

# Системы управления базами данных

- Создание документов по хранимой в БД информации (л.р.6)
- Сценарий и интерфейс программы (л.работа №7)
- Содержание расчетного задания

## Методология разработки проектов на основе СУБД

- Выбор тематики разработки
- Работа с заказчиком
- Этапы проектирования БД

# Тема: документирование информации из БД

2

**Отчеты (Reports)** – формируют единый документ по данным со всех записей одной или нескольких таблиц или представлений данных

**Этикетки (Labels)** – формируют документы индивидуально для каждой записи главной (родительской, ведущей) таблицы или представления. Для формирования документа могут использоваться данные из связанных с родительской дочерних таблиц (представлений)

Структура и технология разработки отчетов и этикеток полностью одинакова.

Отчет (этикетка) состоит из отдельных разделов, представляемых в средах разработки виде отдельных полос (bands)

## **Виды разделов (полос)**

Основная полоса – detail. В нее выводится информация по каждой записи главной таблицы. Изначально формирование документа начинается с первой доступной записи. Таких полос может быть несколько. Эта полоса реплицируется в отчете для каждой записи!

Дополнительные полосы – колонтитулы страниц: верхний (page header) и нижний (page footer)

Дополнительно для отчетов могут создаваться такие разделы, как титул или заголовок отчета (title), и раздел общих итогов по документу (summary).

В любом документе возможно создание дополнительных разделов, рассматриваемых далее

# Структура отчета (этикетки) (среда FoxPro)

**Report Designer - q\_all.frx**

0 1 2 3 4 5 6 7

**Ведомость остатков электронных компонентов** DATEQ

△ Title

Группа Наименование Номинал Кол-во: Лежит В работе Исп-но Куплено Ед.изм

△ Page Header

ALLT(Q\_is\_detal.typeName)

△ Group Header 1: typeName

name Transf(Q\_is\_det Transf(Q\_is\_det Transf(Q\_is\_det Transf(Q\_is\_det id\_name

△ Group Header 2: name

Allt(Q\_is\_detal.nominal) Transf(Q\_is\_det Transf(Q\_is\_det Transf(Q\_is\_det Transf(Q\_is\_det id\_name

▲ Detail

△ Group Footer 2: name

[Итого по группе "] + ALLT(Q\_is\_detal.t q\_s q\_w q\_u -Q\_is\_detal.q

△ Group Footer 1: typeName

"Смп." + TRANSFORM(PAQ

△ Page Footer

[Всего компонентов:] q\_s

△ Summary

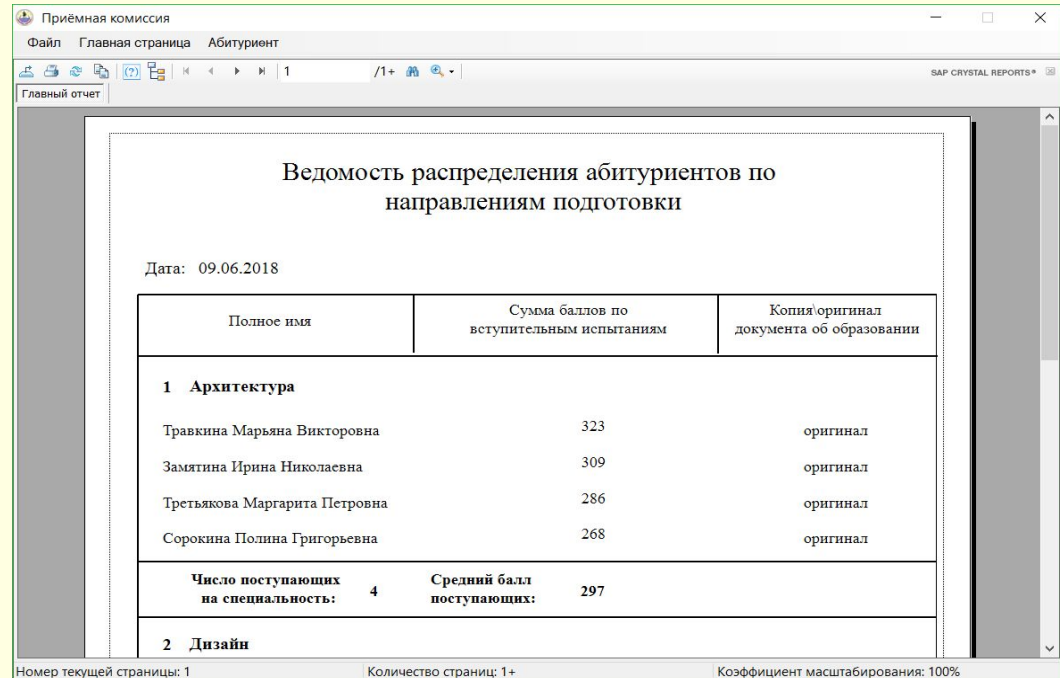
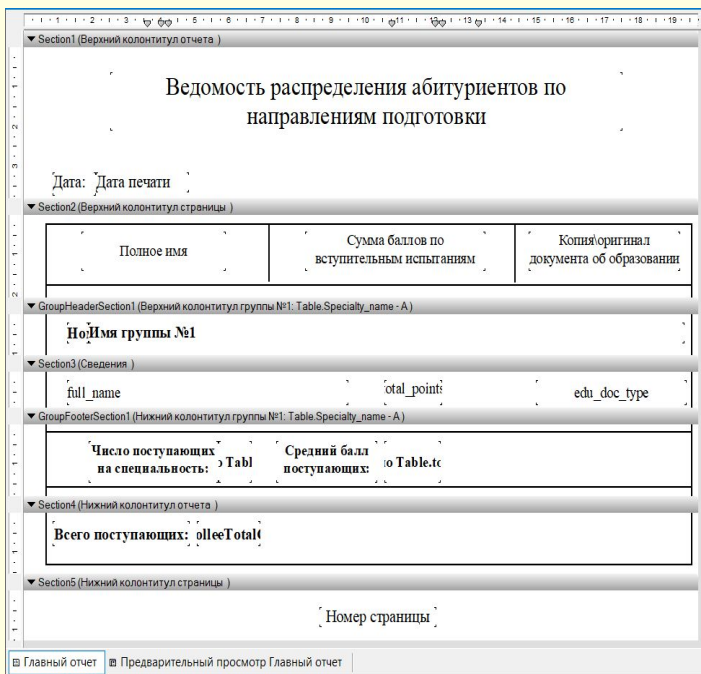
# Среды для разработки отчетов: dbForge

