

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 по выполнению и защите лабораторных и практических работ
по дисциплине «Б1.Д.В.16 Проектирование автоматизированных информационных систем»**

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 бакалавриата

Квалификация

Бакалавр

Форма обучения

Очная

Год начала реализации программы (набора)

2019

г. Орск 2018

Методические указания предназначены для обучающихся очной формы обучения направления подготовки 09.03.01 Информатика и вычислительная техника профилю Программное обеспечение средств вычислительной техники и автоматизированных систем по дисциплине «Б1.Д.Б.16 Проектирование автоматизированных информационных систем»

Составитель _____  О.В. Подсобляева

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения, протокол № 1 от «01» сентября 2018 г.

Заведующий кафедрой _____  Е.Е. Сурина

Согласовано:

Председатель методической комиссии по направлению подготовки 09.03.01 Информатика и вычислительная техника

_____  Е.Е. Сурина

«12» сентября 2018 г.

© Подсобляева О.В., 2018
© Орский гуманитарно-технологический институт (филиал) ОГУ, 2018

Пояснительная записка

В результате изучения дисциплины «Б1.Д.Б.16 Проектирование автоматизированных информационных систем» у обучающихся должны быть сформированы знания, умения и навыки:
сформировать представление о принципах организации и функционирования систем автоматизированного проектирования автоматизированных систем;

- научить методам проектирования автоматизированных систем;
- сформировать представление об основных приемах и способах проектирования технологических процессов с применением современных средств автоматизации.

Целью проведения лабораторных и практических занятий является:

- закрепление знаний студентов по основам проектной деятельности,
- формирование у студентов навыков использования современных технических средств и технологий для решения проектных и исследовательских задач.

Тематический план

Таблица 1 – Тематический план выполнения лабораторных и практических работ по дисциплине «Б1.Д.Б.16 Проектирование автоматизированных информационных систем» для обучающихся направления подготовки 09.03.01 Информатика и вычислительная техника профиль подготовки Программное обеспечение средств вычислительной техники и автоматизированных систем

Лабораторные работы

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	3	Проектирование конкретной (согласно варианта задания) ИС по архитектуре файл-сервер (создание базы данных и приложения обработчика)	12
2	4	Изучение основных возможностей CASE-средств по построению моделей бизнес-процессов и моделей данных.	4
		Итого:	16

Практические работы

№ занятия	№ раздела	Тема	Кол-во часов	
			в 6 и 7 семестре	
1,2	7	Построение диаграмм бизнес-процессов и данных для заданной предметной области (согласно варианта задания)	8	8
3,4	7	Построение диаграмм (вариантов использования, классов и т.д.) при объектно-ориентированном подходе проектирования ИС для заданной предметной области (согласно варианта задания)	8	8
		Итого:	16	16

Методические указания по выполнению и оформлению лабораторных и практических работ

Лабораторные и практические работы по дисциплине «Проектирование автоматизированных информационных систем» предполагают решение задач по темам, представленным в тематическом плане.

В лабораторной и практической работе должны быть выполнены все предусмотренные задания. В работе должна просматриваться логическая последовательность и взаимная увязка

основных частей работы.

Рекомендуемая структура лабораторных и практических работ:

- 1) цель работы;
 - 2) задание в соответствии с выбранным вариантом;
 - 3) теоретическая часть, включающая краткое изложение теоретических положений по теме лабораторной работы, формулы для решения задания;
 - 4) практическая часть, включающая решение задания по теме лабораторной работы.
- Дополнительно для наглядности расчетный материал может быть представлен в виде таблиц, графиков;
- 5) выводы по практической работе;
 - 6) список использованной литературы.

Работы могут быть оформлены:

- машинописным текстом на листах формата А4.

Титульный лист оформляется на основе СТО 02069024. 101 – 2014 «РАБОТЫ СТУДЕНЧЕСКИЕ. Общие требования и правила оформления».

Работа защищается устно и принимается к зачету, если нет замечаний по ее выполнению и оформлению. При отсутствии зачтенных практических работ студент не допускается к зачету по дисциплине «Б1.Д.Б.14 Проектирование автоматизированных информационных систем».

