

МИНОБРНАУКИ РОССИИ

Орский гуманитарно-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования «Оренбургский государственный университет»
(Орский гуманитарно-технологический институт (филиал) ОГУ)

Кафедра программного обеспечения

Методические указания
для обучающихся по освоению дисциплины

«Б.1.В.ОД.16 Технология разработки программного обеспечения»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.01 Информатика и вычислительная техника
(код и наименование направления подготовки)

Программное обеспечение средств вычислительной техники и
автоматизированных систем

(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

Программа академического бакалавриата

Квалификация

Бакалавр

Форма обучения

Очная

Год начала реализации программы (набора)

2014, 2015, 2016, 2017

г. Орск 2017

Методические указания для обучающихся по освоению дисциплины «Б.1.В.ОД.16 Технология разработки программного обеспечения» предназначены для обучающихся очной формы обучения направления подготовки 09.03.01 Информатика и вычислительная техника, профиля «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Составитель  В.С. Богданова

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения, протокол № 2 от «07» июня 2017 г.

Заведующий кафедрой программного обеспечения

 Е.Е. Сурина

© Богданова В.С., 2017
© Орский гуманитарно-технологический институт (филиал) ОГУ, 2017

1 Методические указания по проведению лекционных занятий

Лекционные занятия в высшем учебном заведении являются основной формой организации учебного процесса и должны быть нацелены на выполнение ряда задач:

- ознакомить студентов со структурой дисциплины;
- изложить основной материал программы курса дисциплины;
- ознакомить с новейшими подходами и проблематикой в данной области;
- сформировать у студентов потребность к самостоятельной работе с учебной, нормативной и научной литературой.

Лекционное занятие представляет собой систематическое, последовательное, монологическое изложение преподавателем-лектором учебного материала, как правило, теоретического характера.

Цель лекции – организация целенаправленной познавательной деятельности студентов по овладению программным материалом учебной дисциплины.

Чтение курса лекций позволяет дать связанное, последовательное изложение материала в соответствии с новейшими данными науки, сообщить слушателям основное содержание предмета в целостном, систематизированном виде.

В ряде случаев лекция выполняет функцию основного источника информации, когда новые научные данные по той или иной теме не нашли отражения в учебниках.

Организационно-методической базой проведения лекционных занятий является рабочий учебный план направления подготовки. При подготовке лекционного материала преподаватель обязан руководствоваться учебными программами по дисциплинам кафедры, тематика и содержание лекционных занятий которых представлена в рабочих программах, учебно-методических комплексах.

При чтении лекций преподаватель имеет право самостоятельно выбирать формы и методы изложения материала, использовать различные технические средства обучения.

Рекомендации по работе студентов с конспектом лекций.

Изучение дисциплины студенту следует начинать с проработки рабочей программы, особое внимание, уделяя целям и задачам, структуре и содержанию курса.

При конспектировании лекций студентам необходимо излагать услышанный материал кратко, своими словами, обращая внимание, на логику изложения материала, аргументацию и приводимые примеры. Необходимо выделять важные места в своих записях. Если непонятны какие-либо моменты, необходимо записывать свои вопросы, постараться найти ответ на них самостоятельно. Если самостоятельно не удалось разобраться в материале, впоследствии необходимо либо на следующей лекции, либо на лабораторном занятии или консультации обратиться к ведущему преподавателю за разъяснениями.

Успешное освоение курса предполагает активное, творческое участие студента путем планомерной, повседневной работы. Лекционный материал следует просматривать в тот же день. Рекомендуемую дополнительную литературу следует прорабатывать после изучения данной темы по учебнику и материалам лекции.

Каждая тема имеет свои специфические термины и определения. Усвоение материала необходимо начинать с усвоения этих понятий. Если какое-либо понятие вызывает затруднения, необходимо посмотреть его суть и содержание в словаре (Интернете), выписать его значение в тетрадь для подготовки к занятиям.

При подготовке материала необходимо обращать внимание на точность определений, последовательность изучения материала, аргументацию, собственные примеры, анализ конкретных ситуаций. Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам и тестам.

2 Методические указания по лабораторным работам

Изучение дисциплины «Технология разработки программного обеспечения» предполагает посещение обучающимися не только лекций, но и лабораторных работ. Лабораторные работы со студентами предназначены для проверки усвоения ими теоретического материала дисциплины.

Основные цели лабораторных работ:

- закрепить основные положения дисциплины;
- проверить уровень усвоения и понимания студентами вопросов, рассмотренных на лекциях и самостоятельно изученных по учебной литературе;
- научить пользоваться нормативной и справочной литературой для получения необходимой информации о конкретных технологиях;
- оказать помощь в приобретении навыков расчета точностных характеристик;
- восполнить пробелы в пройденной теоретической части курса и оказать помощь в его усвоении.

Для контроля знаний, полученных в процессе освоения дисциплины на лабораторных занятиях обучающиеся выполняют задания реконструктивного уровня и комплексное практическое задание.

Целью выполнения задания реконструктивного уровня и комплексного задания студентами является систематизация, закрепление и расширение теоретических знаний, полученных в ходе изучения дисциплины.

Ниже приводятся общие методические указания, которые относятся к занятиям по всем темам:

- в начале каждого лабораторного занятия необходимо сформулировать цель, поставить задачи;
- далее необходимо проверить знания студентами лекционного материала по теме занятий;
- в процессе занятия необходимо добиваться индивидуальной самостоятельной работы студентов;
- знания студентов периодически контролируются путем проведения текущей аттестации (рубежного контроля), сведения о результатах которой доводятся до студентов и подаются в деканат;
- время, выделенное на отдельные этапы занятий, указанное в рабочей программе, является ориентировочным; преподаватель может перераспределить его, но должна быть обеспечена проработка в полном объеме приведенного в рабочей программе материала;
- на первом занятии преподаватель должен ознакомить студентов с правилами поведения в лаборатории и провести инструктаж по охране труда и по пожарной безопасности на рабочем месте;
- преподаватель должен ознакомить студентов со всем объемом лабораторных работ и требованиями, изложенными выше;
- преподаватель уделяет внимание оценке активности работы студентов на занятиях, определению уровня их знаний на каждом занятии.

На лабораторных работах решаются задачи из всех разделов изучаемой дисциплины.

Лабораторная работа №1

Создание таблиц баз данных

Создайте в BDE Administrator новый псевдоним со своей фамилией. Укажите драйвер БД по умолчанию Paradox, а в параметре Path укажите путь к созданной вами папке

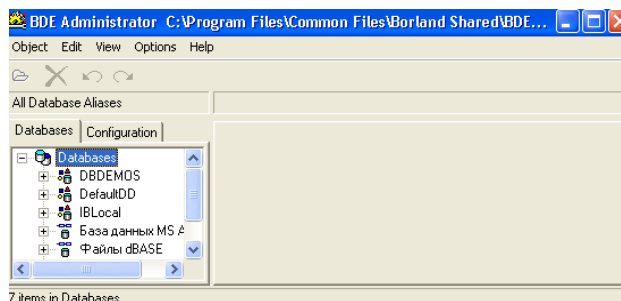


Рисунок 1

Для работы с таблицами БД при проектировании приложения удобно использовать Database Desktop, которая позволяет:

- создавать таблицы
- изменять структуры
- редактировать записи.

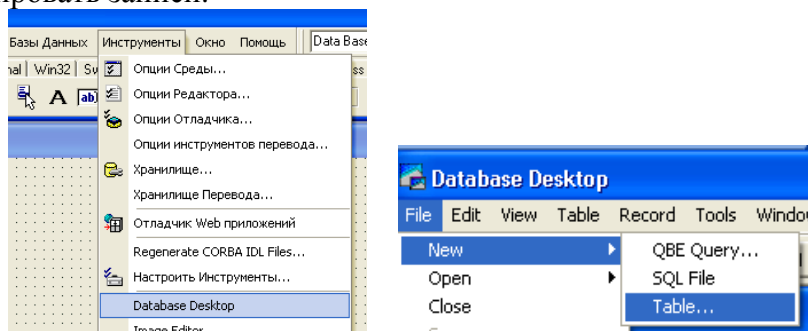


Рисунок 2

Кроме того, с помощью Database Desktop можно выполнять и другие действия над БД (создание, редактирование и выполнение визуальных и SQL – запросов, операций с псевдонимами).

Процесс создания новой таблицы начинается с вызова команды File/New/Table (Файл/Новая/Таблица) и происходит в интерактивном режиме. При этом разработчик должен:

- выбрать формат (тип) таблицы
- задать структуру таблицы.

В начале создания новой таблицы в окне Create Table (создание таблицы) выбирается ее формат. По умолчанию предлагается формат таблицы Paradox, который и будем использовать. Для таблиц других форматов, например dBase IV, действия по созданию таблицы практически не отличаются.

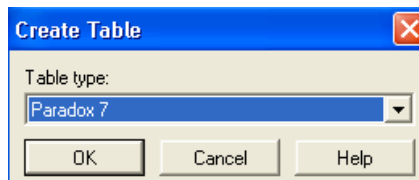


Рисунок 3

После выбора формата таблицы появляется окно определения структуры таблицы, в котором выполняются следующие действия:

- описание полей
- задание ключа
- задание индексов

- определение ограничений на значения полей
- определение условий ссылочной целостности
- задание паролей
- задание языкового драйвера
- задание таблицы для выбора значений

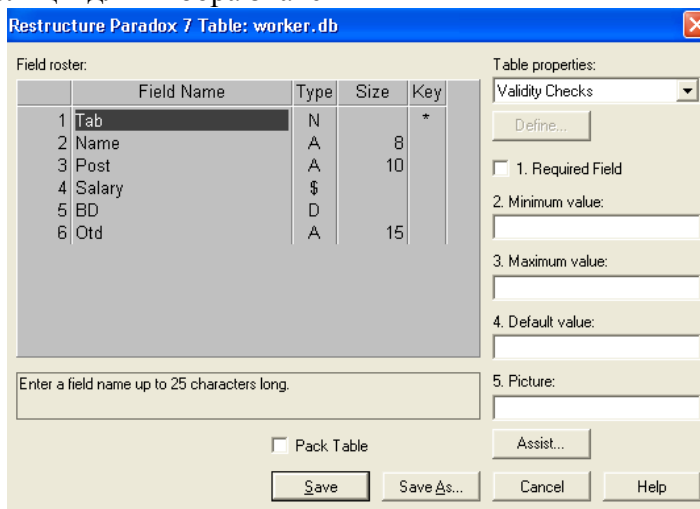


Рисунок 4

В этом списке обязательным является первое действие, т.е. каждая таблица должна иметь хотя бы одно поле. Остальные действия выполняются при необходимости. Часть действий, такие как задание ключа и паролей, производится только для таблиц определенных форматов, например, для таблиц Paradox.

При создании новой таблицы, сразу после выбора ее формата можно не создавать структуру таблицы, а скопировать ее из другой таблицы: при нажатии кнопки Borrow (Взаймы) открывается окно Select Borrow Table (выбор таблицы для заимствования).

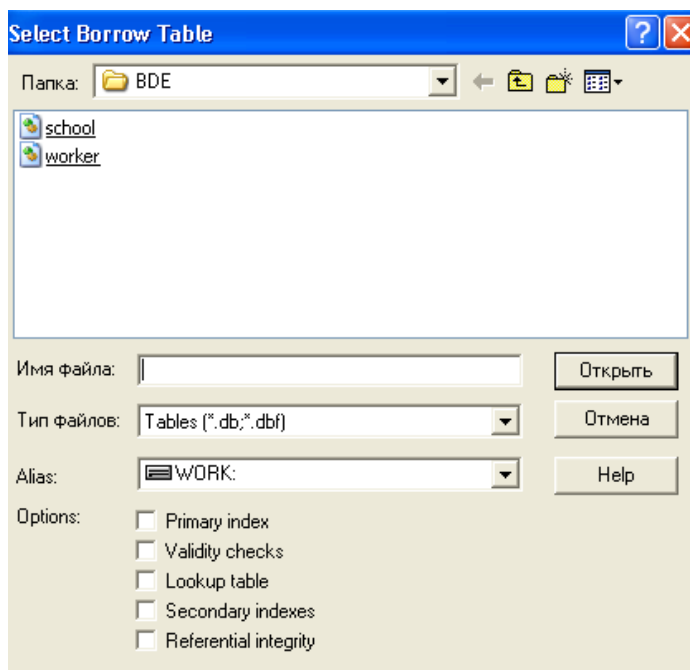


Рисунок 5

В этом окне можно выбрать таблицу и указать копируемые элементы структуры, установив соответствующий флажок, например, Primary index (первичный индекс) для ключа. После нажатия кнопки Open из выбранной таблицы в новую копируются описания полей, а также те элементы, для которых установлен флажок. Если какой – либо элемент в структуре копируемой таблицы отсутствует, то состояние флажка не имеет значения. Например, если в выбранной таблице не определены ограничения ссылочной целостности,

то в новой таблице они не появятся, даже если установлен флажок Referential integrity (ссылочная целостность).

Впоследствии скопированную структуру можно настраивать, изменяя, добавляя или удаляя ее отдельные элементы.

После определения структуры таблицы ее необходимо сохранить, нажав кнопку Save as и указав расположение таблицы на диске и ее имя. В результате на диск записывается новая таблица, первоначально пустая, при этом все необходимые файлы создаются автоматически.

Описание полей

Центральной частью окна определения структуры таблицы является список Field roster (список полей), в котором указываются поля таблицы. Для каждого поля задаются:

- имя поля – в столбце Field Name
- тип поля – в столбце Type
- размер поля - в столбце Size

Имя поля вводится по правилам, установленным для выбранного формата таблиц.

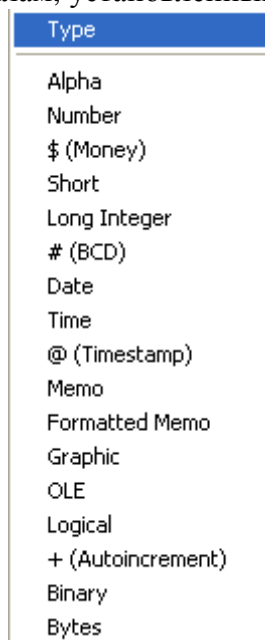


Рисунок 6

Тип поля можно задать непосредственно указав соответствующий символ, выбрав его в списке. Список содержит все типы полей, допустимые для заданного формата таблицы.

Размер поля задается не всегда, необходимость его указания зависит от типа поля. Для полей определенного типа, например, автоинкрементного (+) или целочисленного, размер поля не задается. Для поля строкового типа размер определяет максимальное число символов, которые могут храниться в поле. Добавление к списку полей новой строки выполняется переводом курсора вниз на несуществующую строку, в результате чего эта строка появляется в конце списка.

Для указания ключевых полей в столбце ключа (Key) нужно установить символ *, переведя в эту позицию курсор и нажав любую клавишу.

Для выполнения остальных действий по определению структуры таблицы используется комбинированный список Table Properties (свойства таблицы).

Создание приложения BDE

Для примера рассмотрим создание приложения, использующего механизм доступа BDE и позволяющего перемещаться по записям таблицы БД, просматривать и редактировать поля, удалять записи из таблицы, а также вставлять новые. Файл проекта приложения обычно не требует от разработчика выполнения каких – либо действий. Поэтому при создании приложения главной задачей является конструирование форм, в простейшем случае – одной формы.