The screenshot displays the dbForge Report Designer interface. On the left, the 'Report1.rdb' project tree shows a hierarchical structure of report bands and fields. The main workspace shows a report design for 'Учет деятельности строительных фирм' (Accounting of construction companies) dated '9 июня 2018 г.'. The design includes various header bands (groupHeaderBand1 to groupHeaderBand6) and a detail band (Detail) containing a table with columns for 'Используется', 'В количестве', 'По цене за', and 'Стоимость'. Below the workspace is the 'Group and Sort' panel, which allows configuring the sort order and visibility of fields. The right side of the image shows the 'Report1.rdb' preview window, displaying the rendered report with the same title and date. The preview shows a table with columns for 'Номер работы', 'Название работы', 'Название исполняющей фирмы', 'Дата начала работ', 'Плановая дата завершения работ', and 'Дата окончания работ'. The table contains data for two construction jobs: 'Постройка бани' and 'Проводка'.

Номер работы	Название работы	Название исполняющей фирмы	Дата начала работ	Плановая дата завершения работ	Дата окончания работ
1	Постройка бани	ОАО Киндилат от народа!	2018-03-15	2018-04-10	2018-04-09
Адрес работ ул.Измайловская, 25					
Используется:	Деревянный брус	В количестве:	40	По цене за штуку:	700
Стоимость	28 000,00 Р				
Используется:	Кирпич	В количестве:	3000	По цене за штуку:	120
Стоимость	360 000,00 Р				
Используется:	Покраска фасада	В количестве:	2	По цене за штуку:	1000
Стоимость	2 000,00 Р				
Используется:	Укладка деревянного пола	В количестве:	3	По цене за штуку:	900
Стоимость	2 700,00 Р				
Используется:	Электропроводка	В количестве:	2	По цене за штуку:	500
Стоимость	1 000,00 Р				
2	Проводка				

dbForge – отдельно устанавливаемое проприетарное приложение стоимостью около \$80. Бесплатная пробная версия дается на 30 дней. Функционал по созданию отчетных форм весьма ограничен. Например, не все части отчета можно нумеровать. Это минусы. Достоинство – простота применения. Сначала нужно подключить dbForge к MySQL server. Далее, воспользовавшись конструктором отчетов, нужно создать отчет. Создание отчета происходит в два этапа: сначала пишется SQL – команда на выбор данных из указываемых таблиц. Затем выбранные поля вручную группируются так, как это необходимо. Нужно также выбрать макет будущего отчета (из нескольких имеющихся вариантов). Далее появится окно конструктора отчета для внесения дополнительных изменений. Отчет формируется практически автоматически.