Лабораторная работа №1

Проектирование конкретной (согласно варианту задания) ИС по архитектуре файл-сервер (создание базы данных и приложения обработчика)

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

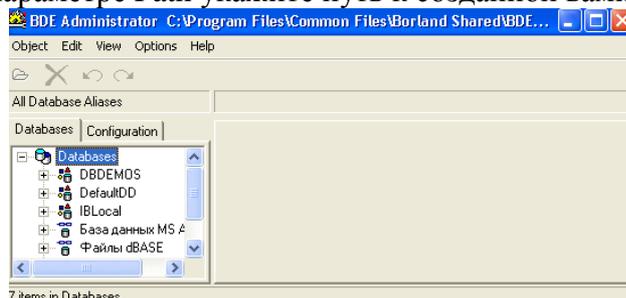


Рисунок 1

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

- создавать таблицы
- изменять структуры
- редактировать записи.

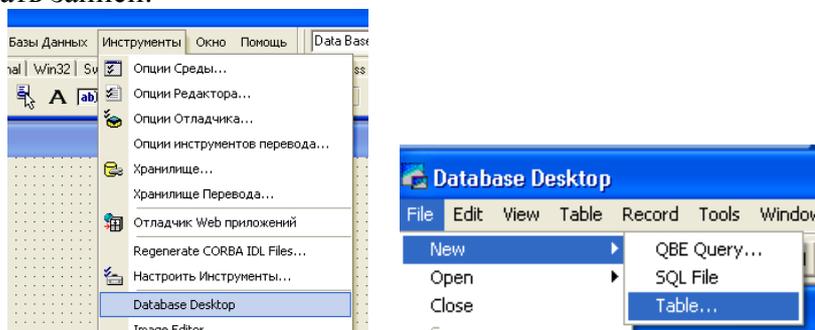


Рисунок 2

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.

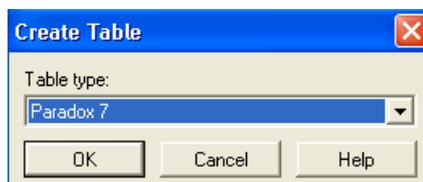


Рисунок 3

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

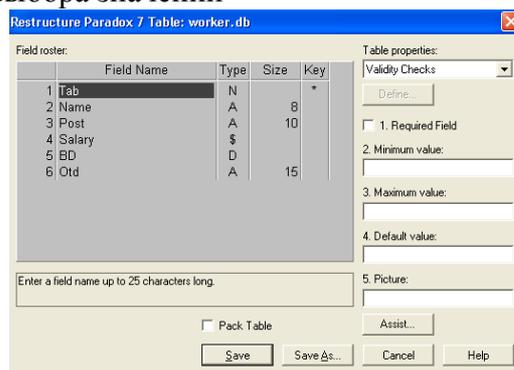


Рисунок 4

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).

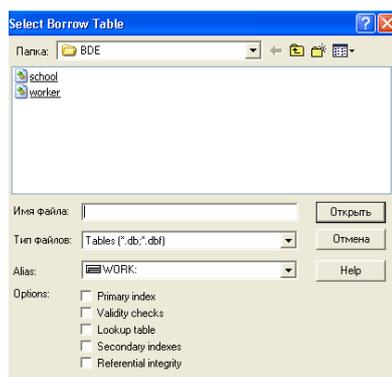


Рисунок 5

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.

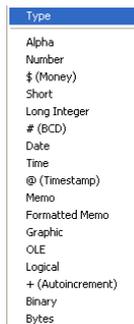


Рисунок 6

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.