Вид формы приложения на этапе проектирования показан на рисунке, где в форме размещены компоненты Table1, DataSource1,DBGrid, DBNavigator1.

Компонент Table1 обеспечивает взаимодействие с таблицей БД. Для связи с требуемой таблицей нужно установить в соответствующее значения свойство DataBaseName, указывающее путь к БД, и свойство TableName, указывающее имя таблицы. После задания таблицы для открытия набора данных свойство Active должно быть установлено в значение True.

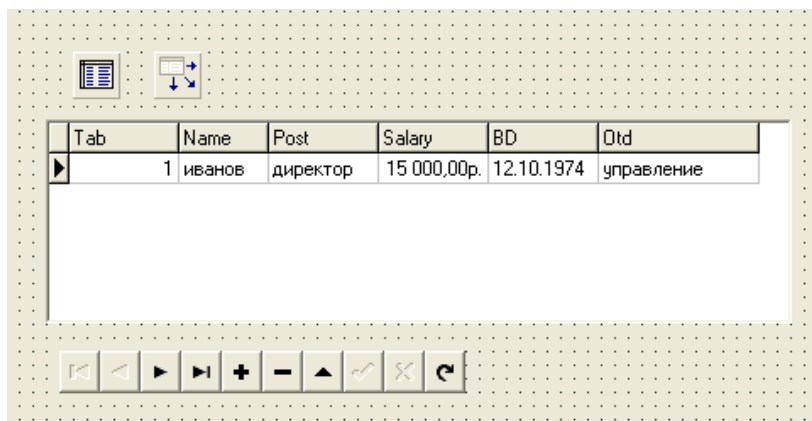


Рисунок 7

Компонент DBGrid отображает содержимое таблицы БД в виде сетки, в которой столбцы соответствуют полям, а строки – записям таблицы. По умолчанию пользователь может просматривать и редактировать данные. Компонент DBNavigator позволяет пользователю перемещаться по таблице, редактировать, вставлять и удалять записи. Компоненты DBGrid и DBNavigator связываются со своим источником данных – компонентов DataSource1 – через свойства DataSource.

Лабораторная работа №2 ***Базы данных в Delphi***

1)С помощью утилиты DataBase Desktop создайте таблицу для хранения информации об учениках школы. Пусть файл School.db содержит поля:

N – личный номер ученика (числовое поле с автоувеличением)

Name – имя (строковый тип)

Fam – фамилия (строковый тип)

Class – класс (строковый тип)

Adr – адрес (строковый тип)

2) В программе используется псевдоним Школа. **Вам следует использовать свой псевдоним** при помощи BDE Administrator.

3) Создайте форму приложения в Delphi. На форму поместите компоненты Table1 (вкладка BDE) и DataSource1(вкладка Data Access), которые обеспечивают доступ к файлу данных, и установите следующие значения их свойств:

Свойство	Значение
Table1.DatadaseName	Школа
Table1.TableName	School.db
Table1.Active	True

DataSource1.Dataset	Table1
---------------------	--------

4) Для просмотра и редактирования данных в режиме таблицы в форму надо добавить компонент DBGrid1 (вкладка Data Controls).

5) Чтобы задать, какая информация будет отображаться в таблице DBGrid1, необходимо задать источник данных для таблицы (значение свойства DataSource = DataSource1).

Свойство Options.dgTitles разрешает вывод строки заголовка столбцов, Options.dgIndicator разрешает вывод колонки индикатора во время работы с базой данных (треугольник – текущая запись, звездочка – новая запись).

6) Значения уточняющих параметров свойства Columns задаются в окне Editing DBGrid1.Columns. Добавьте в компонент DBGrid1 4 колонки с помощью команды Добавить новый(Ins). Свойство Columns представляет собой массив компонентов типа TColumn. Каждой колонке соответствует элемент массива.

Свойство	Определяет
FieldName	Поле записи, содержимое которого выводится в колонке
Width	Ширину колонки в пикселах
Font	Шрифт, используемый для вывода текста в ячейках колонки
Color	Цвет фона колонки
Alignment	Способ выравнивания текста в ячейках колонки. Текст может быть выровнен по левому краю (taLeftJustify), по центру (taCenter) или по правому краю (taRightJustify)
Title.Caption	Заголовок колонки. Значением по умолчанию является имя поля записи
Title.Alignment	Способ выравнивания заголовка колонки. Заголовок может быть выровнен по левому краю (taLeftJustify), по центру (taCenter) ИЛИ ПО правому краю (taRightJustify)
Title.Color	Цвет фона заголовка колонки
Title.Font	Шрифт заголовка колонки

В простейшем случае для каждой колонки достаточно установить значение свойства FieldName, которое определяет имя поля записи, содержимое которого будет отображаться в колонке, а также значение свойства Title.caption, определяющего заголовок колонки.

7) Добавить к форме компонент DBNavigator, настроив его на работу с таблицей-источником данных (свойству DataSource нужно присвоить значение DataSource 1).

8) Откомпилируйте и запустите программу. Следует обратить внимание, что для того чтобы после запуска программы в окне появилась информация или, если база данных пустая, можно было вводить новую информацию, свойство Active таблицы-источника данных должно иметь значение True.

Используя клавиши перемещения курсора вверх и вниз, а также клавиши листания текста страницами (<Page Up> и <Page Down>), можно, перемещаясь от строки к строке, просматривать записи базы данных. Нажав клавишу <Ins>, можно добавить запись, а нажав клавишу — удалить запись. Для того чтобы внести изменения в поле записи, нужно, используя клавиши перемещения курсора влево и вправо, выбрать необходимое поле и нажать клавишу <F2>.

Выбор информации из базы данных

При работе с базой данных пользователя, как правило, интересует не все ее содержимое, а некоторая конкретная информация. Найти нужные сведения можно

последовательным просмотром записей. Однако такой способ поиска неудобен и малоэффективен.

Большинство систем управления базами данных позволяют произвести выборку нужной информации путем выполнения *запросов*. Пользователь в соответствии с определенными правилами формулирует запрос, указывая, каким критериям должна удовлетворять интересующая его информация, а система выводит записи, удовлетворяющие запросу.

Для выборки из базы данных записей, удовлетворяющих некоторому критерию, предназначен компонент Query (вкладка BDE).

Компонент Query похож на компонент Table, но, в отличие от последнего, он представляет не всю базу данных (все записи), а только ее часть — записи, удовлетворяющие критерию запроса. В таблице перечислены некоторые свойства компонента Query.

Свойство	Определяет
Name	Имя компонента. Используется компонентом Datasource для связи результата выполнения запроса (набора записей) с компонентом, обеспечивающим просмотр записей, например DBGrid
SQL	Записанный на языке SQL запрос к базе данных (к таблице)
Active	При присвоении свойству значения True активизирует выполнение запроса

Для того чтобы во время разработки программы задать, какая информация будет выделена из базы данных в результате выполнения запроса, свойство SQL должно содержать представленный на языке SQL запрос на выборку данных.

Например, запрос
SELECT Fam, Name
FROM 'Школа:school.db'
WHERE (Class = '11a')
ORDER BY Name, Fam

обеспечивает выборку из базы данных "Школа" (из таблицы School.db) списка учеников 11-а класса.

Следующий запрос:
SELECT Fam, Name FROM "Школа:school.db"
WHERE (Fam > 'К') and (Fam < 'Л')
ORDER BY Name, Fam

обеспечивает выбор информации об учениках, фамилии которых начинаются на букву К.

Запрос может быть сформирован и записан в свойство SQL во время разработки формы или во время работы программы.

Для записи запроса в свойство SQL во время разработки формы используется редактор списка строк, окно которого открывается в результате щелчка на кнопке с тремя точками в строке свойства SQL окна

Object Inspector.

Свойство sql представляет собой список строк. Поэтому чтобы сформировать запрос во время работы программы, нужно, используя метод Add, добавить строки (Sql-инструкции) в список sql.

Ниже приведен фрагмент кода, который формирует запрос на поиск информации о конкретном человеке (критерий выбора — содержимое поля Fam должно совпадать со значением переменной fam).

```
with form1.Query1 do
```

```
begin
```

```
Close; // закрыть файл — результат выполнения предыдущего запроса
```

```
SQL.Clear; // удалить текст предыдущего запроса
```

```

// записываем новый запрос в свойство SQL
SQL.Add('SELECT Fam, Name, Class');
SQL.Add('FROM ":Школа:school.db" WHERE');
SQL.Add('(Fam = " + fam + ")');
SQL.Add('ORDER BY Fam');
Open; // активизируем выполнение запроса
end;

```

Следующая программа (диалоговое окно приведено на рисунке), демонстрирует возможность изменения запроса, точнее, критерия запроса, во время работы программы. Программа обеспечивает вывод как всего списка учеников, так и его части. Например, посредством выполнения запроса выводится информация только о конкретном ученике.

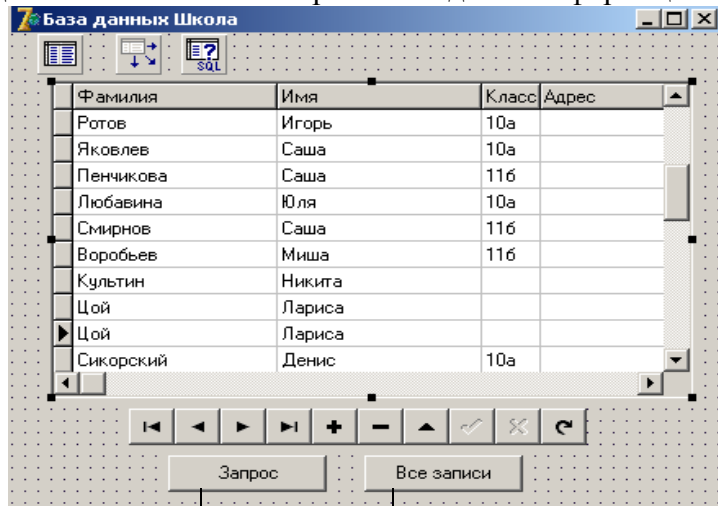


Рис. Форма приложения База данных Школа

Button1

Button2

```

procedure TForm1.Button1Click(Sender: TObject);
var
  fam: string[30];
begin
  fam:=InputBox('Выборка информации', 'Укажите фамилию и нажмите ОК.', '');
  if fam <> '' then
    begin
      with form1.Query1 do begin
        Close;
        SQL.Clear;
        SQL.Add('SELECT Fam, Name, Class');
        SQL.Add('FROM ":Школа:school.db" WHERE Fam = "'+ fam + '"');
        Open;
      end;
      if Query1.RecordCount <> 0 then
        DataSource1.DataSet:=Query1
      else begin
        ShowMessage('В базе нет данных, удовлетворяющих критерию запроса');
        DataSource1.DataSet:=Table1;
      end;
    end;
end; // конец proc

procedure TForm1.Button2Click(Sender: TObject);
begin
  DataSource1.DataSet:=Table1; // задается источник данных - таблица
end;
procedure TForm1.FormActivate(Sender: TObject);
begin
  DataSource1.DataSet := Table1;
  Table1.Active := True;
end;

```

Индивидуальные лабораторные работы

1 Вариант

1. Создать БД по учету кварт платы из двух таблиц:
Жильцы(Код жильца, ФИО, Улица, дом, квартира, площадь, кол-во проживающих)
Платеж (Код платежа, Код жильца, дата, сумма платежа)
2. Определить ключи, связав таблицы
3. Составить запросы:
 - все сведения о квартиросъемщиках с фамилией, начинающейся с «А»
 - данные по текущему месяцу, с подведением итогов
 - расчет квартплаты по каждому жильцу при следующих условиях – 2р. за кв. метр+ 0,5р. за каждого проживающего

2 Вариант

1. Создать БД по учету успеваемости студентов
Предметы(Код_предмета, название предмета, группа, ФИО преподавателя)
Студенты (Код_студента, ФИО, факультет, группа)
Экзамены(Код, Код предмета, Код_студента, оценка, дата сдачи)
2. Определить ключи, связав таблицы по заданным полям
3. Составить запросы:
 - все сведения о студентах первого курса технологического факультета
 - средний балл каждого студента заданной с клавиатуры группы
 - сведения о студентах имеющих 2 по предметам
 - средний балл по каждому курсу каждого факультета

3 Вариант

1. Создать БД по анализу отгрузки оборудования по заводу
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - загрузка оборудования каждого цеха по кварталам, ч/з группировку
 - данные по текущему месяцу, с подведением итогов
 - сведения о заданном оборудовании
 - о начислении штрафа цеху в размере 15%, если не выполнен план.

4 Вариант

1. Создать БД по учету поставки продукции по заводу Продукция (Код, Наименование, Кол-во, цена)
Заказчики (Код, ФИО, Адрес), Заказы (Код,
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - план поставки продукции по заказчикам в разрезе кварталов
 - информация о заданном заказчике
 - итоговая ведомость оплаты заказчикам
 - о расчете долга по уплате поставки

5 Вариант

1. Создать БД с информацией о кол-ве рабочих по цехам, профессиям, разрядам, для анализа средней зарплаты
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - средняя з/плата по заводу в разрезе профессий и разрядов

- данные по определенной профессии с выводом информации о з/п
- сведения о заданном цехе
- о начислении премий работникам, перевыполнившим план, ч/з функцию.

6 Вариант

1. Создать БД по учету товара на складе и его отправке клиентам
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - о клиентах сделавших заказ на МАХ сумму
 - данные по текущему месяцу, с подведением итогов
 - сведения о недостающих на складе товарах
 - о начислении стоимости за товары каждого вида и заказа в целом

7 Вариант

1. Создать БД по анализу выполнения плана рабочими на п/п
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - численность рабочих в определенном цехе
 - данные по выполнению плана рабочими всего п/п
 - сведения о нормативном плане на 1 рабочего (расчитать)
 - о снятии штрафа в размере 1 мин. з/платы если не выполнен план, ч/з функцию

8 Вариант

1. Создать БД по учету выдачи в прокат видео кассет
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - о клиенте взявшем более 3 кассет
 - данные по клиентам, которым начисляется штраф, в связи с невозвращением кассет, ч/з функцию
 - сведения о всех клиентах с учетом стоимости
 - о кассетах с фильмами-боевиками

9 Вариант

1. Создать БД по учету выдачи в прокат книг в библиотеке
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - о клиенте взявшем более 3 книг
 - данные по клиентам, которым начисляется штраф, в связи с невозвращением книг, ч/з функцию
 - сведения о всех клиентах с учетом стоимости
 - о книгах-детективах

3 Методические указания по самостоятельной работе

Для успешного освоения курса «Проектирование информационных систем» необходима самостоятельная работа. В настоящее время актуальными становятся требования к личным качествам современного студента – умению самостоятельно пополнять и обновлять знания, вести самостоятельный поиск необходимого материала, быть творческой личностью.

Самостоятельную работу по освоению дисциплины обучающимся следует

начинать с изучения содержания рабочей учебной программы дисциплины, цели и задач, структуры и содержания курса, основной и дополнительной литературы, рекомендованной для самостоятельной работы.