# Среды для разработки отчетов: Crystal Reports



Расширение Crystal Reports для Visual Studio. Данное приложение может интегрироваться с любыми средами разработки приложений, не только C#

# Альтернативные варианты

1. Самое простое решение – создать HTML - страницу отчета, скопировать ее и вставить из буфера в MS Word
2. Воспользоваться готовыми конвертерами страницы в pdf – файл (<https://habr.com/ru/post/134505/>)  
На данном ресурсе даются ссылки на **wkhtmltopdf** (<http://code.google.com/p/wkhtmltopdf/>), **htmldoc** (<http://www.htmldoc.org/>), **html2ps**, **ps2pdf**.
3. Написать собственный код для конвертирования данных в pdf – файл  
(Динамическое создание PDF-файлов при помощи PHP - <https://www.ibm.com/developerworks/ru/library/x-buildpdfphp/index.html>)

# Содержание лабораторной работы № 6<sup>7</sup>

**Создать один отчет (Report) по теме своего индивидуального задания, который должен удовлетворять следующим требованиям:**

- наличие заголовка и даты документа
- наличие структурированных данных как минимум одного уровня вложения с наличием не менее чем по 2-3 единицы информации на каждом уровне (например: не менее двух факультетов, на каждом из которых имеется не менее двух специальностей);
- на верхнем и следующим за ним структурных уровнях с учебной целью предусмотреть нумерацию пунктов;
- наличие промежуточных итогов для внутреннего уровня отчета
- наличие общих итогов по всему отчету;
- наличие нумерации страниц (общее число страниц в отчете должно быть не менее двух);

**Создать одну этикетку (Label) по теме своего индивидуального задания, основным требованием к которой является разумность содержания и корректность оформления**

# Пример оформления отчета

Report Designer - q\_all.frx - Page 1

<b>Ведомость остатков электронных компонентов</b>							14.03.12
Группа	Наименование	Номинал	Кол-во:	Лежит	В работе	Исп-но	Куплено Ед.изм
<b>Резисторы подстроечные</b>							
	5.60 Ом		5.00	0.00	0.00	0.00	5.00 Ом
	4.70 кОм		1.00	0.00	0.00	0.00	1.00 кОм
Итого по группе "Резисторы подстроечные":				6	0	0	6
<b>Транзисторы</b>							
	1Т806 Б		2.00	0.00	0.00	0.00	2.00
	2Т830 Б		18.00	0.00	0.00	0.00	18.00
	2Т831 Г		17.00	0.00	0.00	0.00	17.00
	ГТ403 Ж		1.00	0.00	0.00	0.00	1.00
	КТ203 АМ		102.00	0.00	0.00	0.00	102.00
	КТ802 А		2.00	0.00	0.00	0.00	2.00
	КТ902 А		2.00	0.00	0.00	0.00	2.00
	П702 А		2.00	0.00	0.00	0.00	2.00
Итого по группе "Транзисторы":				146	0	0	146
<b>Транзисторы полевые</b>							
	КП103 Ж		1.00	0.00	0.00	0.00	1.00
	КП103 К		1.00	0.00	0.00	0.00	1.00
	КП103 Л		1.00	0.00	0.00	0.00	1.00
	КП103 М		1.00	0.00	0.00	0.00	1.00

В отличие от задания, в примере нет нумерации пунктов и не показаны нумерация страниц и итоговые данные по всему отчету



# Примеры оформления этикетки

Label Designer - model.lbx - Page 1

Модель : юпитер5  
 Объем двигателя : 235  
 Макс. скорость : 150  
 Мощность : 24  
 Тип КПП : мех  
 Производитель : Россия  
 Передача на колесо : цепь  
 Тип колес : R-18

Модель : планета5  
 Объем двигателя : 235  
 Макс. скорость : 150  
 Мощность : 24  
 Тип КПП : мех  
 Производитель : Россия  
 Передача на колесо : цепь  
 Тип колес : R-18

Улица : Юрина 208  
 Ленинский район  
 9 этажный дом  
 Дата постройки : 07.07.87

Улица : Юрина 206  
 Ленинский район  
 5 этажный дом  
 Дата постройки : 08.06.89

Улица : Попова 145  
 Ленинский район  
 9 этажный дом  
 Дата постройки : 09.08.80

Улица : А. Петрова 200  
 Октябрьский район  
 5 этажный дом  
 Дата постройки : 07.07.70

Улица : А. Петрова 250

Улица : Юрина 7

Print Preview

100%

# Тема: сценарий и интерфейс приложения

**Интерфейс пользователя** - это набор меню, экранных форм, панелей инструментов и иных средств, с помощью которых пользователь управляет работой программы, ведет диалог, вводит данные и получает необходимую информацию.