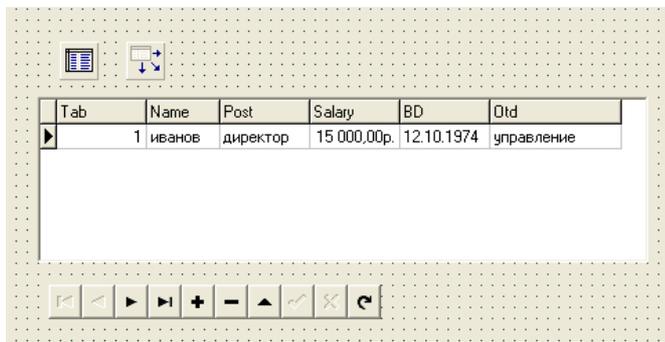


Рисунок 7

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 – компонентом DataSource1 – через свойства DataSource.

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 <code>Datasource</code> для связи результата выполнения запроса (набора записей) с компонентом, обеспечивающим просмотр записей, например <code>DBGrid</code>
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 > 'K') 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.

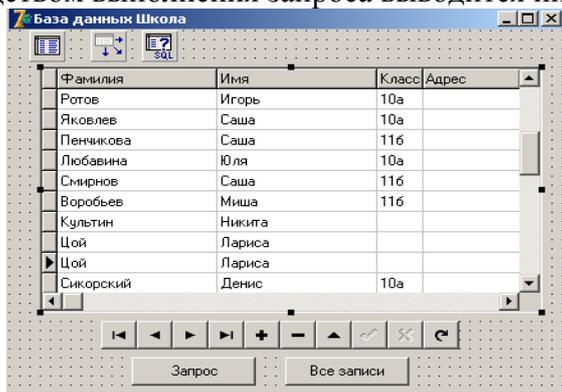


Рис. Форма приложения База данных Школа

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;
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;
```

Лабораторная работа №2

Изучение основных возможностей CASE-средств по построению моделей бизнес-процессов и моделей данных.

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.

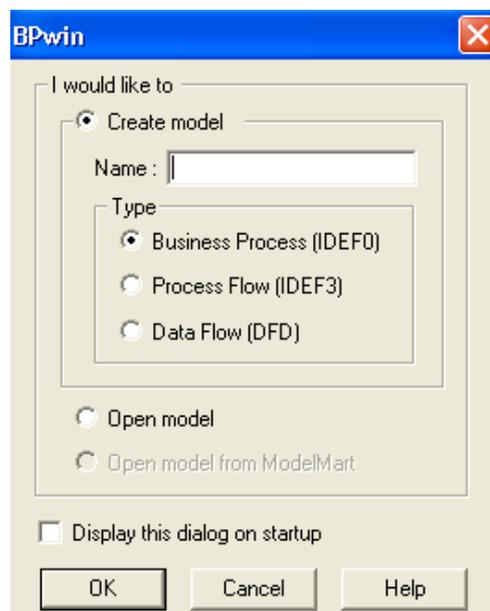


Рисунок 1.

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.

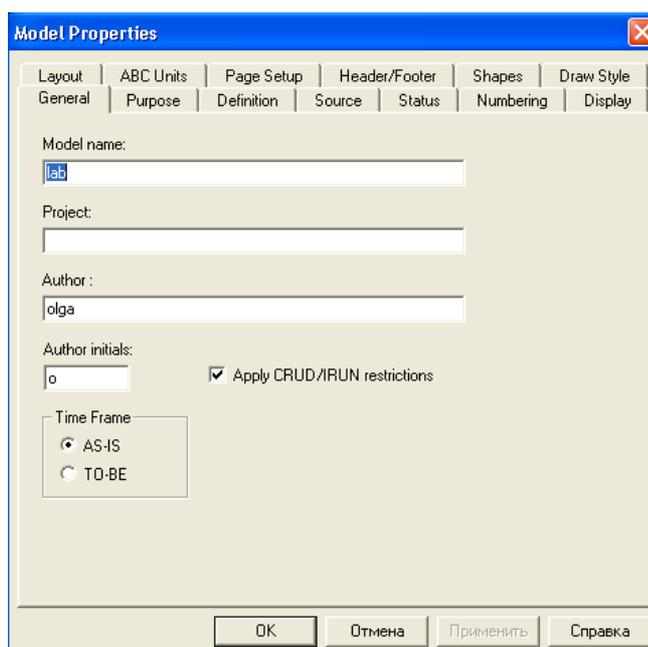


Рисунок 3.

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

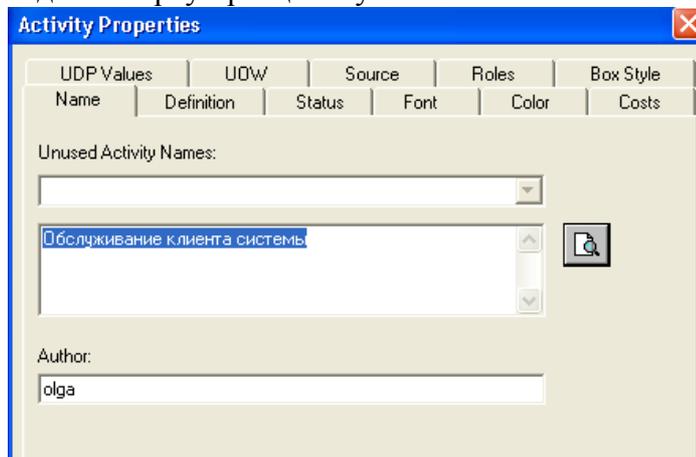


Рисунок 2.

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

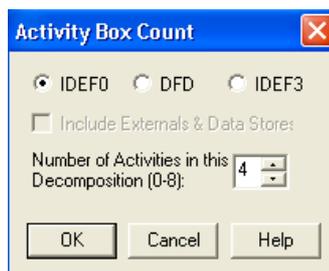


Рисунок 3.

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.

Практическая работа №1,2 **Построение диаграмм бизнес-процессов и данных для заданной предметной области** **(согласно варианта задания)**

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

Практическая работа №3,4

Построение диаграмм (вариантов использования, классов и т.д.) при объектно-ориентированном подходе проектирования ИС для заданной предметной области (согласно варианта задания)

Задание

Предложить для разработки информационную систему (ИС). ИС должна представлять собой программный комплекс, наделенный функциональностью, автоматизирующей конкретную деятельность в рамках предметной области, для которой разрабатывается система. Примером таких систем могут служить:

- автоматизированные системы управления (АСУ)
- электронные магазины, аукционы
- веб-порталы
- сервисы

Что надо сделать?

Составить документ описания требований к разрабатываемой ИС.

1.1 Установление требований

1.1.1 Документ описания требований

Документ, описывающий требования, является осязаемым результатом этапа установления требований. Большинство организаций вырабатывает документ описания требований в соответствии с заранее определенным шаблоном. Шаблон определяет структуру (содержание) и стиль документа.

Ядро документа описания требований состоит из формулировок (изложения) требований. Требования могут быть сгруппированы в виде *формулировок сервисов* (зачастую называемых функциональными требованиями) и *формулировок ограничений*. Формулировки сути сервисов могут быть затем разделены на *требования к функциям* (*function requirements*) и *требования к данным* (*data requirements*). (В литературе термин «функциональные требования» (*functional requirements*) в широком и в узком смысле используется как взаимозаменяемый. При использовании в узком смысле он соответствует тому, что мы называем требованиями к функциям).