Самостоятельная учебная деятельность является необходимым условием успешного обучения. Многие профессиональные навыки, способность мыслить и обобщать, делать выводы и строить суждения, выступать и слушать других, – все это развивается в процессе самостоятельной работы студентов.

Самостоятельная работа по освоению дисциплины включает:

- самостоятельное изучение разделов;
- самоподготовку (проработку и повторение лекционного материала и материала учебников и учебных пособий);
- подготовку к лабораторным работам;
- подготовку к рубежному и итоговому контролю.

Самостоятельная учебная работа – условие успешного окончания высшего учебного заведения. Она является равноправной формой учебных занятий, наряду с лекциями, семинарами, экзаменами и зачетами, но реализуемая во внеаудиторное время.

Эффективность аудиторных занятий во многом зависит от того, насколько умело студенты организуют в ходе них свою самостоятельную учебную познавательную деятельность. Такая работа также способствует самообразованию и самовоспитанию, осуществляемому в интересах повышения профессиональных компетенций, общей эрудиции и формировании личностных качеств.

Самостоятельная работа реализуется:

1. непосредственно в процессе аудиторных занятий – на лекциях, лабораторных занятиях, при проведении рубежного контроля;
2. в контакте с преподавателем вне рамок расписания – на консультациях по учебным вопросам, при ликвидации задолженностей, при выполнении индивидуальных заданий;
3. в библиотеке, дома, в общежитии, на кафедре при выполнении студентом учебных задач.

В процессе проведения самостоятельной работы необходимо производить подбор литературных источников, научной периодической печати и т.д

Лабораторная работа №1

Теоретическое введение в предметную область

Настоящий курс лабораторных работ ориентирован на изучение CASE - средств на примере диаграмм, создаваемых для проекта «Служба занятости в рамках вуза». Процесс создания диаграмм начинается с этапа изучения предметной области, которая описывается в этой лабораторной работе.

1. Описание системы

Система предназначена для того, чтобы помочь студенту устроиться на работу уже в процессе его обучения в вузе. Подав заявление в систему, студент становится ее клиентом и начинает обслуживаться на протяжении всего обучения в вузе. Заявление представляет собой анкету. Система предлагает профессиональные (основанные на изучаемых предметах), психологические тестирования, проводимые регулярно (раз в семестр (полгода)).

Особое внимание уделяется обучению студента, по итогам успеваемости составляются экспертные оценки. На основе собранной информации составляется резюме, представляющее собой полную характеристику человека. Это резюме отсылается всем организациям, имеющим необходимые вакансии.

Основным назначением системы является автоматизация ввода и хранения отчетных данных по студентам, составления характеристик и резюме, поиска вакансий в фирмах. Система позволяет изменять, дополнять, вести поиск и просмотр информации о студентах, накладывать ограничения доступа к системе, хранить списки студентов, закончивших обучение, в виде архива, контролировать выдачу студенту заданий на курсовые работы и проекты, связывать институт с фирмами, заинтересованными в поиске сотрудников.

Также данная система может быть использована для составления отдельных списков групп, для печати зачетных ведомостей, для печати полной базы данных и для статистики.

Система состоит из четырех подсистем:

- контроля успеваемости студентов;
- профессиональных и психологических тестов;
- обработки запросов, определения категорий полномочий пользователей;
- экспертных оценок.

1.1. Подсистема контроля успеваемости студентов

Эта подсистема является частью системы «Служба занятости в рамках вуза», которая отвечает за статистическую отчетность по успеваемости отдельного студента, группы или целого факультета, а также за хранение и правильность ее ввода.

Входными данными подсистемы являются: оценки, даты сдачи экзаменов, имена студентов, номера групп, факультет. На выходе подсистема выдает обработанные данные: средний балл по студенту, группе или факультету, процентное соотношение оценок у студента в группе или на факультете, имена и количество стипендиатов в группе или на факультете.

Подсистема «**Контроль успеваемости студентов**» может функционировать отдельно от всей системы, что дает возможность установить и использовать ее независимо, если это необходимо.

Подсистема «Контроль успеваемости студентов» включает следующие функции:

- ввод, вывод и редактирование информации по информационным объектам подсистемы;
- сохранение информации, поступившей от подсистемы контроля успеваемости студентов;
- расчет процентного соотношения оценок у студента в группе или на факультете и вывод его в виде таблиц, графиков и диаграмм;
- расчет среднего балла по студенту, группе или факультету;
- формирование данных по студенту, группе или факультету;
- выявление сильнейших и слабейших студентов в группе или на факультете;
- расчет количества стипендиатов в группе или на факультете;
- проверку правильности ввода данных.

1.2. Подсистема профессиональных и психологических тестов

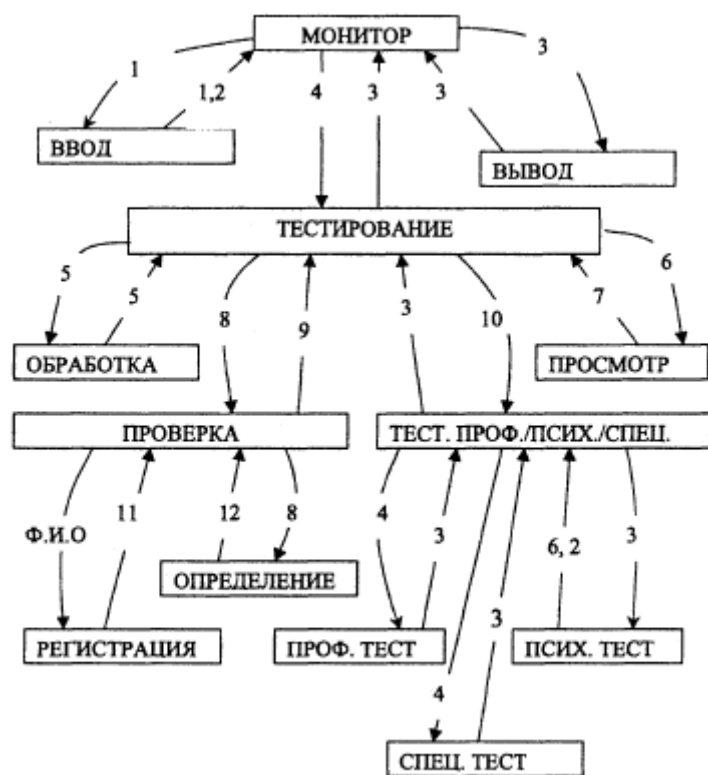


Рисунок 1. Подсистема профессиональных и психологических тестов

1. Ф.И.О., новые тесты, название теста.
2. Ответы на тесты.
3. Вопросы, результаты теста, Ф.И.О.
4. Ответы, Ф.И.О., название теста.
5. Новый тест.
6. Ф.И.О., название теста.
7. Результаты теста, Ф.И.О.
8. Ф.И.О., специальность.

9. Ф.И.О., набор тестов.

10. Ф.И.О., профессионально-психологический тест, психологический тест, специальный тест, ответы.

11. Ф.И.О., информация о пройденных тестах.

12. Ф.И.О., набор тестов для клиента.

Внешние сущности

Фирма	Тип производственного объединения предприятия. В данной ситуации фирма выступает как работодатель. Она отправляет запрос на специалиста.
Клиент	Человек, поступивший учиться в институт и пользующийся услугами СЛУЖБЫ ЗАНЯТОСТИ для нахождения работы. Клиент получает запрос на проведение профессионального, психологического и специального тестов и вопросы тестов. Затем КЛИЕНТ отправляет входные данные, ответы на вопросы тестов, запрос на результаты тестов.
Архив	Хранилище, где хранятся все данные о клиенте, начиная с момента пользования услугами СЛУЖБЫ ЗАНЯТОСТИ.
Дополнительные источники	Источники, из которых поступают новые, более современные тесты. Это INTERNET, журналы, какие-либо специализированные центры.

Работа модулей подсистемы

Монитор - вызывает модуль ввода и получает от него данные для выполнения дальнейших задач; обращается к модулю тестирования, передавая ему исходные данные; после выполнения своей задачи тестирование передает ему свои результаты; вызывает модуль вывода и получает от него конечный результат.

Ввод - получает запрос от монитора, передает ему запрашиваемые данные.

Вывод - вызывается монитором и получает от него результаты, выводя их на дисплей.

Тестирование - вызывается монитором, получает от него исходные данные и в зависимости от них обращается к тому или иному модулю. Выполнив свои функции, модули посылают тестированию результаты действий, а тестирование уже передает полученные результаты монитору.

Проверка - получает исходные данные от тестирования и передает их своим подмодулям: регистрация тестируемых, определение тестов. Подмодули передают полученные результаты модулю проверки. Тестирование (профессиональное, психологическое, специальное) получает исходные данные от модуля тестирования и передает их своим подмодулям: профессиональное, психологическое, специальное тестирование. Подмодули передают полученные результаты.

Обработка - получает данные от тестирования, выполняет свои задачи и возвращает полученные результаты модулю тестирования.

Просмотр - получает данные от модуля тестирования, выполняет свои задачи и посылает полученные результаты модулю тестирования.

1.3. Подсистема обработки запросов, определения категорий пользователей

Данная подсистема предназначена для определения категории, полномочий и обработки запросов пользователей службы занятости. В частности, она выполняет следующие функции:

- регистрацию новых фирм;
- регистрацию новых студентов;
- определение прав доступа зарегистрированного пользователя;
- обработку запросов;
- прием регистрационных данных от фирм;
- прием регистрационных данных от студентов;
- прием регистрационных данных от обслуживающего персонала;
- составление резюме;
- запись данных в БД студентов;
- запись данных в БД фирм;
- запись данных в БД зарегистрированных пользователей.

В соответствии с выполняемыми функциями система работает со следующими данными:

- регистрационными данными студентов;
- регистрационными данными фирм;
- личными данными студентов;
- информацией о студентах (получаемой фирмами);
- информацией о фирмах (получаемой студентами);
- идентификационными данными пользователей;
- информацией о системе;
- запросом;
- служебной информацией (для обслуживающего персонала);
- результатом психологического теста;
- результатом профессионального теста;
- экспертными оценками.



Рисунок 2. Подсистема обработки запросов, определения категорий пользователей

Определение категории - модуль, определяющий категорию пользователя.

Определение полномочий - модуль, определяющий полномочия пользователя.

Обработка запроса - модуль, предназначенный для обработки запросов пользователя.

Выполнение запроса - модуль, предназначенный для выполнения запросов пользователя.

Запись в БД зарегистрированных пользователей - модуль, предназначенный для работы с базой данных зарегистрированных пользователей.

Запись в БД студентов - модуль, предназначенный для работы с БД студентов.

Запись в БД фирм - модуль, предназначенный для работы с БД фирм.

1.4. Подсистема экспертных оценок

Эта подсистема предназначена для установки и просмотра экспертной оценки. Она дает краткую информацию преподавателю о студенте или группе. Студент может с помощью ее ориентироваться в учебе.

Кроме того, подсистема может дать представителю фирмы некоторое представление о студенте для рассмотрения его в качестве новых кадров.

Подсистема является как информационной средой, так и средой установки экспертной оценки.

Определение статуса клиента

- Клиент входит в подсистему с идентификационным номером. По номеру присваивается уровень доступа. Подсистема выдает меню работы, соответствующее уровню доступа.

Обработка отчетного задания

- Эксперт: посылает запрос на предоставление информации о студенте (группе), результатах тестирования. Из общей БД поступает успеваемость студента (группы), результаты тестирования. Делает

запрос на критерии. Из хранилища критериев поступают критерии для установки экспертной оценки. Вводит экспертную оценку. Делает запрос на просмотр экспертной оценки. Из общей БД поступает экспертная оценка.



Рисунок 3. Подсистема экспертных оценок

- Студент: посылает запрос на предоставление информации об успеваемости, результатах тестирования. Из общей БД поступает информация об успеваемости студента, результаты тестирования. Делает запрос на просмотр экспертной оценки, которая поступает из общей БД.

- Представитель фирмы: делает запрос на просмотр экспертной оценки, которая поступает из общей БД.

Свертка: модуль, получающий из временной БД экспертные оценки, проводит их обработку и передает обработанные данные в общую БД.

Общая база данных содержит в себе:

- экспертные оценки (предоставляются подсистемой экспертных оценок),
- успеваемость студента (предоставляется подсистемой контроля студенческой успеваемости),
- успеваемость группы (предоставляется подсистемой контроля успеваемости)
- результаты профессионального теста (предоставляются подсистемой профессиональных и психологических тестов),
- результаты психологического теста (предоставляются подсистемой профессиональных и психологических тестов).

Временная база данных содержит в себе экспертные оценки (поставленные экспертом).

Критерии экспертных оценок

- По 5-балльной системе оценить студента в начале, середине, конце семестра.
- По 5-балльной системе оценить степень запоминания курса студентом в начале, середине, конце семестра.
- По 5-балльной системе оценить степень применения знаний студентом в начале, середине, конце семестра.

2. Модульная структура системы

1. Запрос, имя, уровень доступа.
2. Диаграммы, графики, списки, ведомости, количество стипендиатов, отчеты об успеваемости.
3. Имя, личные данные студента, запросы на экспертные оценки.
4. Данные студента.
5. Данные студента, данные фирмы, запросы, оценки, пароль, имя, уровень доступа.
6. Отчеты, резюме, графики, вопросы тестов.
7. Уровень доступа.
8. Имя.
9. Пароль.
10. Полномочия.
11. Резюме.

12. Результаты тестов.
13. Запрос.
14. Найденные записи.
15. Экспертные оценки.



Рисунок 4. Модульная структура системы

3. Информационные объекты системы

Под информационным объектом хранения (информационным элементом) понимается логически однородная единица информации, для хранения которой достаточно одной записи таблицы.

Информационные объекты хранения для БД системы:

А. Общего назначения:

1. Факультет.
2. Учебная специальность.
3. Группа студентов.

4. Студент.
5. Преподаватель.
6. Учебный предмет (дисциплина).
7. Тип отчетного задания (курсовой проект, реферат, контрольная работа, зачет, экзамен и т. д.).
8. Отчетное задание по конкретному изучаемому предмету – описание сути отчетного задания и необходимых требований для его успешного выполнения.
9. Отчетное задание, сданное конкретным студентом, - экземпляр отчетного задания.
10. Экспертная оценка успеваемости студента - кумулятивная оценка успеваемости студента по предмету (например, за семестр).
11. Экспертная оценка успеваемости группы студентов - аналогично предыдущему, но по целой группе.
12. Контрольный вопрос для контроля студентов по изучаемым дисциплинам.
13. Ответ студента на контрольный вопрос 14. Правильный ответ контрольного вопроса.
15. Вопрос профессионального теста.
16. Ответ студента на вопрос профессионального теста (правильный или нет).
17. Правильный ответ профессионального теста.
18. Вопрос психологического теста.
19. Вариант ответа на вопрос психологического теста.
20. Правильный вариант ответа.
21. Ответ студента на вопрос психологического теста.
22. Экспертная оценка по результату профессионального теста.
23. Экспертная оценка по результату психологического теста.
24. Почтовое сообщение - сообщение по электронной почте.
25. Объявление.
26. Резюме (характеристика) студента.
27. Архивная запись по группе студентов.
28. Архивная запись по студенту.
29. Архивная запись отчетного задания, сданного студентом.