**Сценарий** – это возможные последовательности операций взаимодействия пользователя посредством интерфейса с компьютером с целью получения необходимых результатов.

Следует помнить, что сценарий и интерфейс имеют первоочередное значение в оценке программного продукта. Если пользователь теряется в способе реализации требуемого результата, предлагаемый программный продукт будет забракован

## Общий подход к разработке:

При разработке приложения крайне важно **учесть существующий интерфейс** уже используемых заказчиком продуктов для решения аналогичных (или точно таких же) задач. Самое сложное и проблематичное в силу субъективных причин – когда новое приложение использует новые парадигмы взаимодействия пользователя с программой (например, при смене ОС, переходе от одной офисной программы к другой)

Нужно **стремиться к максимально простой, понятной и логичной структуре** интерфейса, когда пользователю не потребуется изучать большие объемы справочной информации с большим числом переходов по справочной системе для получения желаемой информации. Интерфейс должен быть интуитивно понятен. Для этого следует максимально использовать интерфейсные решения на уровне Wizard, когда каждый шаг пользователя комментируется разработчиком

# Правила создания интерфейса приложения

1. **Минимум новизны.** Оптимально то, что привычно и не требует дополнительного обучения. Используйте горячие клавиши и интерфейсные решения наиболее популярных, или ранее используемых пользователем программных продуктов.
2. **Минимум пестроты.** Избегайте ярких цветов, динамических объектов, обилия расцветок и шрифтов
3. **Максимум простоты.**
  - **Экранные формы.** Не перегружать экранную форму избытием элементов управления (максимум – от 10 до 20). Минимизировать или исключить число динамически изменяемых объектов. Разбивать одну сложную форму на несколько простых (или на несколько закладок), систематизировав информацию на основную и второстепенную по категориям.
  - **Меню.** Избегать длинных названий, большого количества пунктов и многоуровневых вложений пунктов (лучше – не более трех). Как можно меньше пунктов должны приводить к непосредственному результату без дополнительного диалога, если пункт модифицирует хранимую информацию. Избегать применения динамического меню, ограничившись запретом доступа, а не удалением пункта. Использовать меню для вызова типовых универсальных операций.
4. **Максимум контекстно-зависимой помощи** – «всплывающих» подсказок, развернутого комментария в строке статуса. Все требуемые от пользователя действия должны быть ему интуитивно понятны и очевидны, а все подсказки – максимально конкретны.

# Правила создания интерфейса приложения

5. **Минимум манипуляций.** Для получения результата пользователю желательно обходиться максимум тремя переходами между меню и окнами (**правило трех кликов**). Это особенно важно для часто выполняемых операций, которые следует **дублировать** на панелях инструментов и горячими клавишами. Желательно, чтобы ПО помнило настройки пользователя, предлагало возможность их записи в журнал или в файл. Пункты меню, вызываемые по горячим клавишам, не обязательно выносить на передний план. Не стоит использовать задействованные системой клавиши CTRL для вызова специфических функций (например, печать – CTRL+P, закрытие окна – CTRL+F4 – для этого есть клавиша ALT).
6. **Побольше логичности.** Необходимо четко структурировать систему меню. Оптимально, когда, в одно вертикальное меню собраны пункты общего назначения или относящиеся к одному и тому же объекту. Лучше дублировать действия в разных пунктах, чем заставлять пользователя искать их. Заголовки должны быть емкими, конкретными и лаконичными.
7. **Побольше информативности.** Не нужно скупиться на введение в продукт всевозможных дополнительных сообщений и индикаторов состояния программы, в частности, объектов типа Progress Bar (“Градусник”). Это особенно важно, когда система занята длительными вычислениями. Желательно сопровождать программу полноценной документацией как в «твердом» виде, так и в виде встроенного набора подсказок, пояснений и описаний. Чем лучше разработан интерфейс программного продукта, тем реже пользователю приходится обращаться к такой системе, однако для массы программ это не характерно.