Не говоря уже о самих требованиях, документ описания требований должен обращаться к проектным вопросам. Обычно проектные вопросы рассматриваются в начале документа, а затем в конце документа.

Во вводной части документа рассматривается бизнес-контекст проекта, включая цель проекта, участников проекта и основные ограничения. Ближе к заключительной части документа поднимаются другие проектные вопросы, включая план-график выполнения проектных работ, бюджет, риски, документацию и т. д.

1.1.2 Шаблоны документа

Шаблоны для документов описания требований широко доступны. Их можно найти в учебниках, стандартах, выпускаемых такими организациями как ISO, IEEE и т. д., на Web-страницах консалтинговых фирм, программных средствах разработки и т. д. Со временем каждая организация разрабатывает свои собственные стандарты, которые соответствуют принятой в организации практике, корпоративной культуре, кругу читателей, типам разрабатываемых систем и т. д.

Шаблон документа описания требований определяет структуру документа и содержит подробные указания о содержании каждого из разделов документа. Указания могут включать содержание вопросов, мотивацию, примеры и дополнительные соображения.

На рис. 1 показано типичное оглавление документа описания требований. Последующие разделы включают объяснение к приведенному оглавлению.

1.1.3 Предварительные замечания к проекту

Часть документа описания требований, содержащая предварительные замечания к проекту, преимущественно дает ориентиры тем руководителям и участникам проекта, ответственным за принятие решений, которые, вероятно, не станут подробно изучать документ целиком. В начале документа необходимо ясно обозначить цели и рамки проекта, а затем описать деловой контекст системы.

Документ описания требований должен создать прецедент для системы. В частности, необходимо упомянуть обо всех усилиях, приложенных для обоснования необходимости системы на этапе планирования системы. Документ описания требований должен прояснить вопрос о том, каким образом предлагаемая система может способствовать достижению деловых целей и решению задач организацией.

Необходимо обозначить участников проекта системы. Важно, чтобы заказчик выступал не в виде безликого подразделения или офиса — необходимо привести конкретные имена. К концу дня человек должен быть в состоянии решить, приемлемо ли поставляемое программное обеспечение (ПО) для организации.

Особый интерес представляют готовые решения. Всегда неплохо рассмотреть вариант приобретения готового продукта вместо его разработки «с нуля».

Документ описания требований должен предоставлять перечень существующих программных пакетов и компонент, которые должны быть в дальнейшем изучены в качестве вариантов возможных решений. Обратите внимание, что приобретение готового решения изменяет процесс разработки, однако это не избавляет от необходимости проведения анализа требований и проектирования системы!

Наконец, неплохо в заключение раздела предварительных замечаний к проекту документа описания требований привести обзор оставшейся части документа. Это может подтолкнуть к тому, чтобы изучить остальные части документа, а также способствует лучшему пониманию содержания документа. Обзор также может содержать пояснения в отношении методологии анализа проектирования, выбранной разработчиками.