30. Архивная запись экспертных оценок. Каждая экспертная оценка соответствует отчетному заданию.
 31. Архивная запись результатов профессионального теста.
 32. Архивная запись результатов психологического теста.
 33. График успеваемости студента (зависимость средней оценки от временного промежутка).
 34. График успеваемости группы студентов.
 35. Архивная запись графиков успеваемости группы студентов.
- Б. Служебные:
1. Запись об активном соединении.
 2. Запись о пользовательской транзакции.
 3. Уровень доступа в систему.
 4. Учетная запись полномочий пользователя (для организации защиты от несанкционированного доступа).

4. Функциональные характеристики системы

1. Первоначальный ввод информации в БД.
2. Изменение содержания БД:
 - ввод новых данных,
 - изменение существующих данных,
 - архивация данных.
3. Осуществление поиска в БД по запросу пользователя.
4. Удаленный доступ к системе по протоколу ТСР/IP.
5. Обеспечение защиты и безопасности данных, в частности:
 - разграничение прав доступа к ресурсам сервера (владелец, группа и т. д.),
 - контроль вводимой информации,
 - обеспечение целостности БД.
6. Вывод найденной информации.

5. Цели и задачи системы

Система будет обеспечивать хранение, выдачу и обновление информации системы дистанционного обучения студентов и системы «Служба занятости в рамках вуза», а именно:

- обеспечивать информационную поддержку системы дистанционного обучения студентов;
- представлять и получать накопленную информацию по конкретным объектам;
- представлять и получать информацию от подсистемы обработки запросов пользователей;
- представлять и получать информацию от подсистемы контроля студентов по изучаемым на кафедре дисциплинам;
- обеспечивать разграничение доступа к информационным ресурсам системы;
- обеспечивать мониторинг активных и пассивных пользователей и системных событий;
- обеспечивать пользователей возможностью информационного обмена;
- обеспечивать связь между фирмами и службой (институтом);
- обеспечивать регулярное прохождение студентами профессиональных и психологических тестов;
- обеспечивать поиск данных по запросам.

6. Категории пользователей

При работе с системой на стадиях заполнения эксплуатации БД необходимо участие следующих категорий пользователей:

- администратора БД,
- группы экспертов.

Администратор системы осуществляет заполнение БД информацией, подготовленной учебной частью, деканатом или группой экспертов. Внесение изменений в БД системы осуществляется лишь администратором системы под руководством группы экспертов. Преподаватели и студенты являются внешними пользователями, работающими с системой в соответствии с ролями доступа в информационно-поисковом режиме.

Предоставляемые возможности пользователям системы студенту:

- ввод личных анкетных данных;
- просмотр экспертных оценок по отчетным заданиям и результатам тестов;

- прохождение психологических и профессиональных тестов;
- просмотр сводных таблиц и графиков;
- получение и сдача контрольных заданий;
- доступ к справочным материалам (данные из службы удаленного обучения, а именно методическое обеспечение);
- просмотр сообщений и внесение изменений в сообщения доски объявлений;
- поиск вакансии в БД по запросу;

эксперту (преподавателю):

- предоставление экспертной оценки, а также изменение ее;
- просмотр других оценок;
- просмотр программы курса и внесение изменений в нее;
- ввод контрольных заданий и назначение их студенту;
- контроль ответов на задания;
- доступ к интеллектуальным ресурсам;
- составление резюме (характеристик);

деканату (декану, зам. декана и т. д.):

- просмотр программы курса;
- просмотр динамики успеваемости курса, группы, отдельного студента;
- просмотр сводных таблиц и графиков;
- просмотр экспертных оценок и характеристик преподавателей;

администратору:

- определение прав доступа;
- ввод и корректировка системных данных;
- контроль работы системы;
- осуществление контроля защиты системы от несанкционированного доступа;
- изменение физической модели данных системы;

оператору:

- составление сводных таблиц и графиков;
- заполнение полей БД системы (ввод информации);

фирме:

- поиск в БД данных о студенте по запросу;
- просмотр резюме студентов.

Лабораторная работа №2 «Изучение основных функций пакета BPwin»










BPwin позволяет аналитику создавать сложные модели бизнес-процессов при минимальных усилиях. BPwin поддерживает три методологии - IDEF0, IDEF3 и DFD. Каждая из них призвана решать свои специфические задачи. Также можно строить смешанные модели.

Модель в BPwin рассматривается как совокупность работ, каждая из которых оперирует с некоторым набором данных. Работы изображаются в виде прямоугольников (блоков), данные - в виде стрелок (дуг).

При запуске BPWin по умолчанию появляется основная панель инструментов, палитра инструментов и Model Explorer.

Таблица 1

Описание элементов управления основной панели инструментов.

Элемент управления	Описание	Соответствующий пункт меню
	Создать новую модель	File->New
	Открыть модель	File-> Open
	Сохранить модель	File->Save
	Напечатать модель	File->Print
	Выбор масштаба	View-> Zoom
	Масштабирование	View-> Zoom
	Проверка орфографии	Tools->Spelling
	Включение выключение навигатора модели	View-> Model Explorer
	Включение выключение дополнительной панели инструментов	ModelMart

При создании новой модели возникает диалог, в котором следует указать, будет ли создана модель заново, или она будет открыта, внести имя модели и выбрать методологию, в которой будет построена модель (рисунок 1).

BPWin поддерживает три методологии – IDEF0, IDEF3 и DFD. В BPWin возможно построение смешанных моделей, т. е. модель может содержать одновременно как диаграммы IDEF0, так и IDEF3 и DFD. Состав палитры инструментов изменяется автоматически, когда происходит переключение с одной нотации на другую.

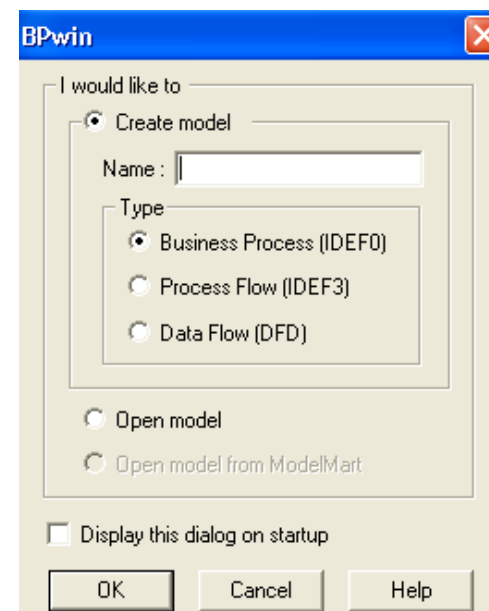


Рисунок 1.

На основной панели инструментов (либо в любом желаемом месте экрана) расположены инструменты редактора BPWin:

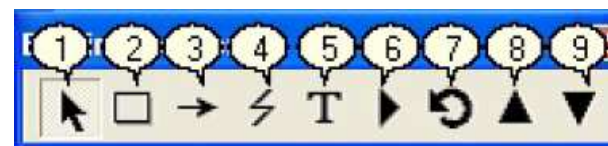


Рисунок 2.

1. Pointer Tool – используется для выбора и определения позиции объектов добавленных в диаграмму.
2. Activity Box Tool – используется для установки блоков в диаграмме.
3. Arrow Tool – используется, чтобы устанавливать дуги в диаграмме.
4. Squiggle Tool – используется для создания тильды (squiggle), которая соединяет дугу с ее названием.
5. Text Block Tool – используется для создания текстовых блоков.
6. Diagram Dictionary Editor – открывает диалоговое окно Diagram Dictionary Editor, где можно перейти на какую-либо диаграмму или создать новую диаграмму.
7. Go to Sibling Diagram – используется для отображения следующей диаграммы того же уровня.
8. Go to Parent Diagram – переход на родительскую диаграмму.
9. Go to Child Diagram – используется, чтобы отобразить диаграмму потомка или разложить выделенный блок на диаграмму потомка.

Построение модели системы должно начинаться с изучения всех документов, описывающих ее функциональные возможности. Одним из таких документов является техническое задание, а именно разделы «Назначение разработки», «Цели и задачи системы» и «Функциональные характеристики системы».

IDEF0-модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения. Для внесения области, цели и точки зрения в модели IDEF0 в BPwin следует выбрать пункт меню **Model/Model Properties**, вызывающий диалог **Model Properties** (рисунок 3).

В закладке **Purpose** следует внести цель и точку зрения, а в закладку **Definition** – определение модели и описание области. В закладке **Status** того же диалога можно описать статус модели (черновой вариант, рабочий, окончательный и т.д.), время создания и последнего редактирования (отслеживается в дальнейшем автоматически по системной дате). В закладке **Source** описываются источники информации для построения модели (например, "Опрос

экспертов предметной области и анализ документации"). Закладка **General** служит для внесения имени проекта и модели, имени и инициалов автора и временных рамок.

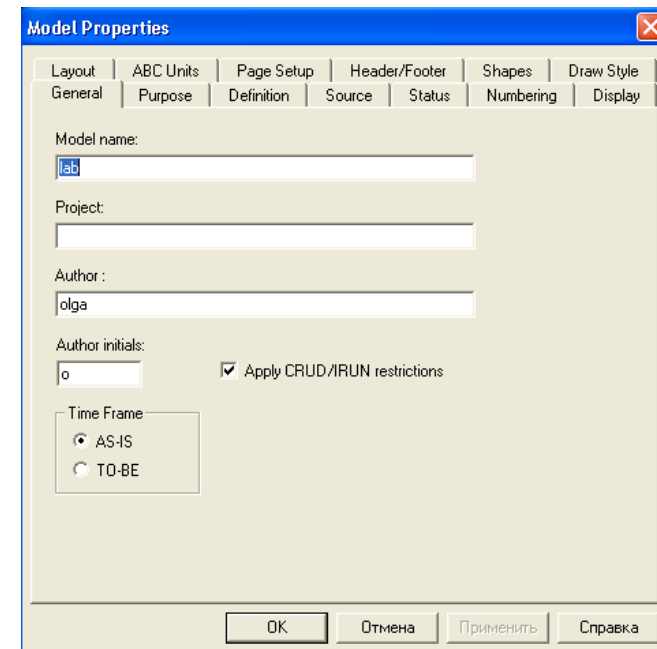


Рисунок 3.

Работа с блоками и дугами

Методология IDEF0 предписывает построение иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом (контекстная диаграмма), после чего проводится декомпозиция – система разбивается на подсистемы, и каждая подсистема описывается отдельно.

Контекстная диаграмма

После создания проекта мы видим окно с единственным блоком. Назовем данный блок «Обслуживание клиента системы».

Для этого необходимо щелкнуть правой клавишей мыши по блоку и выбрать команду **Name** и в диалоговом окне ввести название (рисунок 2).

По отношению к тексту применимы стандартные способы форматирования, для их использования следует выполнить команду контекстно-зависимого меню **Font**. Вы можете изменять размеры блока. Для изменения высоты необходимо перетащить мышью верхнюю или нижнюю стороны блока, аналогично меняется размер по горизонтали.

После создания объекта необходимо обозначить его основные функции и элементы взаимодействия. В Vrpwin этими элементами являются дуги. Для построения дуг управления, входа, выхода и механизмов необходимо выбрать инструмент (Arrow Tool), затем щелкнуть мышью со стороны периметра и второй щелчок с соответствующей стороны блока. Для построения дуги выхода щелкнуть первоначально справа стороны блока, затем со стороны периметра.

То с какой стороны дуга подходит к блоку является своего рода значением данной дуги.

Слева – вход в блок

Справа – выход в блок

Сверху – управляющая информация

Снизу – механизмы (средства производства)

Дугам, как и блокам можно придавать свои имена. Для этого необходимо: щелкнуть правой клавишей мыши по блоку и выбрать команду **Name** и в диалоговом окне ввести название дуги.

Для удаления блока и дуги или текста необходимо их выделить щелчком левой кнопки мыши и нажать клавишу **Delete**, а затем подтвердить намерения по поводу удаления.

Декомпозиция

После создания контекстной диаграммы необходимо расписать работу отдельных участков. Для декомпозиции необходимо в браузере щелкнуть левой кнопкой мыши на имени

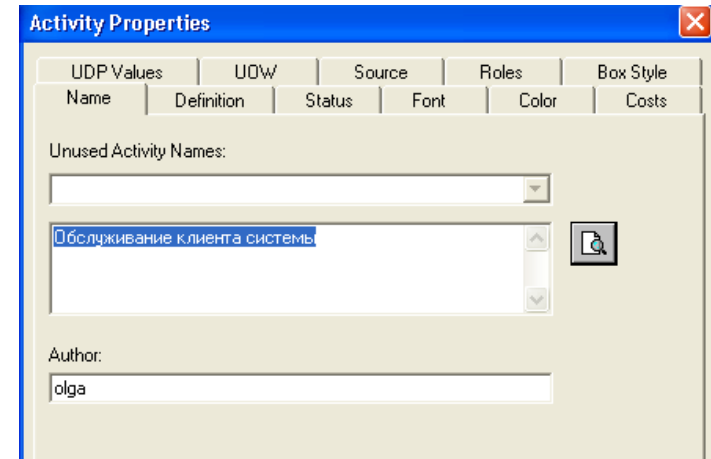


Рисунок 2.

диаграммы, а затем нажать кнопку (Go to Child Diagram), затем в диалоговом окне (рисунок 3) ввести необходимое количество блоков и тип диаграммы:

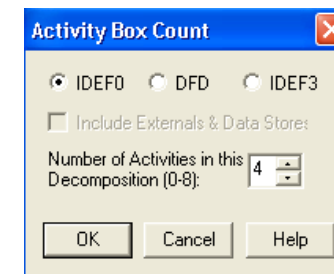


Рисунок 3.

Если в дальнейшем необходимо добавить блоки на диаграмме, то необходимо выбрать инструмент (**Activity Box Tool**) и щелкнуть мышью в нужном месте диаграммы.

После декомпозиции необходимо соединить получившиеся блоки дугами. Для этого необходимо выбрать инструмент, щелкнуть мышью по исходной стороне блока затем по конечной стороне следующего блока. Аналогично строятся разветвления и слияния дуг.

ЗАДАНИЕ

Рассмотрим технологию ее построения на примере системы «Служба занятости в рамках вуза», основные возможности которой были описаны в лабораторной работе № 1.

Сформулируем цель моделирования: описать функционирование системы, которое было бы понятно ее пользователю, не вдаваясь в подробности, связанные с реализацией. Модель будем строить с точки зрения пользователей (студент, преподаватель, администратор, деканат, фирма).

Начнем с построения контекстной IDEF0-диаграммы. Согласно описанию системы основной функцией является обслуживание ее клиентов посредством обработки запросов, от них поступающих. Таким образом, определим единственную работу контекстной диаграммы как «Обслужить клиента системы». Далее определим входные и выходные данные, а так же механизмы и управление.

Для того чтобы обслужить клиента, необходимо зарегистрировать его в системе, открыть доступ к БД и обработать его запрос. В качестве входных данных будут использоваться «имя клиента», «пароль клиента», «исходная БД», «запрос клиента».