# Возможная структура меню приложения

## Тема: регистратура поликлиники

- |  |   |   |
|--|---|---|
| <ol style="list-style-type: none"> <li>1. <b>Файл:</b> <ul style="list-style-type: none"> <li>• Настройки</li> <li>• -----</li> <li>• Печать отчета</li> <li>• Печать этикетки</li> <li>• -----</li> <li>• Выход</li> </ul> </li> <li>2. <b>Текст</b> <ul style="list-style-type: none"> <li>• Копировать</li> <li>• Вырезать</li> <li>• Вставить</li> </ul> </li> <li>3. <b>Запись:</b> <ul style="list-style-type: none"> <li>• Добавить</li> <li>• Удалить</li> <li>• Редактировать</li> <li>• Дублировать</li> <li>• -----</li> <li>• Фильтр вкл./выкл.</li> <li>• Найти</li> <li>• -----</li> <li>• Подсчет строк</li> <li>• Суммы</li> </ul> </li> </ol> | <p>Слева – типовые действия, справа – типовые только названия групп, а сами пункты в группе – специфические для конкретной предметной области</p> | <ol style="list-style-type: none"> <li>4. <b>Справочники+данные:</b> <ul style="list-style-type: none"> <li>• Пациенты</li> <li>• Врачи</li> <li>• Специальности</li> <li>• Предприятия</li> <li>• Участки</li> <li>• -----</li> <li>• Расписания</li> <li>• Посещения</li> </ul> </li> <li>5. <b>Действия</b> <ul style="list-style-type: none"> <li>• Записаться к врачу</li> <li>• Посмотреть анализы</li> <li>• Получить консультацию</li> </ul> </li> <li>6. <b>Анализ</b> <ul style="list-style-type: none"> <li>• Посещаемость</li> <li>• Заболевания</li> <li>• Контингент</li> <li>• Больничные</li> </ul> </li> <li>7. <b>Помощь</b> <ul style="list-style-type: none"> <li>• Руководство</li> <li>• О программе</li> </ul> </li> </ol> |
|--|---|---|

# Работа № 7: разработка пользовательского интерфейса

1. Продумать интерфейс пользователя для реализации всего функционала создаваемого приложения, включающий набор экранных форм, меню, панель инструментов, дерево реализующих функционал операций
2. Создать эскизы для всех типов экранных форм интерфейса. Для всех однотипных форм (в основном – по справочной информации) достаточно привести один эскиз.
3. Продумать систему меню, из которой могут вызываться данные формы
4. Подготовить список документов, который может создавать приложение (с применением разработанных экранных форм для задания их параметров)
5. Создать главную форму будущего приложения, содержащую необходимые элементы управления и таблицу с отображаемыми данными. Если она содержит много закладок – проработать из них одну – две основных.
6. Реализовать для основных экранных форм (главной и, возможно, вспомогательной) запросы по добавлению и удалению записей из БД (из таблицы, содержащей не справочную информацию, а основные операции при работе с приложением). Сменная часть запросов задается переменными.

**Примечание:** эскизы форм можно создавать в любом графическом редакторе, включая средства MS Word или Paint Brush.

# Содержание расчетного задания

**Задача:** Создать приложение, запускаемое из экранной формы, включающее меню и обеспечивающее реализацию главной функции приложения, в том числе возможность ввода, удаления и редактирования соответствующих основному объекту приложения записей, их отбор по группе критериев с подсчетом результата и просмотр отчета (отчет – по возможности)

1. Создать для приложения главную форму, с помощью которой можно бы было просматривать списки, содержащие основные объекты или операции в соответствующей предметной области.
2. Предусмотреть возможность отбирать в списках записи, удовлетворяющие некоторому набору условий (значений полей), свойств или требований, количество которых должно быть **не менее двух**. Выбор **необходимых** для отбора условий (одно любое по выбору или одновременно несколько из имеющихся в любой комбинации) определяется пользователем. На форму нужно вносить все условия (даже не реализованные в программе).
3. На форме должна присутствовать информация об общем (отобранном) числе содержащихся в списке записей. Иные результаты анализа списка (суммы, средние значения и прочее) – по необходимости, согласовав с преподавателем
4. Приложение должно иметь панель инструментов, хотя бы одна кнопка которой должна быть синхронизирована с пунктом основного меню и кнопкой главной формы
5. Остальное содержание задания (необходимость включения дополнительных функций и форм) определяется исполнителем самостоятельно
6. Примечание: не претендующие на высокий рейтинг могут меню, панель инструментов и отчет не делать

# Требования к функционалу приложения

Расчетное задание должно обеспечивать работу приложения в соответствии с выбранной тематикой в его основной части, а именно:

- добавлять, удалять и редактировать информацию, **касающуюся основного функционала** приложения, обеспечивая при этом **поддержание целостности** базы данных. Не должно в процессе добавления или изменения данных появляться дублирующей (или полностью идентичной) или противоречивой информации! Работу с таблицами, содержащими справочную, вспомогательную информацию – не реализовывать;
- выводить в окно предварительного просмотра документ – отчет, содержащий данные, относящиеся к основной информации и имеющий как минимум один уровень группировки с подведением итогов по созданным группам и по отчету в целом;
- управлять работой приложения в части реализуемых функций посредством меню, комбинации горячих клавиш, элементов управления на экранной форме и элементов управления на панели инструментов (на панели инструментов реализовать только основные функции: удаление, вставка и редактирование, но можно ограничиться работой и одной кнопки). При этом нужно обеспечить синхронную работу элементов интерфейса пользователя.


