какое-либо событие, факт или договорённость). Устанавливается СС S1, где владелец – объект СОТРУДНИК, подчиненный – объект ПРОТОКОЛ 1, направление движения С1=ВП. Устанавливается СС S2, где владелец – объект АВТОМОБИЛЬНЫЙ НОМЕР, подчиненный – объект ПРОТОКОЛ 1, направление движения С1=ПВ.

2) Анализируется ФС 2.1. Так как тип соответствия T (УЧАСТОК, АВТОМОБИЛЬНЫЙ НОМЕР) = M:M, то отображение ФС в СС должно выполняться по правилу 3. По этому правилу выделяется объект-связка 2, экземпляр объекта-связки 2 – ПРОТОКОЛ 2 (документ, фиксирующий какое-либо событие, факт или договорённость). Устанавливается СС S3, где владелец – объект УЧАСТОК, подчиненный – объект ПРОТОКОЛ 2, направление движения С1=ВП. Устанавливается СС S4, где владелец – объект АВТОМОБИЛЬЕЫЙ НОМЕР, подчиненный – объект ПРОТОКОЛ 2, направление движения С1=ПВ.

3) Анализируется ФС 3.1. Так как тип соответствия Т (УЧАСТОК, СОТРУДНИК) = 1: М, то отображение ФС в СС должно выполняться по правилу 1. По этому правилу устанавливается СС S5, где владелец – объект УЧАСТОК, подчиненный – объект СОТРУДНИК, направление движения С1=ВП.

4) Анализируется ФС 4.1. Так как тип соответствия T (УЧАСТОК, СОТРУДНИК) = 1:M, то отображение ФС в СС должно выполняться по правилу 1. Владелец – объект УЧАСТОК, подчиненный – объект СОТРУДНИК, направление движения С1=ВП.

5) Анализируется ФС 4.2. Так как тип соответствия T (СОТРУДНИК, ВОДИТЕЛЬ) = M:M, то отображение ФС в СС должно выполняться по правилу 3. По этому правилу выделяется объект-связка 3, экземпляр объекта-связки 3 – ПРОТОКОЛ 3 (документ, фиксирующий какое-либо событие, факт или договорённость). Устанавливается СС S6, где владелец – объект СОТРУДНИК, подчиненный – объект ПРОТОКОЛ 3, направление движения С1=ВП. Устанавливается СС S7, где владелец – объект ВОДИТЕЛЬ, подчиненный – объект ПРОТОКОЛ 3, направление движения С1=ПВ.

6) Анализируется ФС 5.1. Так как тип соответствия Т (ГОРОД, СОТРУДНИК) = 1: М, то отображение ФС в СС должно выполняться по правилу 1. По этому правилу устанавливается СС S8, где владелец – объект ГОРОД, подчиненный – объект СОТРУДНИК, направление движения С1=ВП.

7) Анализируется ФС 5.2. Так как тип соответствия Т (СОТРУДНИК, УЧАСТОК) = М:1, то отображение ФС в СС должно выполняться по правилу 2. По этому правилу корректируются СС S5, новое направление движения С1=ВПВ.

8) Анализируется ФС 6.1. Так как тип соответствия Т (ГОРОД, УЧАСТОК) = 1: М, то отображение ФС в СС должно выполняться по правилу 1. По этому правилу устанавливается СС S9, где владелец – объект ГОРОД, подчиненный – объект УЧАСТОК, направление движения С1=ВП.

9) Анализируется ФС 6.2. Так как тип соответствия T (УЧАСТОК, ВОДИТЕЛЬ) = M:M, то отображение ФС в СС должно выполняться по правилу 3. По этому правилу выделяется объект-связка 4, экземпляр объекта-связки 4 – ПРОТОКОЛ 4 (документ, фиксирующий какое-либо событие, факт или договорённость). Устанавливается СС S10, где владелец – объект УЧАСТОК, подчиненный – объект ПРОТОКОЛ 4, направление движения С1=ВП. Устанавливается СС S11, где владелец – объект ВОДИТЕЛЬ, подчиненный – объект ПРОТОКОЛ 4, направление движения С1=ПВ.

10) Анализируется ФС 7.1. Так как тип соответствия Т (ГОРОД, УЧАСТОК) = 1: М, то отображение ФС в СС должно выполняться по правилу 1. Владелец – объект ГОРОД, подчиненный – объект СОТРУДНИК, направление движения С1=ВП.