Выполнение запроса ведет либо к получению информации от системы, либо к изменению содержимого БД (например, при составлении экспертных оценок), поэтому выходными данными будут являться «отчеты» и «измененная БД». Процесс обработки запросов будет выполняться монитором системы под контролем администратора.

Таким образом, определим контекстную диаграмму системы (рисунок 4).

Проведем декомпозицию контекстной диаграммы, описав последовательность обслуживания клиента:

1. Определение уровня доступа в систему.
 2. Выбор подсистемы.
 3. Обращение к подсистеме.
 4. Изменение БД (при необходимости).
- Получим диаграмму, изображенную на рисунке 5.

Закончив декомпозицию контекстной диаграммы, переходят к декомпозиции диаграммы следующего уровня. Обычно при рассмотрении третьего и более нижних уровней модели возвращаются к родительским диаграммам и корректируют их.

Декомпозируем последовательно все блоки полученной диаграммы. Первым этапом при определении уровня доступа в систему является определение категории пользователя. По имени клиента осуществляется поиск в базе пользователей, определяя его категорию. Согласно определенной категории выясняются полномочия, предоставляемые пользователю системы.

Далее проводится процедура доступа в систему, проверяя имя и пароль доступа. Объединяя информацию о полномочиях и уровне доступа в систему, для пользователя формируется набор разрешенных действий. Таким образом, определение уровня доступа в систему будет выглядеть как показано на рисунке 6.

После прохождения процедуры доступа в систему монитор анализирует запрос клиента, выбирая подсистему, которая будет обрабатывать запрос. Декомпозиция работы «Обращение к подсистеме» не отвечает цели и точке зрения модели. Пользователя системы не интересуют внутренние алгоритмы ее работы. В данном случае ему важно, что выбор подсистемы будет произведен автоматически, без его вмешательства, поэтому декомпозиция обращения к подсистеме только усложнит модель.

Декомпозируем работу «Обработка запроса клиента», выполняемую подсистемой обработки запросов, определения категорий и полномочий пользователей. Перед осуществлением поиска ответа на запрос необходимо открыть БД (подключиться к ней). В общем случае БД может находиться на удаленном сервере, поэтому может потребоваться установление соединения с ней. Определим последовательность работ:

1. Открытие БД.
2. Выполнение запроса.
3. Генерация отчетов.

После открытия БД необходимо сообщить системе об установлении соединения с БД, после чего выполнить запрос и сгенерировать отчеты для пользователя (рисунок 7). Перейдем к декомпозиции последнего блока «Изменение БД» (рисунок 8).

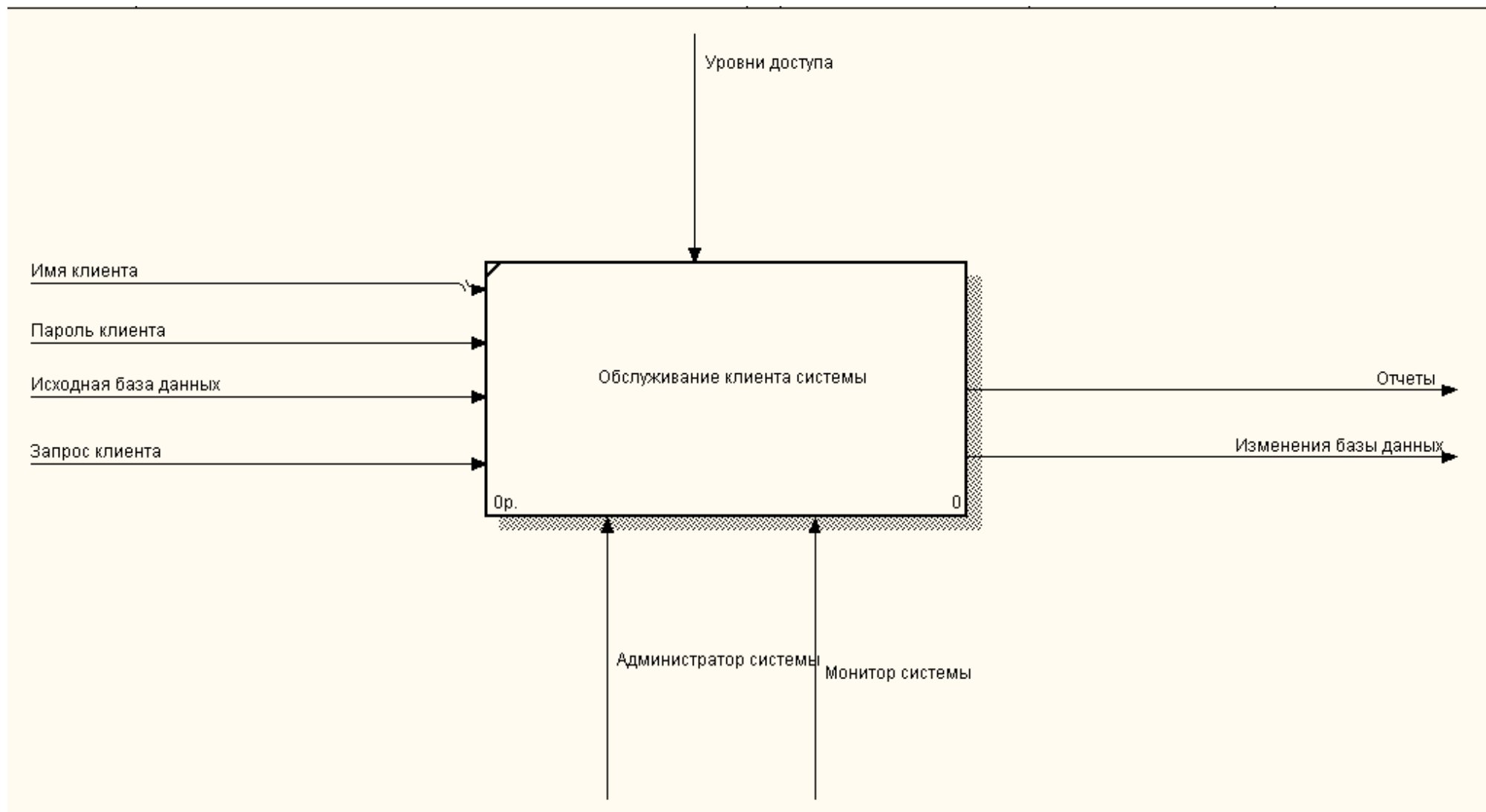


Рисунок 4. Контекстная диаграмма системы

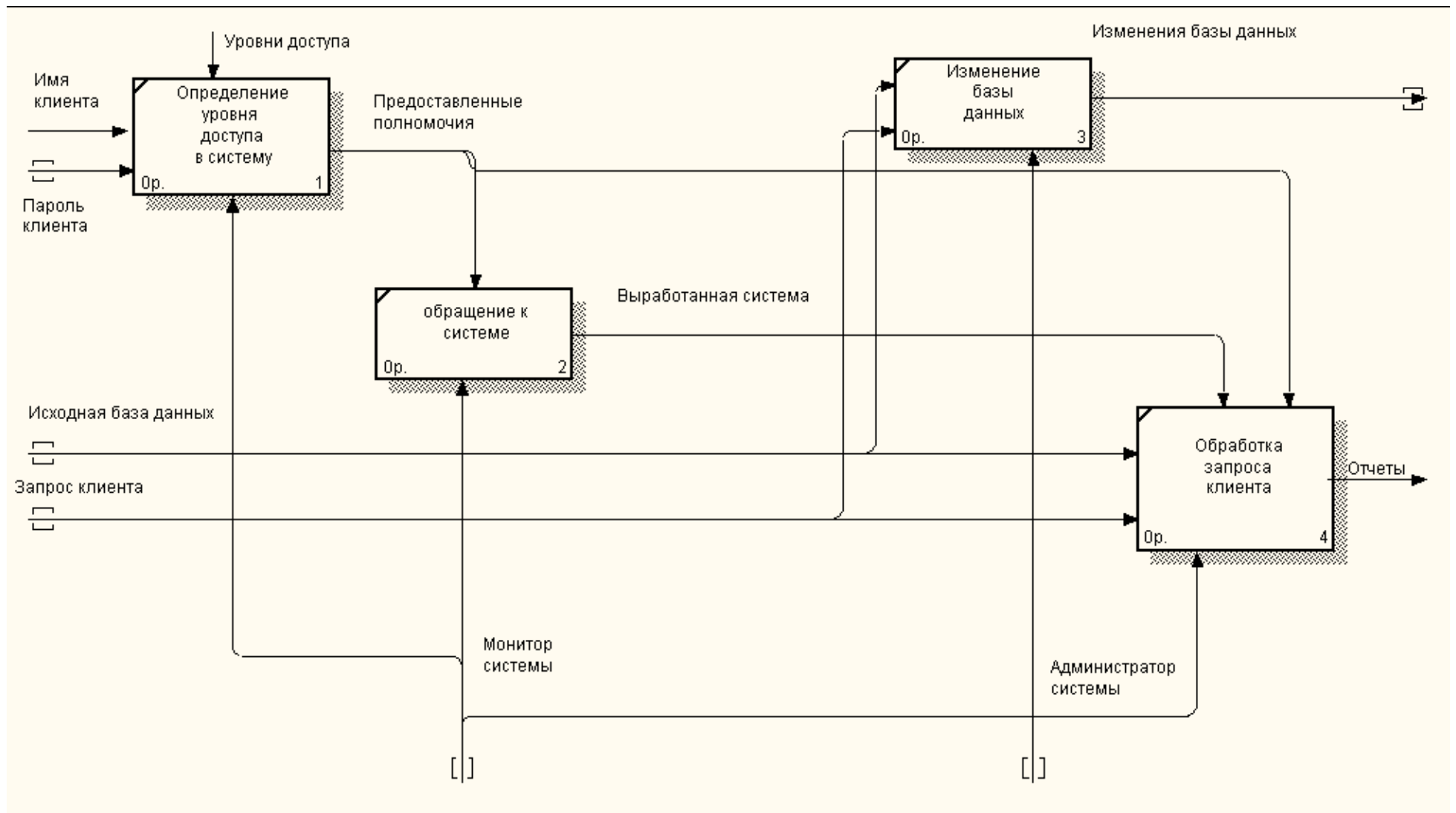


Рисунок 5. Декомпозиция работы «Обслуживание клиента системы»

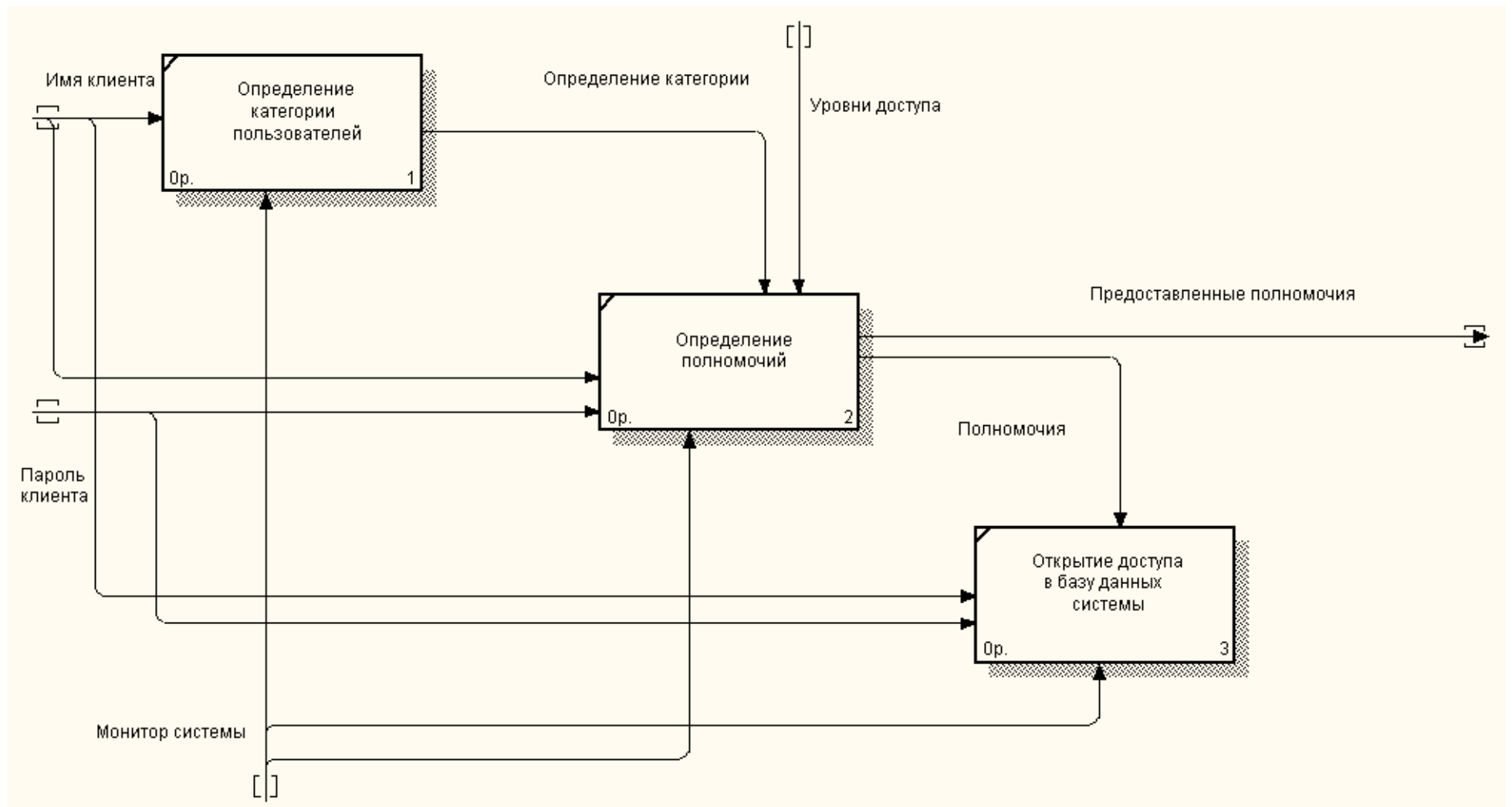


Рисунок 6. Декомпозиция работы «Определение уровня доступа в систему»