## Примечания:

1. При реализации функций особое внимание уделить наглядности интерфейса, чтобы не требовалось прибегать к применению контекстно-зависимой помощи
2. При редактировании/добавлении/удалении записей следить за соблюдением всех видов целостности данных, информируя пользователя о допущенных им ошибках, а также не забывать при добавлении и удалении записи корректировать информацию о числе отображаемых записей
3. Реализацию перечисленных функций осуществить с применением минимального набора экранных диалоговых форм. Обычно для этого достаточно разработать одну-три формы и одну панель инструментов. **Формы для реализации должны быть обязательно согласованы с преподавателем!**
4. Панель инструментов в виде отдельной формы не делается при использовании web - интерфейса



# Пример 1: реализация с одной формой

Предпродажная подготовка

Склад | Модели | Персонал | Отчет | **ООО "Мото Мир +"** | 

Этикетки

**Фильтр**

Модель	Цвет	Готов
<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Не учитывать <input type="checkbox"/> Если ДА

Фильтровать | Сброс фильтра

Код тов	Модель	№Двигателя	№Рамы	Цвет	Дата приема	Готов
▶ 1	юпитер 5	1259100236	4561283	красный	22.02.08	<input checked="" type="checkbox"/>
2	планета 5	8723847378438	343434343	красный	22.02.08	<input type="checkbox"/>
3	юпитер 5	212313213232	232323232	красный	22.02.08	<input checked="" type="checkbox"/>
4	юпитер 6	12314567	125148	зеленый	22.02.08	<input checked="" type="checkbox"/>
5	юпитер 5	12121212	121212121	красный	22.12.08	<input checked="" type="checkbox"/>
6	CRX	4343434343434	4343554r4545454	желтый	22.12.08	<input checked="" type="checkbox"/>
7	юпитер 6	54465786545	545456+45	синий	22.12.08	<input type="checkbox"/>
10	юпитер 5	1249576565	2342343434343	синий	22.12.08	<input checked="" type="checkbox"/>
11	планета 5	232323232323232	343546554655676	синий	22.12.08	<input checked="" type="checkbox"/>

**Сведения о подготовке**

№	Расконсерв	Протяжка	Масло	Двигатель	Контр.пуск	Дата	Примечания	Проверил
▶ 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	...	примечаний нет	2