11) Анализируется ФС 8.1. Так как тип соответствия T (ВОДИТЕЛЬ, УЧАСТОК) = M:M, то отображение ФС в СС должно выполняться по правилу 3. По этому правилу корректируются СС S10, новое направление движения С1=ВПВ, и S11, новое направление движения С1=ВПВ.

12) Анализируется ФС 8.2. Так как тип соответствия Т (УЧАСТОК, ГОРОД) = М:1, то отображение ФС в СС должно выполняться по правилу 2. По этому правилу корректируются СС S9, новое направление движения С1=ВПВ.

13) Анализируется ФС 9.1. Так как тип соответствия Т (ГОРОД, АВТОМОБИЛЬНЫЙ НОМЕР) = 1: М, то отображение ФС в СС должно выполняться по правилу 1. По этому правилу устанавливается СС S12, где владелец – объект ГОРОД, подчиненный – объект АВТОМОБИЛЬНЫЙ НОМЕР, направление движения С1=ВП.

14) Анализируется ФС 9.2. Так как тип соответствия T (АВТОМОБИЛЬНЫЙ НОМЕР, СОТРУДНИК) = M:M, то отображение ФС в СС должно выполняться по правилу 3. По этому правилу корректируются СС S1, новое направление движения С1=ВПВ, и S2, новое направление движения С1=ВПВ.

15) Анализируется ФС 10.1. Так как тип соответствия T (ВОДИТЕЛЬ, ГОРОД) = M:M, то отображение ФС в СС должно выполняться по правилу 3. По этому правилу выделяется объект-связка 5, экземпляр объекта-связки 5 – ПРОТОКОЛ 5 (документ, фиксирующий какое-либо событие, факт или договорённость). Устанавливается СС S13, где владелец – объект ВОДИТЕЛЬ, подчиненный – объект ПРОТОКОЛ 5, направление движения С1=ВП. Устанавливается СС S14, где владелец – объект ГОРОД, подчиненный – объект ПРОТОКОЛ 5, направление движения С1=ПВ.

16) Анализируется ФС 10.2. Так как тип соответствия Т (ГОРОД, УЧАСТОК) = 1: М, то отображение ФС в СС должно выполняться по правилу 1. Владелец – объект ГОРОД, подчиненный – объект УЧАСТОК, направление движения С1=ВПВ.

Каждой СС присваивается уникальное имя и определяются ее характеристики. Значение этих характеристик определяется в результате анализа предметной области либо на основе характеристик ФС. Перечень характеристик СС приведен в табл. 2.10.

Характеристики структурных связей

|  |  |
| --- | --- |
| Код | Название характеристики |
| С1 | Направление движения по СС |
| С2 | Способ упорядочения экземпляров подчиненного объекта |
| С3 | Ограничения на право движения по СС |
| С4 | Частота использования |
| С5 | Количество экземпляров подчиненного объекта в СС |
| С6 | Класс членства подчиненного объекта |
| С7 | Перемещаемость экземпляров подчиненного объекта СС |
| С8 | Ограничения на время движения по СС |

Перечень структурных связей и их характеристики приведены в табл.2.11.

Таблица 2.11

Перечень структурных связей

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя СС | С1 | С2 | С3 | С4 | С5 | С6 | С7 | С8 |
| S1 | ВП | - | - | - | ПЕР | ОБ | - | - |
| S2 | ПВ | - | - | - | ПЕР | ОБ | - | - |
| S3 | ВП | - | - | - | ПЕР | ОБ | - | - |
| S4 | ПВ | - | - | - | ПЕР | ОБ | - | - |
| S5 | ВПВ | ↑ФИО | - | - | ПЕР | ОБ | - | - |
| S6 | ВП | - | - | - | ПЕР | ОБ | - | - |
| S7 | ПВ | - | - | - | ПЕР | ОБ | - | - |
| S8 | ВП | ↑ФИО | - | - | ПЕР | ОБ | - | - |
| S9 | ВПВ | ↑НОМЕР | - | - | ПЕР | ОБ | - | - |
| S10 | ВПВ | - | - | - | ПЕР | ОБ | - | - |
| S11 | ВПВ | - | - | - | ПЕР | ОБ | - | - |
| S12 | ВП | ↑НОМЕР | - | - | ПЕР | ОБ | - | - |
| S13 | ВП | - | - | - | ПЕР | ОБ | - | - |
| S14 | ПВ | - | - | - | ПЕР | ОБ | - | - |