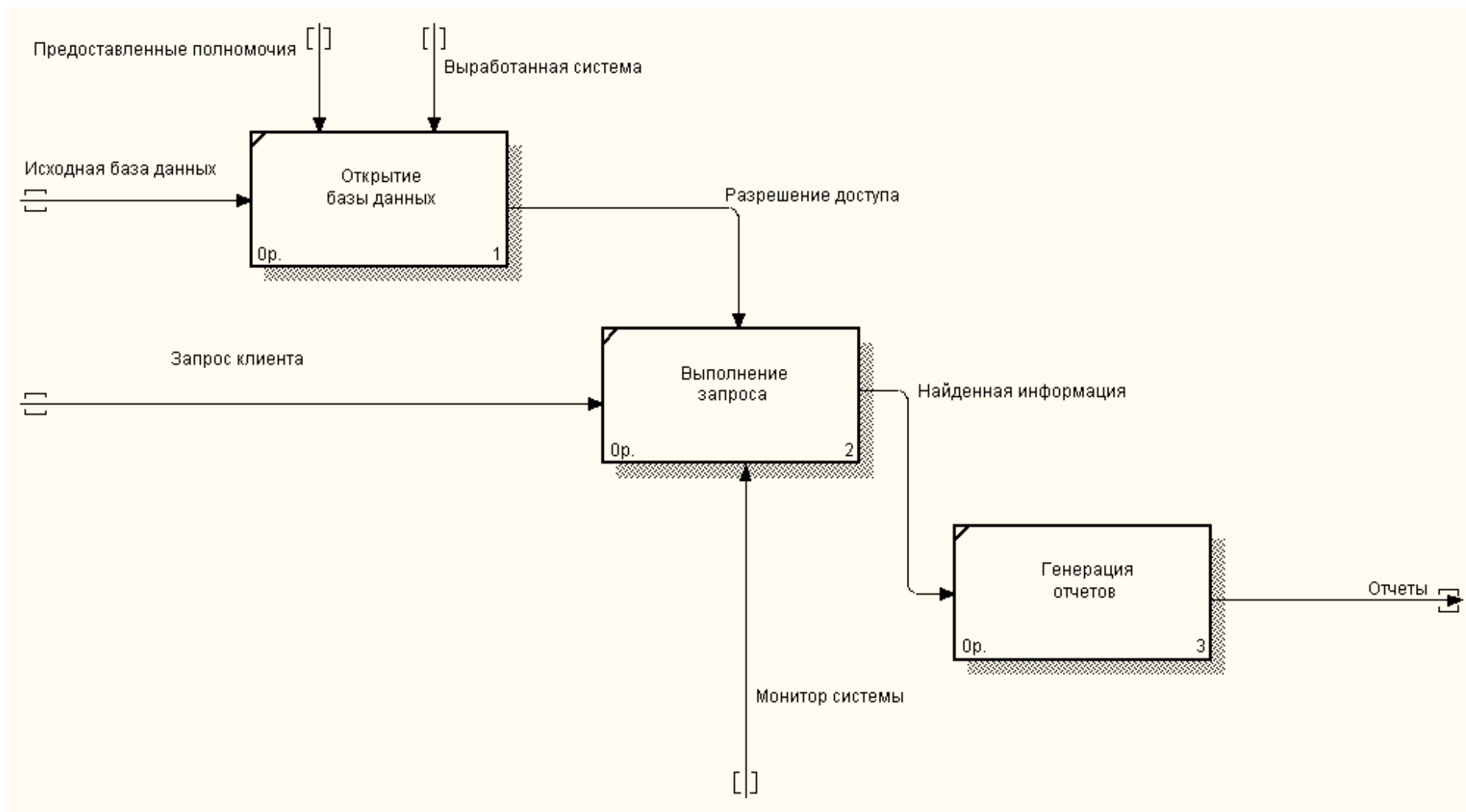


Рисунок 7. Декомпозиция работы «Обработка запроса клиента»

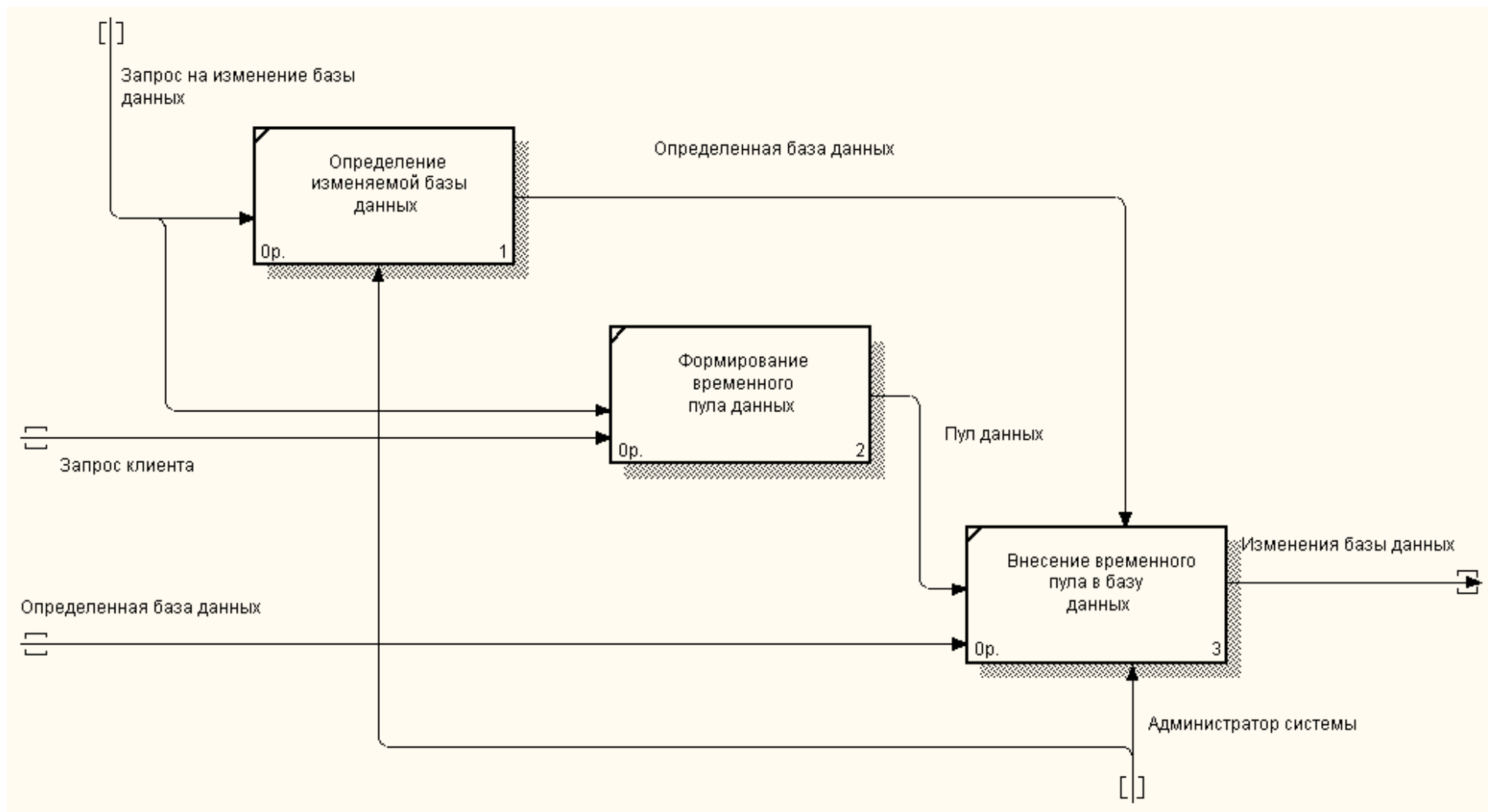


Рисунок 8. Декомпозиция работы «Изменение базы данных»

Лабораторная работа №3

Создание контекстной диаграммы и проведение декомпозиции

Задание: Построить модель по образцу

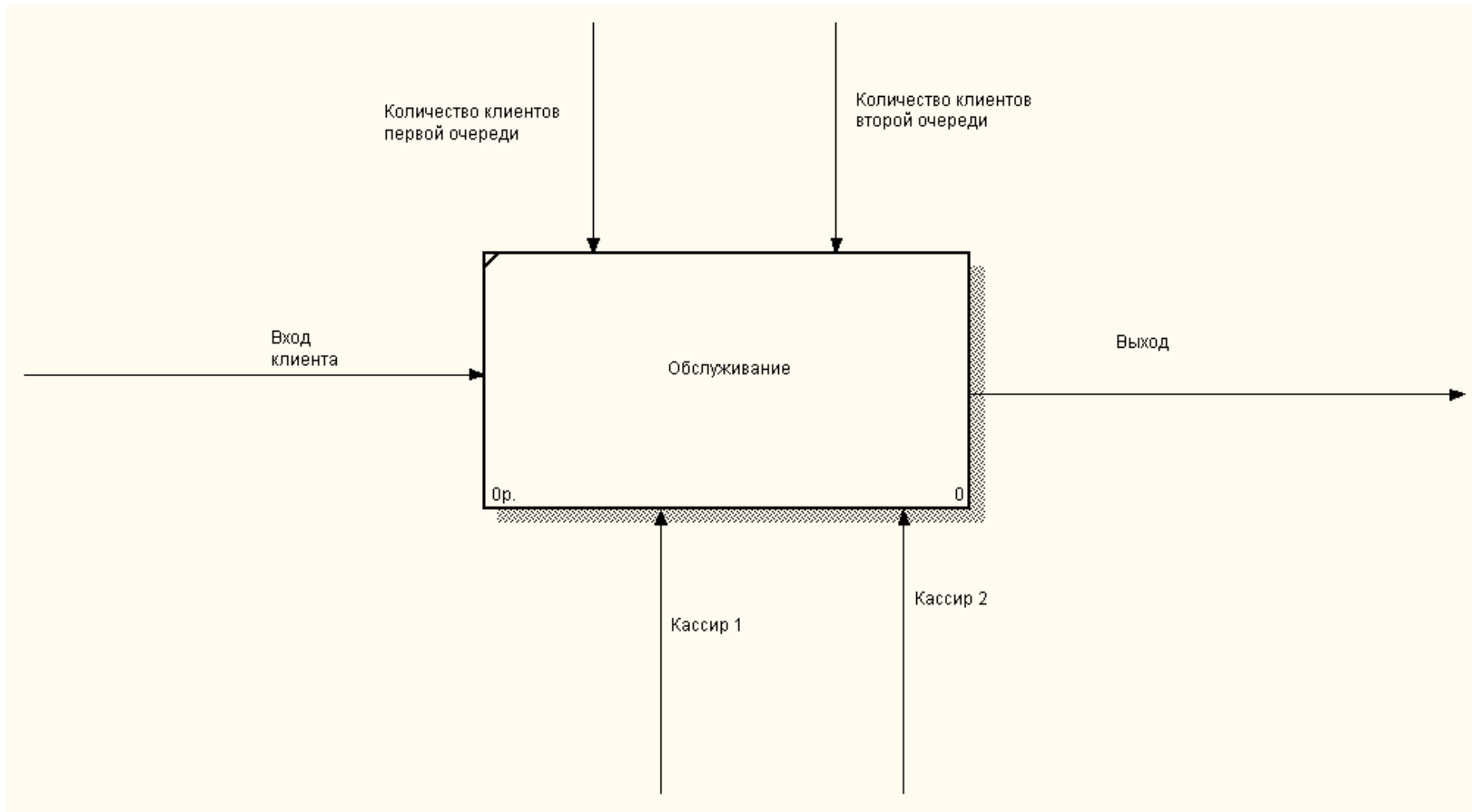


Рисунок 1

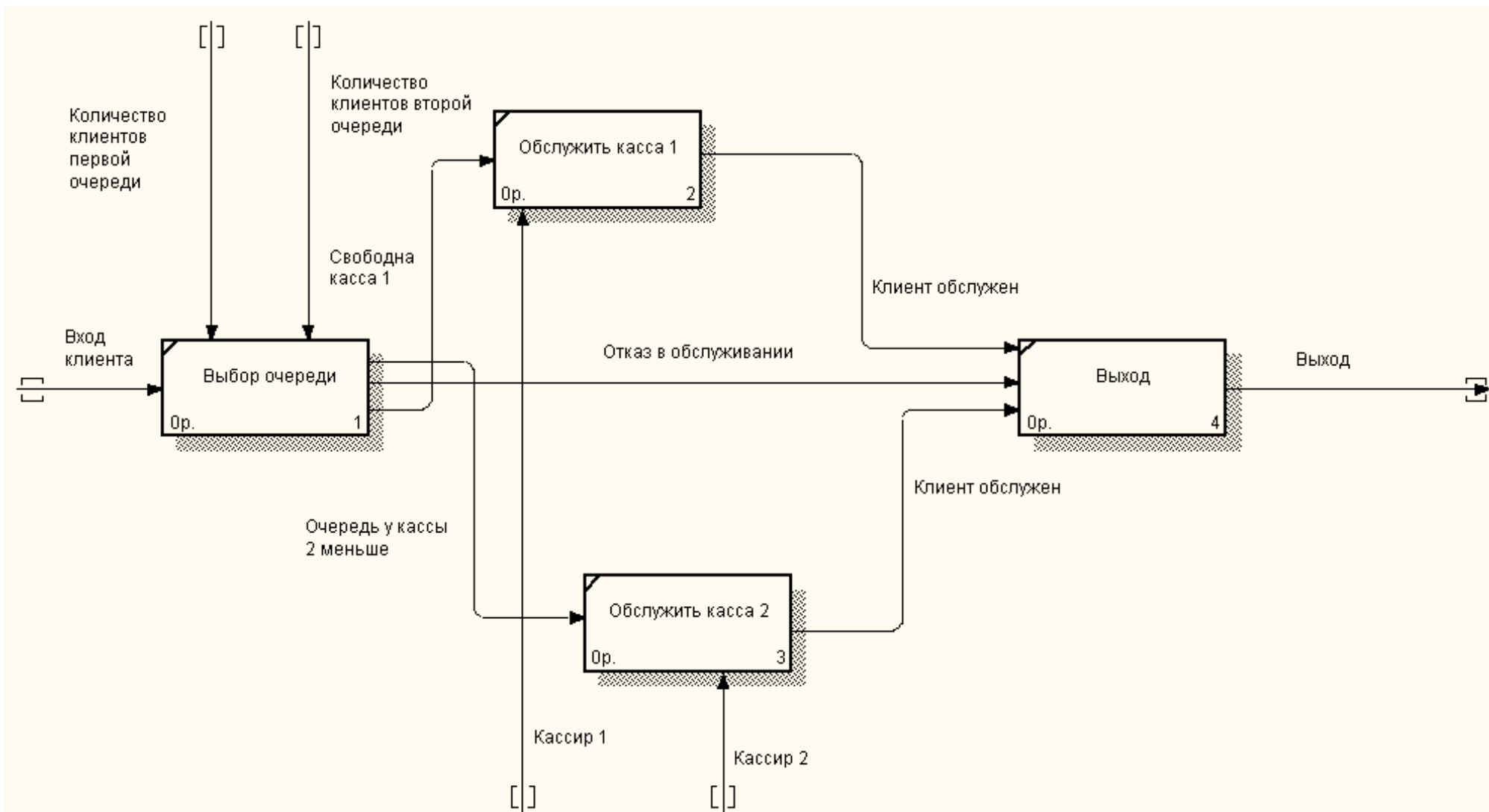


Рисунок 2.

Лабораторная работа №4

Дополнение моделей процессов диаграммами DFD


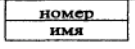
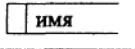
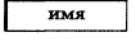
Потоки данных (Data Flow Diagrams) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ.

DFD описывает:

1. функции обработки информации (работы);
2. документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации;
3. внешние ссылки (external reference), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
4. таблицы для хранения документов (хранилища данных, data store).

Для построения диаграмм DFD в BPWin используется нотация Гейна - Сарсона (таблица 1).

Таблица 1

Компонент	Обозначение
Поток данных	
Процесс	
Хранилище	
Внешняя сущность	

Создание диаграммы DFD.

Для создания диаграммы DFD необходимо в диалоге New Model (рисунок 1) выбрать кнопку DFD.