**Подготовку провел**

Фамилия	Имя	Отчество	Должность
▶ Петров	Петр	Николаевич	техник

Выход

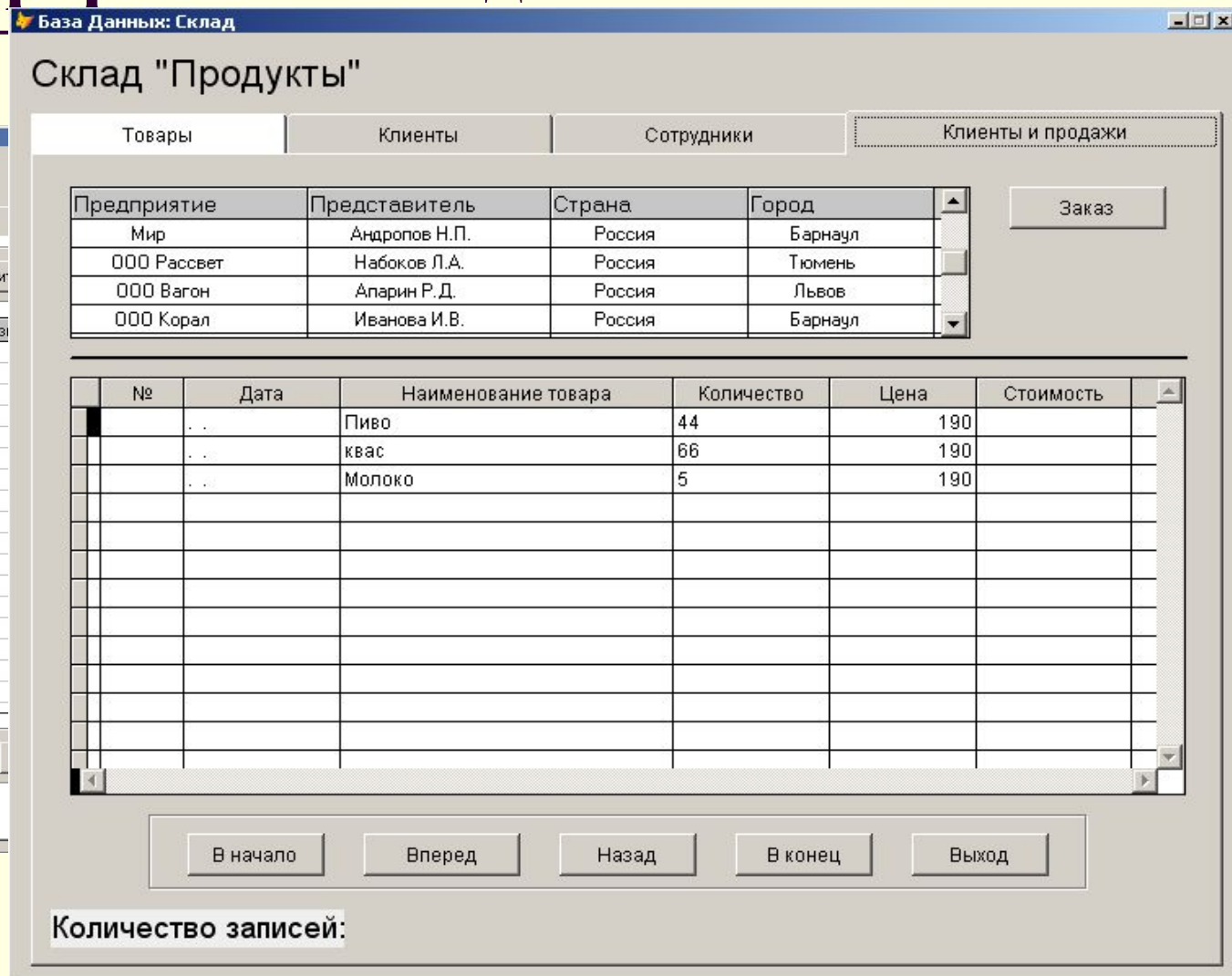
Проект посвящен подготовке мототехники к продаже (предпродажная подготовка)

**Недостатки:** отсутствуют элементы для добавления, удаления и редактирования записей, нет итоговых данных по отобранным записям

# Пример 2: форма с закладками



Тема проекта –  
оптовая торговля



**Недостатки:** отсутствуют элементы для добавления, удаления и редактирования записей на основной закладке, тогда как на вспомогательных они присутствуют, что делает возможным непреднамеренное изменение данных. А вот кнопки перемещения по БД при наличии таблицы не нужны. Нет элементов для фильтрации данных и подсчета выбранных строк на главной вкладке

# Пример 3: многооконное решение

Прайс-лист \* Автомобили \*

Марка	Страна	Цвет	Год	Количество	Цена, тыс.
ваз	Россия	синий	2000	5	30
волга	Россия	бордо	1999	20	125
ниссан	Англия	белый	2001	30	300
москвич	Россия	белый	1998	10	20
ниссан	Япония	корица	1987	12	124

Добавить    Удалить    **Изменить**    Поиск    Сортировка    ВЫХОД

Количество записей 5

Изменить

Марка  
ваз

Цвет  
синий

Страна  
Россия

Год  
2000

Количество  
5

Цена  
30

Изменить    Отмена

Поиск

Марка    Страна    Цвет    Год выпуска

Марка	Страна	Год	Количество	Цена
ваз	Россия	2000	5	30
волга	Россия	1999	20	125
ниссан	Англия	2001	30	300
москвич	Россия	1998	10	20
ниссан	Япония	1987	12	124

Россия  
Англия  
Германия  
США  
Корея  
Япония

Поиск    Отмена    Очистить

Проект – каталог автомобилей

## Недостатки:

1. форма для поиска по заданию не требуется, а вот формы для отбора (фильтрации) данных, причем с полем для текстового ввода условия фильтрации, данных, нет;
2. Критерии фильтрации/ поиска можно было без ущерба разместить на основной форме