4. Абстрактный анализ и синтез модели предметной области

4.1. Построение экземпляра инфологической схемы в виде абстрактной структуры данных (АСД) и ее предварительные преобразования

Совокупность структурных связей представляет инфологическую схему предметной области. Для графического изображения структурных связей и инфологической схемы используются ER-диаграммы (диаграммы «сущность-связь») в нотации Баркера. Использование ER-модели позволяет представить инфологическую схему предметной области в виде графа, вершинам которого соответствуют объекты (сущности), а дугам – структурные связи. Инфологическая схема показана на рис. 2.1.

Инфологическая схема предметной области

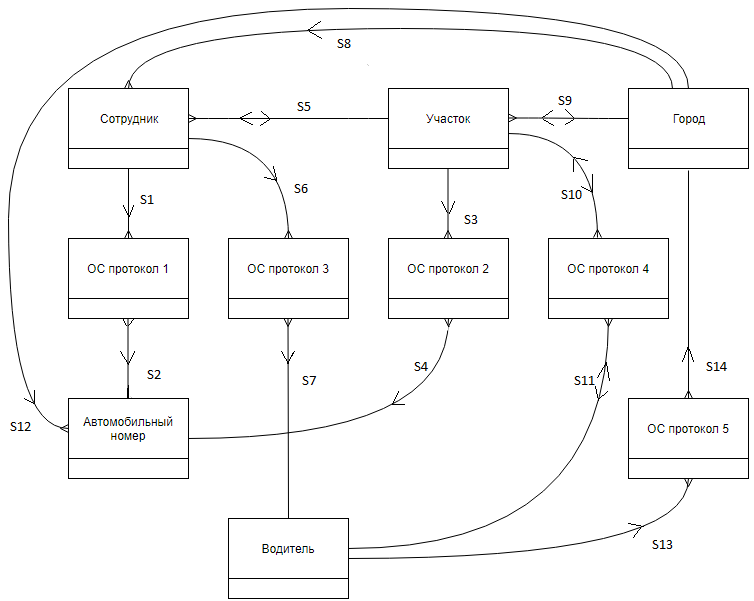


Рис. 2.1

На основе инфологической схемы (рис. 2.1), а также перечней ФС и СС построен экземпляр инфологической схемы, достаточно полно отражающий информационные процессы и режимы работы информационной системы. Экземпляр инфологической схемы представлен на рис 3.1.

Экземпляр инфологической схемы

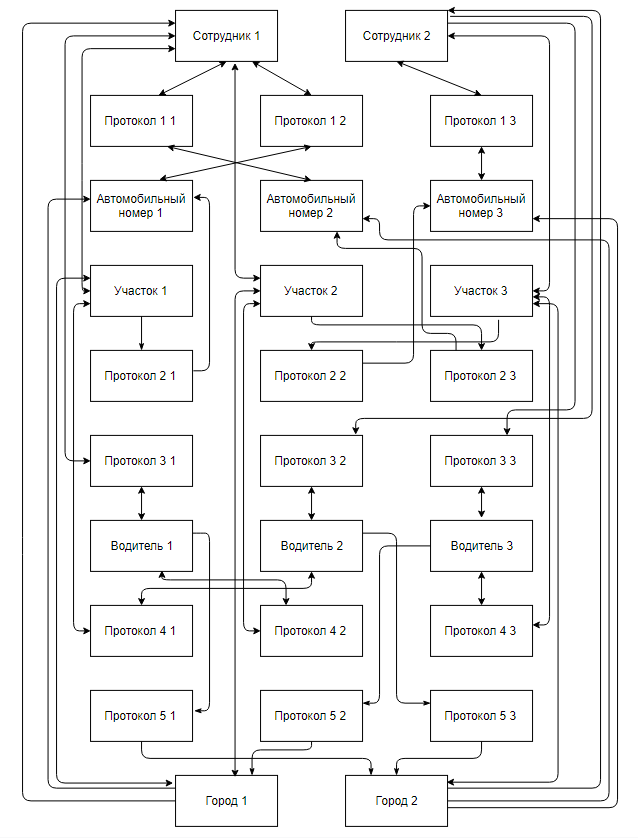


Рис. 3.1

Экземпляр инфологической схемы, показанный на рис. 3.1, представляет собой абстрактную структуру данных типа «граф».