1.1.4 Системные сервисы

Основная часть документа описания требований посвящена определению системных сервисов. Эта часть может занимать до половины всего объема документа. Это также, пожалуй, единственная часть документа, которая может содержать обобщенные модели — модели бизнес-требований.

Рамки системы можно моделировать с помощью диаграммы контекста. В пояснениях к диаграмме контекста должны быть четко определены рамки системы. Без подобного определения проект не может быть застрахован от попыток «растянуть» его рамки.

Функциональные требования можно моделировать с помощью диаграммы бизнес-прецедентов. Однако диаграмма охватывает перечень функциональных требований только в самом общем виде. Все требования необходимо обозначить, классифицировать и определить.

Требования к данным можно моделировать с помощью диаграммы бизнес-классов. Так же, как и в случае функциональных требований, диаграмма бизнес-классов не дает полного определения структур данных для бизнес-процессов. Каждый бизнес-класс требует дальнейших пояснений. Необходимо описать атрибутивное наполнение классов и определить идентифицирующие атрибуты классов. В противном случае невозможно правильно представить

ассоциации.

1.1.5 Системные ограничения

Системные сервисы определяют, что должна делать система. Системные ограничения определяют, насколько система ограничена при выполнении обслуживания. Системные ограничения связаны со следующими видами требований.

- Требования к интерфейсу.
- Требования к производительности.
- Требования к безопасности.
- Эксплуатационные требования.
- Политические и юридические требования.

Требования к интерфейсу определяют, как система взаимодействует с пользователями. В документе описания требований определяется только «впечатление и ощущение» от GUI-интерфейса.

Начальное проектирование (закрашивание экрана) GUI-интерфейса проводится во время спецификации требований и позже во время системного проектирования.

В зависимости от области приложения требования к производительности могут играть довольно значительную роль в успехе проекта. В узком смысле они задают скорость (время отклика системы), с которой должны выполняться различные задания. В широком смысле, требования к производительности включают другие ограничения в отношении надежности, готовности, пропускной способности и т. д.

Требования к безопасности описывают пользовательские права доступа к информации, контролируемые системой. Пользователям может быть предоставлен ограниченный доступ к данным или ограниченные права на выполнение определенных операций с данными.

Эксплуатационные требования определяют программно-техническую среду, если она известна на этапе проектирования, в которой должна функционировать система. Эти требования могут оказывать влияние на другие стороны проекта, такие как подготовка пользователей и сопровождение системы.

Политические требования и юридические требования скорее подразумеваются, чем явно формулируются в документе описания требований. Подобная ошибка может обойтись очень дорого. Пока эти требования не выведены явно, программный продукт может быть трудно развернуть по политическим или юридическим причинам.