Пример 1. Диаграммы DFD можно использовать как дополнение к диаграммам IDEF0 для описания документооборота и обработки информации. Рассмотрим работу «Обработка запросов клиента» из лабораторной работы №1.

Запросы в систему поступают от пользователей, поэтому запросы от каждой категории будут обрабатываться отдельно. Выделим внешние сущности диаграммы согласно каждой категории пользователей, определяя потоки данных, которыми они обмениваются с системой. Получим диаграмму, изображенную на рисунке 2.

Согласно описанию системы проведем декомпозицию полученных блоков (рис. 3 - 6).

Все процессы обработки запросов контролируются и выполняются монитором системы, поэтому стрелка-механизм «Монитор системы» будет повторяться на декомпозированных диаграммах. Точка зрения модели, определенная в лабораторной работе № 1, не требует рассмотрения внутренних особенностей функционирования системы, поэтому затуннелируем стрелку «Монитор системы» с тем, чтобы не переносить ее на диаграммы нижнего уровня.

Проведем анализ полученных диаграмм. Рассматривая диаграмму, изображенную на рисунке 2, необходимо отметить наличие в ней лишнего блока «Обработать запрос администратора».

Администратор не обслуживается системой как обычный клиент, он обеспечивает ее мониторинг. Администратор может добавлять пароли, изменять уровень доступа в систему, добавлять нового пользователя и т. д. С точки зрения клиента системы, деятельность администратора является второстепенной, поэтому на всех диаграммах отсутствует описание функций администратора, представляя его влияние как «механизм» для других функций. Поэтому принимаем решение удалить работу «Обработать запрос администратора».

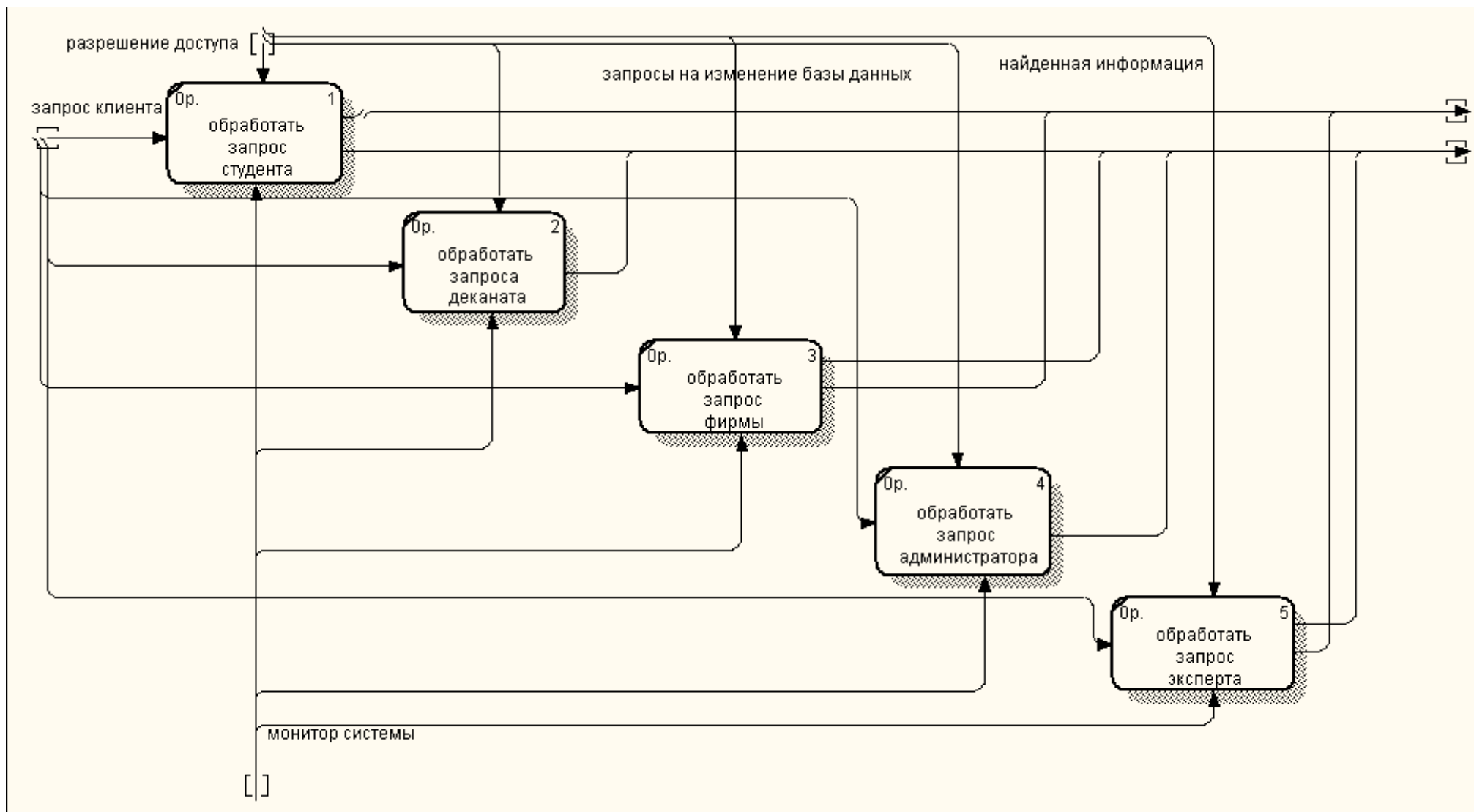


Рисунок 2. DFD-декомпозиция работы «Выполнение запроса»

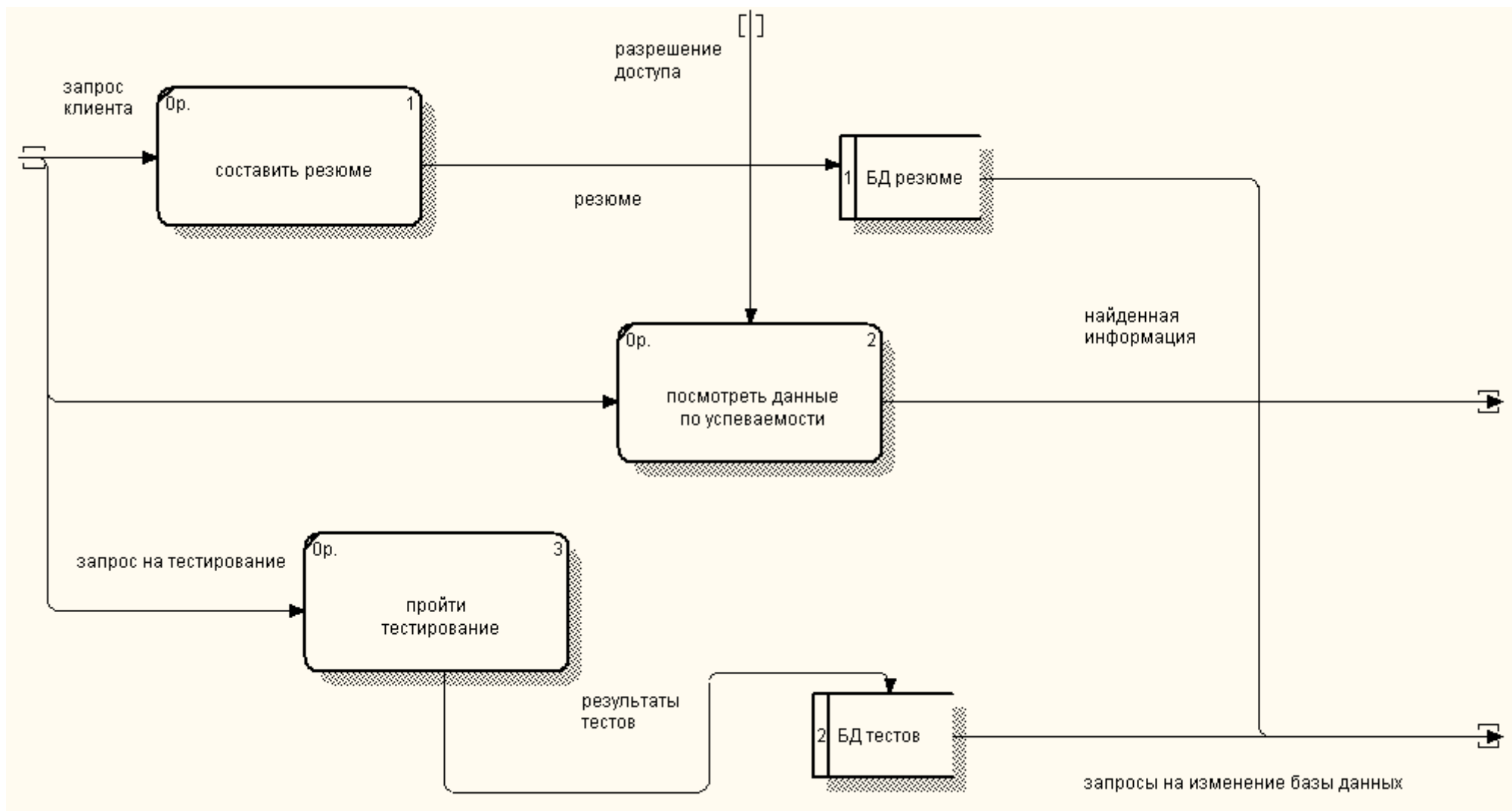


Рисунок 3 Декомпозиция работы «Обработать запрос студента»

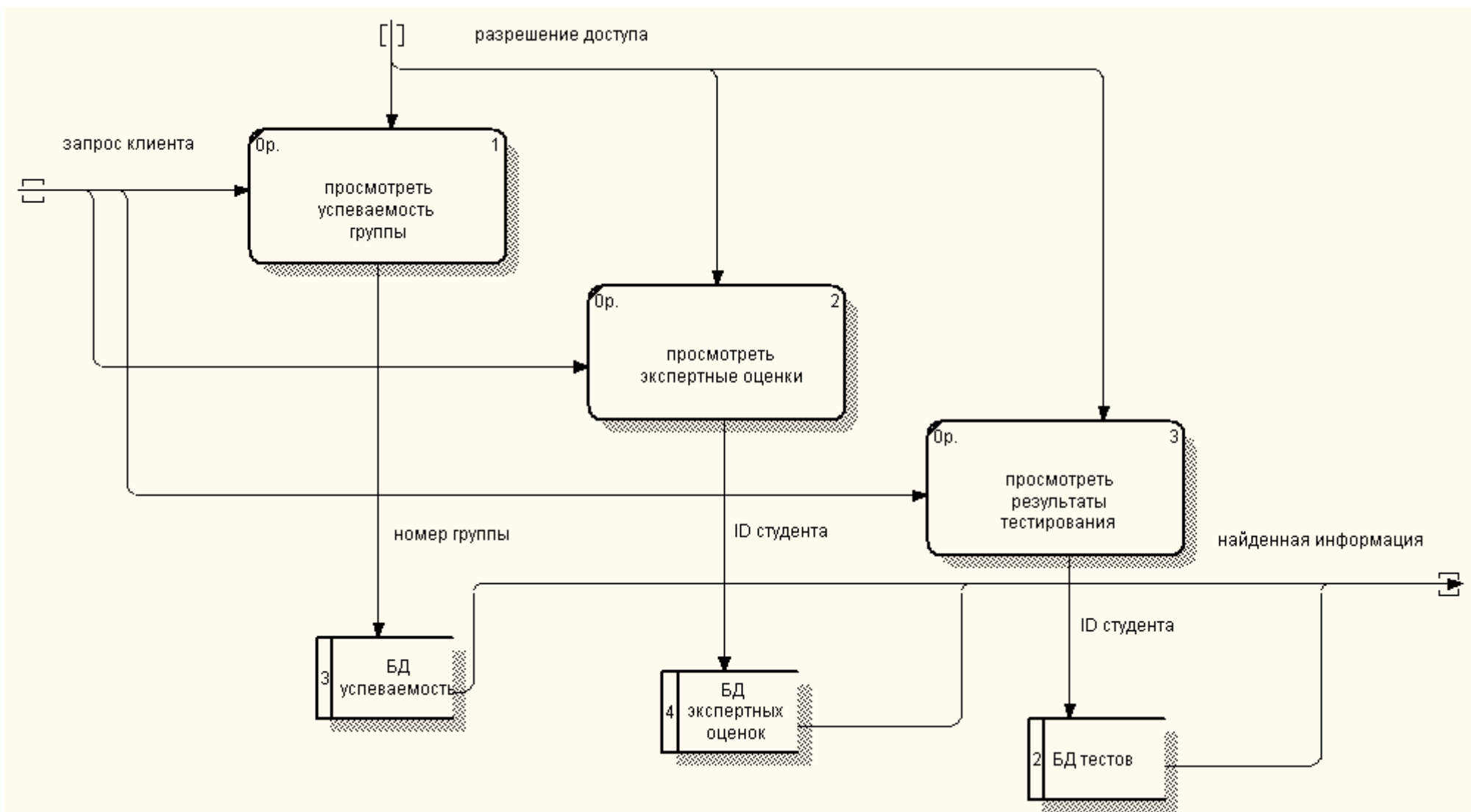


Рисунок 4 Декомпозиция работы «Обработать запрос декана»

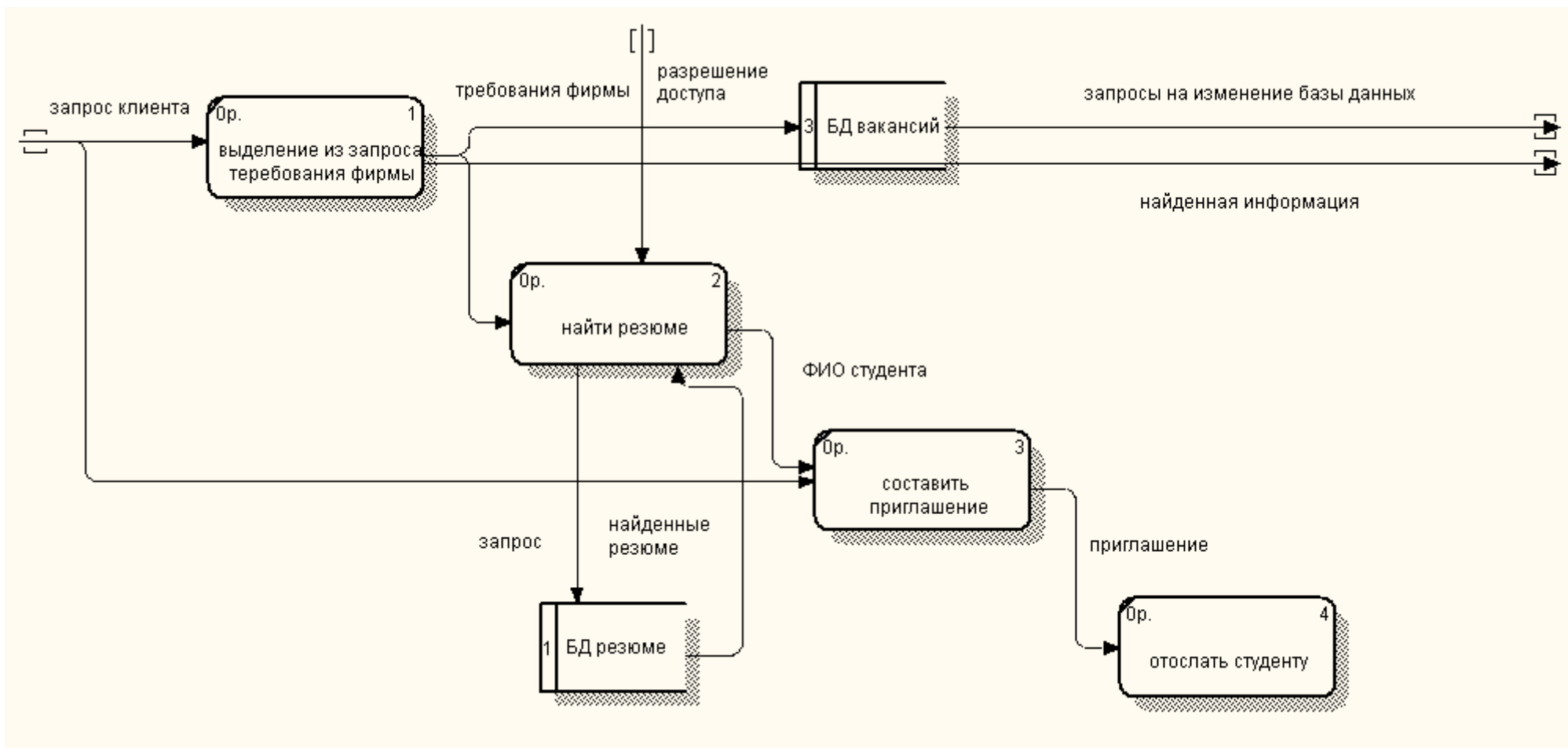


Рисунок 5 Декомпозиция работы «Обработать запрос фирмы»