3.2. Предварительные преобразования структуры данных

Предварительные преобразования структуры данных выполняются в соответствии с алгоритмами, изложенными в курсе лекций «Алгоритмы и структуры данных». Исходными данными для выполнения этих алгоритмов являются таблицы «Перечень функциональных связей» (табл. 2.9) и «Перечень структурных связей» (табл. 2.11).

В перечне ФС (табл. 2.9) у всех объектов установлены параметры выборки, поэтому экземпляры каждого из этих объектов связываем в прямую разомкнутую цепь.

На основе перечня СС (табл. 2.11) выполняются следующие преобразования:

1) Структурные связи S1, S3, S6, S8, S12, S13 имеют характеристику С1=ВП. В данных СС каждый экземпляр объекта-владельца и соответствующие ему экземпляры подчиненного объекта связываются в прямую разомкнутую цепь. Экземпляры подчиненного объекта в такой цепи упорядочиваются по возрастанию или убыванию значений ключа.

2) Структурные связи S5, S9, S10, S11имеют характеристику С1=ВПВ. В данных СС каждый экземпляр объекта-владельца и соответствующие ему экземпляры подчиненного объекта связываются в двунаправленную разомкнутую цепь. Экземпляры подчиненного объекта в такой цепи упорядочиваются по возрастанию или убыванию значений ключа.

После применения предварительных преобразований граф экземпляра инфологической схемы преобразуется в размеченный (раскрашенный) граф, показанный на рис. 3.2

Экземпляр инфологической схемы после предварительных преобразований

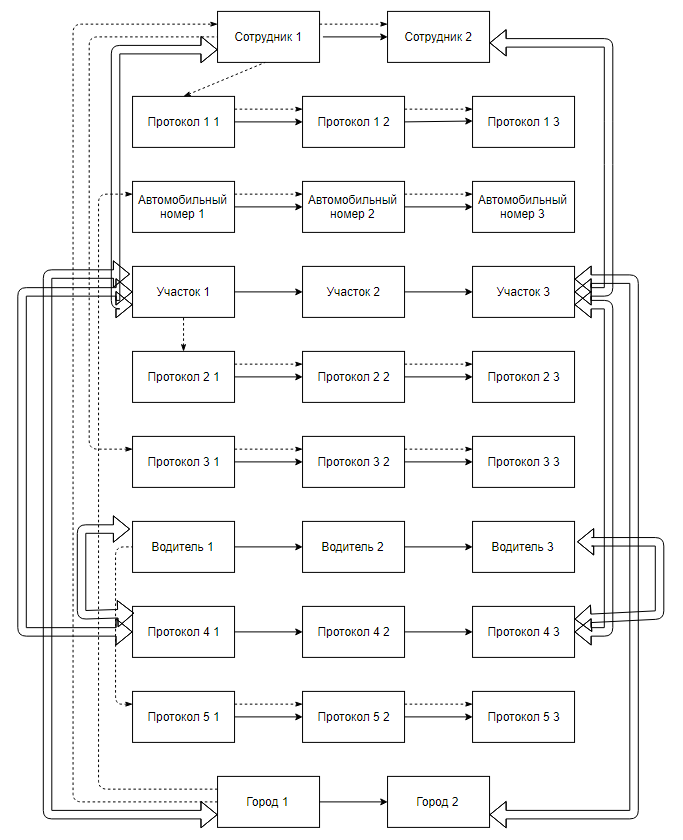


Рис. 3.2

4.2. Декомпозция АСД, синтез новых АСД

Граф экземпляра инфологической схемы (рис. 3.2.), являясь АСД типа «размеченный граф», подлежит декомпозиции с целью получения более простых АСД. Полученные новые АСД представлены на рис. 3.3 – 3.5.