Возможны и другие виды ограничений. Например, в отношении некоторых систем могут предъявляться повышенные требования к легкости их использования (требования в отношении пригодности к использованию) или легкости их сопровождения (требования в отношении пригодности к сопровождению).

Значение выработки недвусмысленных определений для системных ограничений трудно переоценить. Существует немало примеров проектов, которые провалились из-за упущенных или неверно понятых ограничений. Эта проблема в равной мере относится как к заказчикам, так и к разработчикам. Недобросовестные или нерассудительные разработчики могут разыграть «карту системных ограничений», чтобы получить преимущество в своем стремлении уклониться от ответственности.

1.1.6 Проектные вопросы

Заключительная часть документа описания требований обращается к другим проектным вопросам. Один из важных разделов этой части называется «Открытые вопросы».

Здесь поднимаются все вопросы, которые могут сказаться на успехе проекта и которые не рассматривались в других разделах документа. Сюда относится ожидаемое возрастание значения

некоторых требований, которые в текущий момент выходят за рамки проекта. Сюда можно отнести также любые потенциальные проблемы и отклонения в поведении системы, которые могут начаться в связи с развертыванием системы.

В этой же части необходимо представить предварительный план- график выполнения основных проектных заданий. Сюда же относится предварительное распределение людских и других ресурсов. Для выработки стандартных плановых графиков можно использовать программные средства управления проектами, например, такие как система PERT (program evaluation_and_review technique — метод оценки и пересмотра планов) или карты Ганта.

Прямым результатом составления план-графика может быть разработка предварительного бюджета. Стоимость проекта может быть выражена скорее в виде диапазона значений затрат, а не конкретного значения. При наличии надлежащим образом документированных требований для оценки затрат можно использовать один из подходящих методов.

1.1.7 Приложения

Приложения содержат остальную, полезную для понимания требований, информацию. Основным добавлением здесь служит глоссарий. Глоссарий определяет термины, сокращения и аббревиатуры, используемые в документе описания требований. Значение толкового глоссария трудно переоценить. Неверное истолкование терминологии таит в себе большую опасность для проекта.

Одна из особенностей, которую часто упускают из виду при составлении документа описания требований, состоит в том, что в проблемной области, определяемой документом, можно довольно неплохо разобраться с помощью изучения документов и форм, используемых в процессах делопроизводства. При возможности следует включать в документ заполненные формы — «пустые» формы не дают такого же уровня понимания бизнес-процессов.

Раздел ссылок содержит перечень документов, которые упоминаются или используются при подготовке документа описания требований. К ним могут относиться книги и другие опубликованные источники информации, но — что, пожалуй, даже более важно — необходимо также упомянуть протоколы совещаний, служебные записки и внутренние документы.

Пример документа описания требований