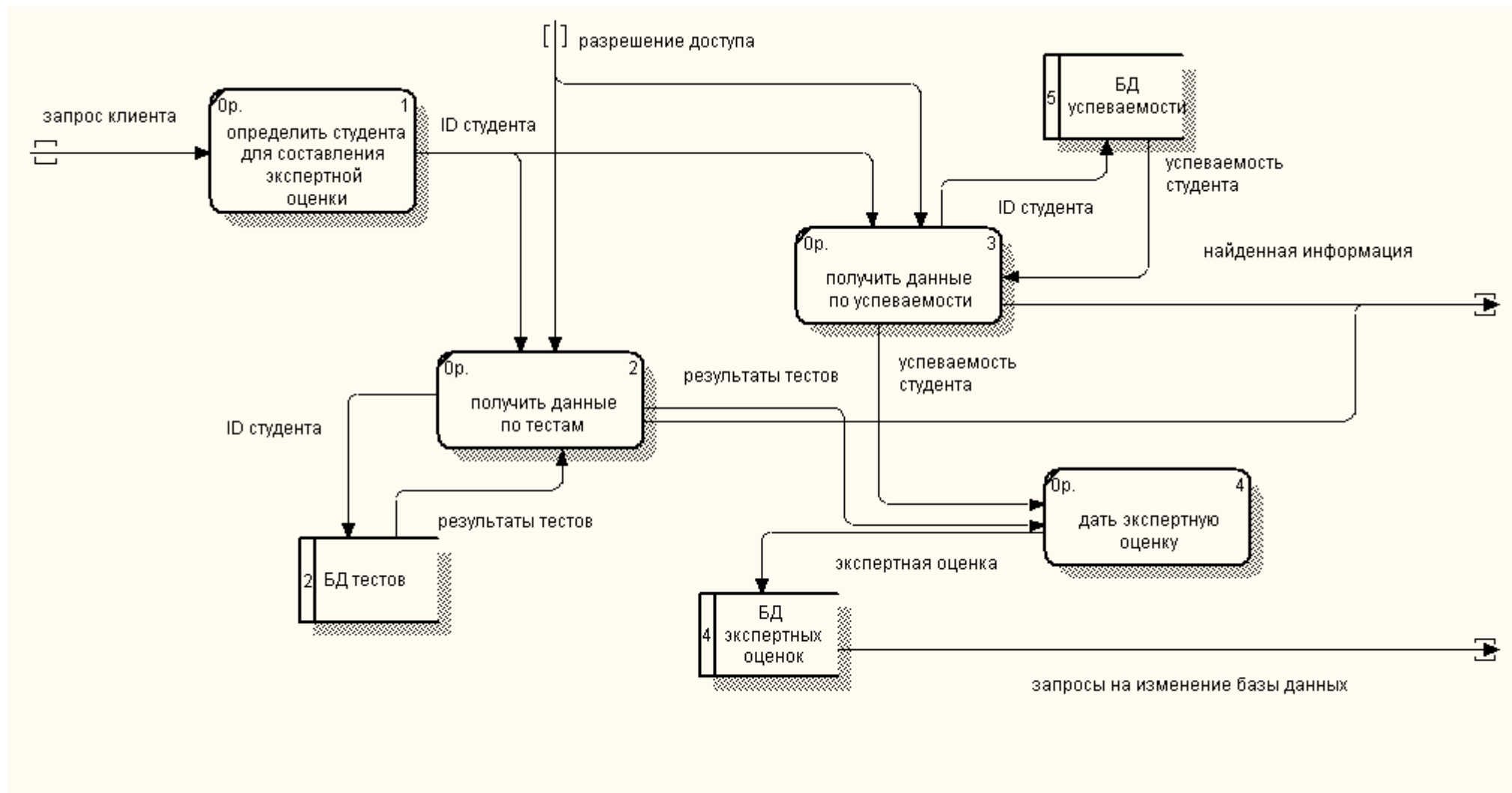


Рисунок 6. Декомпозиция работы «Обработать запрос эксперта»

3 Методические указания по самостоятельной работе

Для успешного освоения курса «Технология разработки программного обеспечения» необходима самостоятельная работа. В настоящее время актуальными становятся требования к личным качествам современного студента – умению самостоятельно пополнять и обновлять знания, вести самостоятельный поиск необходимого материала, быть творческой личностью.

Самостоятельную работу по освоению дисциплины обучающимся следует начинать с изучения содержания рабочей учебной программы дисциплины, цели и задач, структуры и содержания курса, основной и дополнительной литературы, рекомендованной для самостоятельной работы.

Самостоятельная учебная деятельность является необходимым условием успешного обучения. Многие профессиональные навыки, способность мыслить и обобщать, делать выводы и строить суждения, выступать и слушать других, – все это развивается в процессе самостоятельной работы студентов.

Самостоятельная работа по освоению дисциплины включает:

- самостоятельное изучение разделов;
- самоподготовку (проработку и повторение лекционного материала и материала учебников и учебных пособий);
- подготовку к лабораторным работам;
- подготовку к рубежному и итоговому контролю.

Самостоятельная учебная работа – условие успешного окончания высшего учебного заведения. Она является равноправной формой учебных занятий, наряду с лекциями, семинарами, экзаменами и зачетами, но реализуемая во внеаудиторное время.

Эффективность аудиторных занятий во многом зависит от того, насколько умело студенты организуют в ходе них свою самостоятельную учебную познавательную деятельность. Такая работа также способствует самообразованию и самовоспитанию, осуществляемому в интересах повышения профессиональных компетенций, общей эрудиции и формировании личностных качеств.

Самостоятельная работа реализуется:

1. непосредственно в процессе аудиторных занятий – на лекциях, лабораторных занятиях, при проведении рубежного контроля;
2. в контакте с преподавателем вне рамок расписания – на консультациях по учебным вопросам, при ликвидации задолженностей, при выполнении индивидуальных заданий;
3. в библиотеке, дома, в общежитии, на кафедре при выполнении студентом учебных задач.

В процессе проведения самостоятельной работы необходимо производить подбор литературных источников, научной периодической печати и т.д.

4 Методические указания по итоговому контролю

Итоговый контроль знаний по дисциплине «Технология разработки программного обеспечения» проводится в форме экзамена. Для подготовки к итоговому контролю знаний по дисциплине «Технология разработки программного обеспечения» обучающиеся используют перечень вопросов, приведенный в фонде оценочных средств. Экзамен проводится в устной форме. В экзаменационный билет включен один теоретический вопрос. На подготовку студенту отводится 20-25 минут. На дифференцированном зачете ответы обучающегося оцениваются с учетом их полноты, правильности и аргументированности с учетом шкалы оценивания.

Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически его излагает,

умеет тесно увязывать теорию с практикой, свободно справляется с вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе профессиональные термины, правильно обосновывает принятое решение.

Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов, владеет необходимыми навыками и приемами их выполнения.

Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.

Оценка «неудовлетворительно» выставляется студенту за отсутствие знаний по дисциплине, представления по вопросу, непонимание материала по дисциплине, наличие коммуникативных «барьеров» в общении, отсутствие ответа на предложенный вопрос.

5 Учебно-методическое обеспечение дисциплины

5.1 Основная литература

1. Золотов, С.Ю. Проектирование информационных систем : учебное пособие / С.Ю. Золотов ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Эль Контент, 2013. - 88 с. : табл., схем. - ISBN 978-5-4332-0083-8 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=208706](http://biblioclub.ru/index.php?page=book&id=208706), коэффициент книгообеспеченности 0,8

2. Стасышин, В.М. Проектирование информационных систем и баз данных : учебное пособие / В.М. Стасышин. - Новосибирск : НГТУ, 2012. - 100 с. - ISBN 978-5-7782-2121-5 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=228774](http://biblioclub.ru/index.php?page=book&id=228774), коэффициент книгообеспеченности 0,8

5.2 Дополнительная литература

1. Проектирование информационных систем [Текст] : учебное пособие / Г. Н. Исаев.- 2-е изд., стер. - Москва : Омега - Л, 2015. - 424 с. - (Высшее техническое образование) - ISBN 978-5-370-03507-4. (ОГТИ СПО 16), коэффициент книгообеспеченности 1

2. Абрамов, Г.В. Проектирование информационных систем : учебное пособие / Г.В. Абрамов, И.Е. Медведкова, Л.А. Коробова. - Воронеж : Воронежский государственный университет инженерных технологий, 2012. - 172 с. : ил.,табл., схем. - ISBN 978-5-89448-953-7 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=141626](http://biblioclub.ru/index.php?page=book&id=141626), коэффициент книгообеспеченности 1

3. Заика, А.А. Разработка прикладных решений для платформы 1С:Предприятие 8.2 в режиме «Управляемое приложение» / А.А. Заика. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 239 с. : ил. ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=429019](http://biblioclub.ru/index.php?page=book&id=429019), коэффициент книгообеспеченности 1.

5.3 Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий»
2. Журнал «Информационные технологии и вычислительные системы»
3. Журнал «Стандарты и качество»

4. Журнал «Прикладная информатика»

5.4 Интернет-ресурсы

5.4.1 Современные профессиональные базы данных и информационные справочные системы:

1. Информационная система «Единое окно доступа к образовательным ресурсам»- <http://window.edu.ru/>
2. КиберЛенинка - <https://cyberleninka.ru/>
3. Университетская информационная система Россия– uisrussia.msu.ru
4. Бесплатная база данных ГОСТ– <https://docplan.ru/>

5.4.2 Тематически профессиональные базы данных и информационные справочные системы:

1. Портал искусственного интеллекта – [AIPortal](#)
2. Web-технологии – [Web-технологии](#)
3. Электронная библиотека Института прикладной математики им. М.В. Келдыша – [Электронная библиотека публикаций Института прикладной математики им. М.В. Келдыша РАН](#)

5.4.3 Электронные библиотечные системы

1. ЭБС «Университетская библиотека онлайн» – <http://www.biblioclub.ru/>
2. ЭБС Znanium.com – <https://znanium.com/>

5.4.4 Дополнительные Интернет-ресурсы

1. <https://www.ixbt.com> - Интернет-издание о компьютерной технике, информационных технологиях и программных продуктах. На сайте публикуются новости IT, статьи с обзорами и тестами компьютерных комплектующих и программного обеспечения.
2. <http://www.intuit.ru> – ИНТУИТ – Национальный открытый университет.
3. <http://cppstudio.com/> - Основы программирования на языках Си и C++.
4. http://citforum.ru/SE/project/arkhipenkov_lectures – Лекции по управлению программными проектами автор А. Архипенков

5.5 Программное обеспечение, профессиональные базы данных и информационные справочные системы современных информационных технологий

Тип программного обеспечения	Наименование	Схема лицензирования, режим доступа
Операционная система	Microsoft Windows	Подписка Enrollment for Education Solutions (EES) по государственному контракту: № 2К/17 от 02.06.2017 г.;
Текстовый редактор	Notepad++	Свободное ПО, https://notepad-plus-plus.org/
Интернет-браузер	Google Chrome	Бесплатное ПО, http://www.google.com/intl/ru/policies/terms/

Векторный графический редактор, редактор диаграмм и блок-схем	Microsoft Visio Standard 2007	Сертификат MicrosoftOpenLicense № 46284547 от 18.12.2009 г., академическая лицензия на рабочее место
Интегрированная среда разработки программного обеспечения	Microsoft Visual Studio Professional 2008	Сертификат MicrosoftOpenLicense № 46284547 от 18.12.2009 г., академическая лицензия на рабочее место
	Embarcadero RAD Studio 2010 Professional	Образовательная лицензия по государственному контракту № 32/09 от 17.12.2009 г., сетевой конкурентный доступ
	Turbo Pascal 7.0 for DOS	Образовательная лицензия по государственному контракту № 34/10 от 10.12.2010 г., лицензия на рабочее место
	Borland C++ 3.1 for DOS	Образовательная лицензия по государственному контракту № 34/10 от 10.12.2010 г., лицензия на рабочее место
	Dev-C++	Свободное ПО, http://www.gnu.org/licenses/gpl.html

6 Материально-техническое обеспечение дисциплины

Учебные аудитории для проведения занятий лекционного типа, семинарского типа, для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Для проведения лабораторных работ используются компьютерный класс (ауд. № 4-113, 4-116, 4-117), оборудованный средствами оргтехники, программным обеспечением, персональными компьютерами, объединенными в сеть с выходом в Интернет.

Аудитории оснащены комплектами ученической мебели, техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой, подключенной к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду Орского гуманитарно-технологического института (филиала) ОГУ (ауд. № 4-307).

Наименование помещения	Материально-техническое обеспечение
Учебные аудитории: - для проведения занятий лекционного типа, семинарского типа, - для групповых и индивидуальных консультаций; - для текущего контроля и промежуточной аттестации	Учебная мебель, классная доска, мультимедийное оборудование (проектор, экран, ноутбук с выходом в сеть «Интернет»)
Компьютерные классы № 4-113, 4-116, 4-117	Учебная мебель, компьютеры (29) с выходом в сеть «Интернет», проектор, экран, лицензионное программное обеспечение
Помещение для самостоятельной работы обучающихся, для курсового проектирования (выполнения курсовых работ)	Учебная мебель, компьютеры (3) с выходом в сеть «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Орского гуманитарно-технологического института (филиала) ОГУ, программное обеспечение

Для проведения занятий лекционного типа используются следующие наборы демонстрационного оборудования и учебно-наглядные пособия:

- презентации к курсу лекций