АСД 1

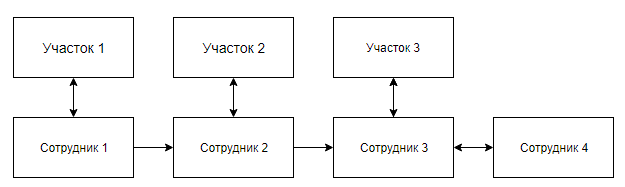


Рис. 3.3

АСД 2

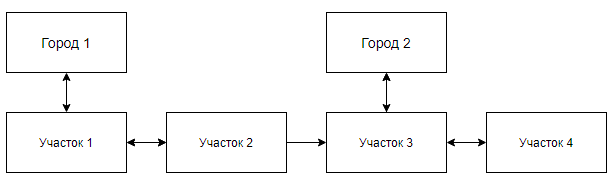


Рис. 3.4

АСД 3

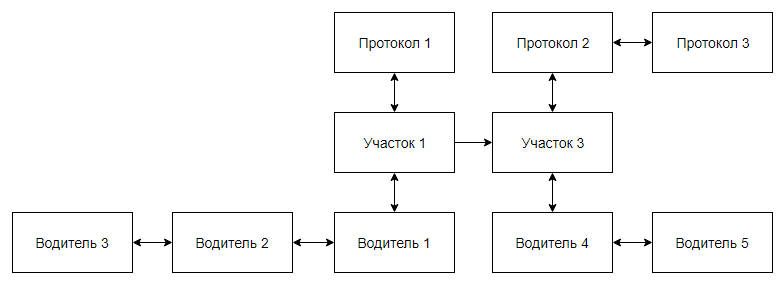
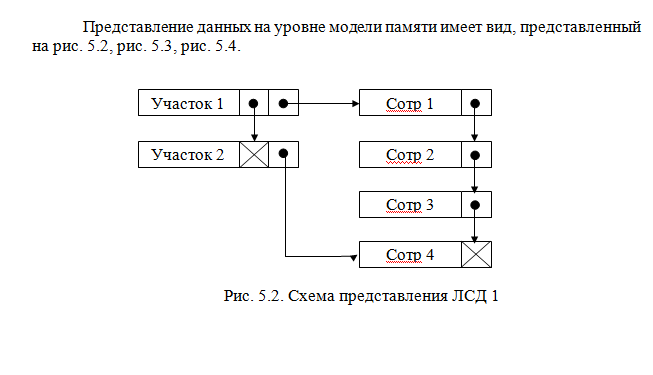
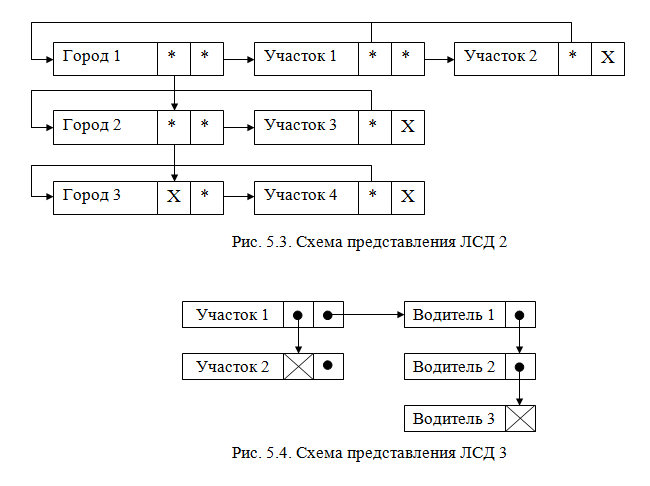


Рис. 3.5

5. Отображение полученных АСД в логические структуры данных (ЛСД) и их представление в модели памяти





6. Программная реализация синтезированных структур данных (описание полученных структур данных на языке программирования)

ТЕКСТ ПРОГРАММЫ

#include <iostream>

#include<algorithm>

#include <string>

#include <iomanip>

#include <map>

#include <conio.h>

#include <stdlib.h>

#include <vector>

using namespace std;

/\*void input(const real\_1d\_array &a, barycentri &p)

{

alglib\_impl::ae\_state \_alglib\_env\_state;

ae\_int\_t n;

double c;

double s;

n = a.length();

c = 0;

s = 1;

alglib\_impl::ae\_state\_init(&\_alglib\_env\_state);

try

{

alglib\_impl::polynomialpow2bar(const\_cast<alglib\_impl::ae\_vector\*>(a.c\_ptr()), n, c, s, const\_cast<alglib\_impl::barycentricinterpolant\*>(p.c\_ptr()), &\_alglib\_env\_state);

alglib\_impl::ae\_state\_clear(&\_alglib\_env\_state);

return;

}

catch(alglib\_impl::ae\_error\_type)

{

throw ap\_error(\_alglib\_env\_state.error\_msg);

}

catch(...)