Документ описания требований ИС «Домашняя бухгалтерия»

1. Предварительные замечания к проекту

1.1. Цели и рамки проекта

Целью данного проекта является разработка информационной системы для ведения и оптимизации семейного бюджета. ИС «Домашняя бухгалтерия» должна быть проста в использовании и не требовать от пользователя знаний бухгалтерского учета.

1.2. Деловой контекст

Многие семьи в наше время планируют семейный бюджет. Ведение семейного бюджета при помощи подручных средств – карандаш, бумага – не всегда удобно и всегда трудоемко. Использование для этих целей компьютерных программ для ведения бухгалтерии не оправдано с точки зрения сложности их освоения и избыточного функционала для ведения домашней бухгалтерии. В связи с этим возникает необходимость создания специализированной программы ведения домашней бухгалтерии.

1.3. Участники проекта

Заказчик – Васильева Марья Федоровна (m.vasileva@mypochta.ru)

Разработчик – Петров Степан Николаевич (petrov@coolsoft.com)

1.4. Идеи в отношении решений

Программа должна быть реализована в виде настольного приложения для операционных

систем семейств MS Windows.

1.5. Обзор документа

В разделе «Системные сервисы» описывается, что должна делать система. В разделе «Системные ограничения» определяется, насколько система ограничена при выполнении обслуживания.

В разделе «Проектные вопросы» освещаются прочие проектные вопросы.

2. Системные сервисы

2.1. Рамки системы

Рамки системы можно моделировать с помощью диаграммы контекста.

ИС «Домашняя Бухгалтерия» получает данные о доходах и расходах от внешней сущности «Домохозяин». Для передачи этих данных сущности «Домохозяин» должен авторизоваться. В своей работе сущность «Домашняя Бухгалтерия» использует информацию о ценах на товары и тарифах и курсах валют, получаемую от внешних сущностей «База тарифов и цен на товары» и «База курса валют». Результаты своей работы ИС «Домашняя Бухгалтерия» может отображать как внешней сущности «Домохозяин», так и генерировать в виде отчетов формата MS Excel для внешней сущности «MS Excel».

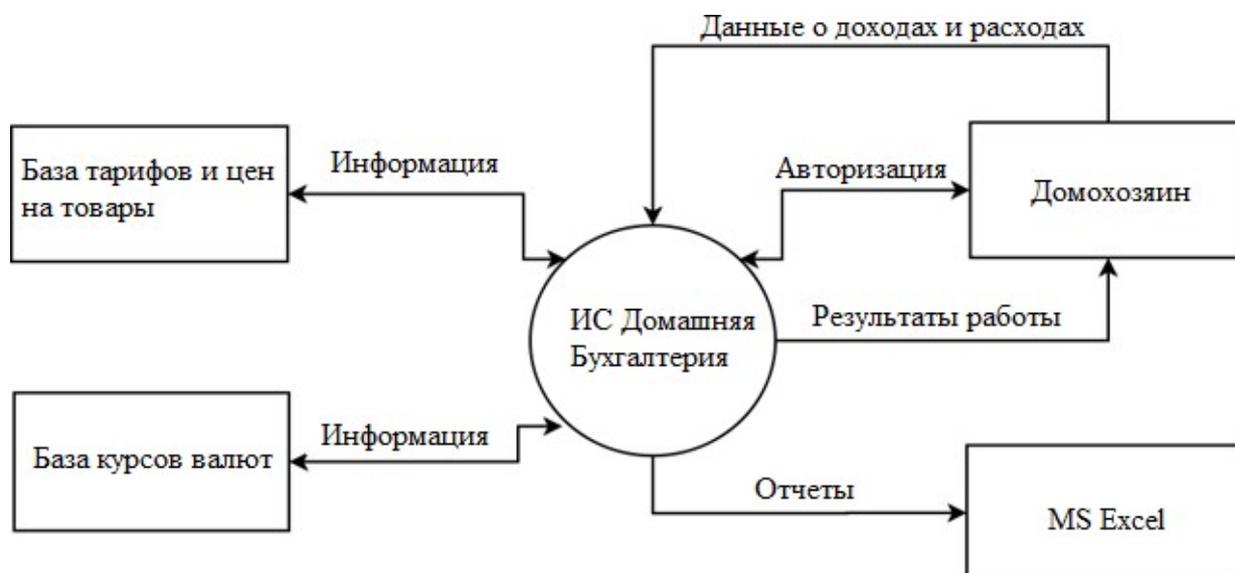


Рис.1 Контекстная диаграмма ИС «Домашняя Бухгалтерия»

2.2. Функциональные требования

ИС должна обеспечивать следующие функциональные возможности:

- учет расходов;
- учет доходов;
- учет денег, отданных и взятых в долг;
- погашение долгов частями;
- проценты по долгам;
- контроль возврата долгов;
- система напоминания по долгам;
- составление бюджета расходов и доходов;
- планирование расходов;

- планирование доходов;
- система счетов;
- возможность использовать до пяти валют включительно;
- получение курсов валют из интернет;
- обмен валют;
- импорт данных из файлов Microsoft Excel;
- поиск по базе данных;
- фильтры и быстрый поиск по базе данных;
- экспорт данных в Excel, XML, текстовый файл;
- перенос данных;
- резервное копирование;
- печать данных;
- построение отчетов и диаграмм;
- настройка пользовательского интерфейса.

2.3. Требования к данным

ИС должна хранить свои данные в специализированных XML-файлах.

3. Системные ограничения

3.1. Требования к интерфейсу

ИС должна иметь стандартный интерфейс приложений, разработанных для ОС MS Windows.

3.2. Требования к производительности