# Системы управления базами данных

**Тема: методология разработки проектов на основе СУБД:**

# Выбор тематики разработки

- Для проекта нужно выбирать самостоятельное направление, но главное – наличие на рынке аналогов. Нет аналогов – значит: а) никому не нужно; б) это гениально.
- Нужно хорошо ориентироваться в предметной области. Но это не исключает консультаций и советов со стороны специалистов.
- В проекте должны сочетаться достоинства известных решений плюс содержаться элементы, дающие дополнительный положительный эффект
- В проекте следует максимально использовать существующие наработки и решения сторонних разработчиков (при условии соблюдения авторских прав), а также свои ранее разработанные решения
- Современные проекты – это, как правило, коллективный труд, что требует применения соответствующих технологий.
- Оптимально – делать проекты под реальных конкретных заказчиков, а не рассчитывать на абстрактный рынок.
- По возможности, новый проект должен предполагать возможность его массового тиражирования, возможно, после некоторой доработки.
- Если проект – не результат «допиливания» прежних доработок под нужды Заказчика, очень хорошо, если он рождает новые идеи и решения, которые могут пригодиться в будущем

# Работа с заказчиком

## 1. Рекомендации по презентации проекта:

- идеально - продемонстрировать портфолио и прообраз предлагаемого решения, лучше последним;
- нужно показать преимущества своего решения перед решениями конкурентов;
- в качестве главного аргумента использовать подход «стоимость – эффективность»

## 2. Характеристика круга лиц заказчика:

- Руководители: высшего уровня (решение о финансировании), среднего уровня (заинтересованность), проекта (кураторы, координаторы). Часто бывает только два уровня
- Прямые специалисты – определяют функциональность и требования
- Рядовые исполнители – оценивают результат, сопротивляются внедрению, требуют обучения

## 3. Советы по выбору ценовой политики

- Предлагать стоимость для минимального решения (цена от ...)
- При ограниченности ресурсов оптимизировать функциональность
- Договариваться о поэтапной оплате с возможностью развития работ
- Снижать цену за отдельные этапы, а увеличивать стоимость – расширением функций, добавлением модулей и этапов

## 3. Порядок сбора информации

- Опрос представителей заказчика, начиная с верхнего уровня, с конкретизацией целей, задач и требований, после чего – статистическая обработка информации. Оставляем лишь нужное большинству представителям заказчика!
- Сбор образцов входных и выходных документов, правил ведения дел
- Анализ существующих программных решений, способов ввода и вывода данных, интерфейса. HMI (Human Machine Interface) – максимальное внимание!
- Оценка оптимального объема хранимых данных, необходимости сохранения динамики изменения данных и их архивации

Главное – найти разумный компромисс между требованиями заказчика и Вашими возможностями

# Этапы проектирования

1. **Дать проекту имя**
2. Составить функциональную **модель**, модель информационных потоков. Изучить предметную область
3. Конкретизировать **цель и задачи** проекта в терминах ввода-вывода по результатам опроса, собранных документов, анализа деятельности и технических возможностей заказчика
4. Разработать исходный вариант диаграммы «сущность-связь»
5. Выбрать **среду разработки**: СУБД и клиентского приложения
6. Разработать БД. «Правильная» БД – это от 50% до 90% эффективности работы над проектом и его дальнейшим сопровождением. Стоимость исправления ошибок, выявленных на этапе начала работы над проектом относятся к стоимости исправления ошибок в почти готовом проекте и к стоимости исправления ошибок в уже переданном заказчику и работающем проекте в пропорции 1:5:20 и даже 1:100:10000 (зависит от масштабности проекта)
7. Взять у конкурентов **хорошие идеи** (функциональные возможности, элементы сценария и интерфейса и прочее) и придать им привлекательный и оригинальный вид, чтобы избежать проблем с соблюдением авторских прав
8. Разработать и **согласовать функциональность** приложения. Предварительно согласовать основные экранные формы, интерфейс пользователя. Актуально даже согласование дизайна, стилевых решений, фирменного цвета, слогана, логотипа
9. **Составить календарный план** (этапы, стоимость и сроки их реализации) и оценить общую стоимость работы.
10. Согласовать календарный план, подготовить **договор** на реализацию проекта и подписать его у заказчика, включая **техническое задание**.
11. Окончательно согласовать **интерфейсные решения**: экранные и отчетные формы, элементы оформления (цвета, логотипы и проч.), структуру меню. Лучше все утвердить документально
12. **Разработать и отладить** ПО. Многоэтапаная процедура и разные технологии.
13. Провести **пробную эксплуатацию** ПО и его **тестирование** в реальных условиях. Устранить выявленные ошибки, недостатки и замечания. Подготовить **техдокументацию** (руководства пользователя, программиста, системного администратора)