{

throw;

}

}

string zapros1(vector<char> item, vector<char> universum) {

vector<bool> newArray;

for (int i = 0; i < universum.size(); i++) {

if (IsContain(universum[i], item)) {

newArray.push\_back(true);

}

else {

newArray.push\_back(false);

}

}

return newArray;

}

string zapros2(const double a, const double b, const real\_1d\_array &f, const double t)

{

alglib\_impl::ae\_state \_alglib\_env\_state;

ae\_int\_t n;

n = f.length();

alglib\_impl::ae\_state\_init(&\_alglib\_env\_state);

try

{

double result = alglib\_impl::polynomialcalceqdist(a, b, const\_cast<alglib\_impl::ae\_vector\*>(f.c\_ptr()), n, t, &\_alglib\_env\_state);

alglib\_impl::ae\_state\_clear(&\_alglib\_env\_state);

return \*(reinterpret\_cast<double\*>(&result));

}

catch(alglib\_impl::ae\_error\_type)

{

throw ap\_error(\_alglib\_env\_state.error\_msg);

}

catch(...)

{

throw;

}

}

string zapros3(const real\_1d\_array &x, const ae\_int\_t boundltype, const double boundl, const ae\_int\_t boundrtype, const double boundr, real\_1d\_array &d)

{

alglib\_impl::ae\_state \_alglib\_env\_state;

alglib\_impl::ae\_state\_init(&\_alglib\_env\_state);

try

{

alglib\_impl::spline1dgriddiffcubic(const\_cast<alglib\_impl::ae\_vector\*>(x.c\_ptr()), const\_cast<alglib\_impl::ae\_vector\*>(y.c\_ptr()), n, boundltype, boundl, boundrtype, boundr, const\_cast<alglib\_impl::ae\_vector\*>(d.c\_ptr()), &\_alglib\_env\_state);

alglib\_impl::ae\_state\_clear(&\_alglib\_env\_state);

return;

}

catch(alglib\_impl::ae\_error\_type)

{

throw ap\_error(\_alglib\_env\_state.error\_msg);

}

catch(...)

{

throw;

}

}

string zapros4(const spline1dinterpolant &c, const double x)

{

alglib\_impl::ae\_state \_alglib\_env\_state;

alglib\_impl::ae\_state\_init(&\_alglib\_env\_state);

try

{

double result = alglib\_impl::spline1dintegrate(const\_cast<alglib\_impl::spline1dinterpolant\*>(c.c\_ptr()), x, &\_alglib\_env\_state);

alglib\_impl::ae\_state\_clear(&\_alglib\_env\_state);

return \*(reinterpret\_cast<double\*>(&result));

}

catch(alglib\_impl::ae\_error\_type)

{

throw ap\_error(\_alglib\_env\_state.error\_msg);

}

catch(...)

{

throw;

}

}

string zapros5(vector<char> item, vector<char> universum) {

vector<bool> newArray;

for (int i = 0; i < universum.size(); i++) {

if (IsContain(universum[i], item)) {

newArray.push\_back(true);

}

else {

newArray.push\_back(false);

}

}

return newArray;

}

string zapros6(const double a, const double b, const real\_1d\_array &f, const double t)

{

alglib\_impl::ae\_state \_alglib\_env\_state;

ae\_int\_t n;

n = f.length();

alglib\_impl::ae\_state\_init(&\_alglib\_env\_state);

try

{

double result = alglib\_impl::polynomialcalceqdist(a, b, const\_cast<alglib\_impl::ae\_vector\*>(f.c\_ptr()), n, t, &\_alglib\_env\_state);

alglib\_impl::ae\_state\_clear(&\_alglib\_env\_state);

return \*(reinterpret\_cast<double\*>(&result));

}

catch(alglib\_impl::ae\_error\_type)

{

throw ap\_error(\_alglib\_env\_state.error\_msg);

}

catch(...)

{

throw;

}

}

string zapros7(const real\_1d\_array &x, const ae\_int\_t boundltype, const double boundl, const ae\_int\_t boundrtype, const double boundr, real\_1d\_array &d)

{

alglib\_impl::ae\_state \_alglib\_env\_state;

alglib\_impl::ae\_state\_init(&\_alglib\_env\_state);

try

{

alglib\_impl::spline1dgriddiffcubic(const\_cast<alglib\_impl::ae\_vector\*>(x.c\_ptr()), const\_cast<alglib\_impl::ae\_vector\*>(y.c\_ptr()), n, boundltype, boundl, boundrtype, boundr, const\_cast<alglib\_impl::ae\_vector\*>(d.c\_ptr()), &\_alglib\_env\_state);

alglib\_impl::ae\_state\_clear(&\_alglib\_env\_state);

return;

}

catch(alglib\_impl::ae\_error\_type)

{

throw ap\_error(\_alglib\_env\_state.error\_msg);

}

catch(...)