Особых требований к производительности ИС нет.

3.3. Требования к безопасности

С программой могут работать несколько человек, входя в программу под своими именами. Для обеспечения конфиденциальности каждое имя можно защитить паролем. Добавление, изменение и удаление пользователей осуществляется в администраторе пользователей.

3.4. Эксплуатационные требования

ИС должна функционировать на ОС Windows XP, ОС Windows Vista, ОС Windows 7. Минимальные аппаратные требования определяются минимальными аппаратными требованиями к выше-перечисленным ОС.

3.5. Политические и юридические требования Нет.

3.6. Другие ограничения Нет.

1.1.8 Варианты заданий

1. АСУ деятельностью отдела кадров предприятия
2. АСУ складского хранения
3. АСУ деятельностью библиотеки
4. Веб-магазин по продаже часов
5. Веб-магазин по продаже фотоаппаратов
6. АСУ деятельностью аптечной сети
7. Веб-сайт букмекерской конторы
8. ИС учета успеваемости студентов
9. Веб-магазин по продаже компьютерных комплектующих

10. Программный RSS-агрегатор
11. Веб RSS-агрегатор
12. ИС «Ежедневник»
13. АСУ деятельностью магазина видеопроката
14. АСУ деятельностью автосалона
15. Веб-магазин по продаже одежды
16. ИС «Почтовый коллектор»
17. АСУ деятельностью магазина бензозаправки
18. АСУ учетом пациентов в поликлинике
19. АСУ учетом коммунальных платежей
20. АСУ деятельностью службы такси
21. ИС сбора и обработки ошибок (багтрекер)
22. Веб-сайт кафедры
23. Веб-сайт факультета
24. ИС хранения и каталогизации фотографий
25. ИС «Каталог недвижимости»

Рекомендуемая литература

Основная литература

1. Антонов, В.Ф. Методы и средства проектирования информационных систем : учебное пособие / В.Ф. Антонов, А.А. Москвитин ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». – Ставрополь : СКФУ, 2016. – 342 с. : ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=458663> коэффициент книгообеспеченности 1
2. 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. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=208706> (дата обращения: 28.12.2019). – ISBN 978-5-4332-0083-8. коэффициент книгообеспеченности 1
3. Ипатова, Э.Р. Методологии и технологии системного проектирования информационных систем : учебник / Э.Р. Ипатова, Ю.В. Ипатов. – 2-е изд., стер. – Москва : Флинта, 2016. – 257 с. : табл., схем. – (Информационные технологии). – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=79551>. – Библиогр.: с. 95-96. – ISBN 978-5-89349-978-0. коэффициент книгообеспеченности 1
4. Киселева, Т.В. Проектирование информационных систем: курс лекций : [16+] / авт.-сост. Т.В. Киселева ; Министерство науки и высшего образования Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». – Ставрополь : СКФУ, 2018. – Ч. Часть 1. – 150 с. : ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=563326> коэффициент книгообеспеченности 1

Дополнительная литература

1. 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: <http://biblioclub.ru/index.php?page=book&id=141626>, коэффициент книгообеспеченности 1

2. Исаев, Г.Н. 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

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

4. 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 1

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»