{

throw;

}

}

string zapros8(const spline1dinterpolant &c, const double x)

{

alglib\_impl::ae\_state \_alglib\_env\_state;

alglib\_impl::ae\_state\_init(&\_alglib\_env\_state);

try

{

double result = alglib\_impl::spline1dintegrate(const\_cast<alglib\_impl::spline1dinterpolant\*>(c.c\_ptr()), x, &\_alglib\_env\_state);

alglib\_impl::ae\_state\_clear(&\_alglib\_env\_state);

return \*(reinterpret\_cast<double\*>(&result));

}

catch(alglib\_impl::ae\_error\_type)

{

throw ap\_error(\_alglib\_env\_state.error\_msg);

}

catch(...)

{

throw;

}

}

string zapros9(vector<char> item, vector<char> universum) {

vector<bool> newArray;

for (int i = 0; i < universum.size(); i++) {

if (IsContain(universum[i], item)) {

newArray.push\_back(true);

}

else {

newArray.push\_back(false);

}

}

return newArray;

}

string zapros10(const double a, const double b, const real\_1d\_array &f, const double t)

{

alglib\_impl::ae\_state \_alglib\_env\_state;

ae\_int\_t n;

n = f.length();

alglib\_impl::ae\_state\_init(&\_alglib\_env\_state);

try

{

double result = alglib\_impl::polynomialcalceqdist(a, b, const\_cast<alglib\_impl::ae\_vector\*>(f.c\_ptr()), n, t, &\_alglib\_env\_state);

alglib\_impl::ae\_state\_clear(&\_alglib\_env\_state);

return \*(reinterpret\_cast<double\*>(&result));

}

catch(alglib\_impl::ae\_error\_type)

{

throw ap\_error(\_alglib\_env\_state.error\_msg);

}

catch(...)

{

throw;

}

}\*/

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Главное меню" << endl << endl;

cout << "1. Выдать список НОМЕРОВ АВТОМОБИЛЕЙ, остановленных данным СОТРУДНИКОМ." << endl;

cout << "2. Выдать список НОМЕРОВ АВТОМОБИЛЕЙ, нарушивших на данном УЧАСТКЕ." << endl;

cout << "3. Выдать список СОТРУДНИКОВ, работающих на данном УЧАСТКЕ в определенный день. В определенный день каждый сотрудник работает только на одном участке." << endl;

cout << "4. Выдать список ВОДИТЕЛЕЙ, остановленных данным СОТРУДНИКОМ на данном УЧАСТКЕ В определенный день." << endl;

cout << "5. Выдать список СОТРУДНИКОВ, закрепленных на данном УЧАСТКЕ, отправленных на повышение квалификации в данный ГОРОД. Сотрудник может быть закреплен только на одном участке." << endl;

cout << "6. Выдать список ВОДИТЕЛЕЙ, задержанных на данном УЧАСТКЕ в данном ГОРОДЕ в определенный момент. В определенный момент водитель может быть задержан на одном участке." << endl;

cout << "7. Выдать список УЧАСТКОВ, находящихся в данном ГОРОДЕ." << endl;

cout << "8. Выдать список УЧАСТКОВ, на которых останавливали данного ВОДИТЕЛЯ в данном ГОРОДЕ." << endl;

cout << "9. Выдать список СОТРУДНИКОВ, задерживавших автомобиль с данным АВТОМОБИЛЬНЫМ НОМЕРОМ в данном ГОРОДЕ." << endl;

cout << "10. Выдать список ГОРОДОВ, в которых данного ВОДИТЕЛЯ задерживали на данном УЧАСТКЕ." << endl;

cout << "0. Выход из программы" << endl << endl;

int command = -1;

cout << "Введите номер запроса" << endl;

while (command != 0)

{

cin >> command;

switch (command)

{

case 1:

{

cout << "Иванов" << endl << endl << "1. а023кс" << endl << "2. м263вт" << endl << "3. р888рм" << endl << "4. с912ан" << endl << endl;

cout << "Смирнов" << endl << endl << "1. в538нк" << endl << "2. р834ту" << endl << endl;

cout << "Березин" << endl << endl << "1. е936см" << endl << "2. т165вв" << endl << "3. м579то" << endl << endl;

break;

}

case 2:

{

cout << "Участок 1" << endl << endl << "1. р834ту" << endl << "2. к729мт" << endl << "3. е777ее" << endl << endl;

cout << "Участок 2" << endl << endl << "1. о693вт" << endl << "2. к491оа" << endl << endl;

break;

}

case 3:

{

cout << "Участок 1" << endl << endl << "1. Новиков" << endl << "2. Васильев" << endl << "3. Соколов" << endl << endl;

cout << "Участок 2" << endl << endl << "1. Петров" << endl << "2. Михайлов" << endl << "3. Кузнецов" << endl << endl;

break;

}

case 4:

{

cout << "Участок 1" << endl << endl << "Сотрудник Кожевников" << endl << endl << "Перевощиков" << endl << "Путин" << endl << endl;

cout << "Участок 2" << endl << endl << "Сотрудник Данилов" << endl << endl << "Кузьмин" << endl << "Еремин" << endl << "Афанасьев" << endl << endl;

break;

}

case 5:

{

cout << "Участок 1" << endl << endl << "Екатеринбург" << endl << endl << "Максимов" << endl << "Ведерников" << endl << "Орлов" << endl << endl;

cout << "Участок 2" << endl << endl << "Казань" << endl << endl << "Афанасьев" << endl << "Харламов" << endl << endl;

cout << "Участок 3" << endl << endl << "Москва" << endl << endl << "Белов" << endl << "Ершов" << endl << endl;

break;

}

case 6:

{

cout << "Санкт-Петербург" << endl << endl << "участок 1" << endl << endl << "Кутявин" << endl << "Ялышев" << endl << "Овчинников" << endl << endl;

cout << "Санкт-Петербург" << endl << endl << "участок 2" << endl << endl << "Ялышев" << endl << "Царев" << endl << "Хохряков" << endl << endl;

cout << "Рязань" << endl << endl << "участок 1" << endl << endl << "Самсонов" << endl << "Глебова" << endl << endl;

break;

}

case 7:

{

cout << "Москва" << endl << endl << "участок 1" << endl << endl << "Участок 2" << endl << "Участок 3" << endl << "Участок 4" << endl << "Участок 5" << endl << "Участок 6" << endl << "Участок 7" << endl << endl;

cout << "Сочи" << endl << endl << "участок 1" << endl << endl << "Участок 2" << endl << "Участок 3" << endl << "Участок 4" << endl << endl;

cout << "Пенза" << endl << endl << "участок 1" << endl << endl << "Участок 2" << endl << endl;

break;

}

case 8:

{

cout << "Пермь" << endl << endl << "Белоусов" << endl << endl << "Участок 1" << endl << "Участок 3" << endl << endl;

cout << "Пермь" << endl << endl << "Григорьев" << endl << endl << "Участок 3" << endl << "Участок 4" << endl << endl;

break;

}

case 9:

{

cout << "Воронеж" << endl << endl << "в538нк" << endl << endl << "Гогунский" << endl << "Пушин" << endl << "Фролов" << endl << endl;

cout << "Челябинск" << endl << endl << "к491оа" << endl << endl << "Морозов" << endl << "Калинин" << endl << endl;

cout << "Челябинск" << endl << endl << "с912ан" << endl << endl << "Калинин" << endl << "Гришин" << endl << endl;

break;

}

case 10:

{

cout << "Карпов" << endl << endl << "участок 2" << endl << endl << "Краснодар" << endl << "Новосибирск" << endl << "Владивосток" << endl << endl;

cout << "Логинов" << endl << endl << "участок 5" << endl << endl << "Самара" << endl << "Ярославль" << endl << "Калининград" << endl << endl;

cout << "Антонов" << endl << endl << "участок 3" << endl << endl << "Ижевск" << endl << "Уфа" << endl << endl;

break;

}

default:

break;

cout << "Введите команду" << endl;

}

}

system("pause");

return 0;

}

7. Вывод

В ходе лабораторной работы приобретены навыки анализа, структурирования и формализации информационного пространства, ограниченного предметной областью, а также навыки применения ER-моделей и методик их анализа для синтеза необходимых структур данных.