

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ

В.В. Кашковский

Архитектура вычислительных систем

Лабораторный практикум

Иркутск 2011

УДК 004.2
ББК 32.973-04
К 31

Рецензенты

доцент кафедры авиационных электросистем и пилотажно-навигационных комплексов Иркутского филиала МГТУ ГА, к.т.н., доцент **А.А. Чигвинцев**;
начальник отдела информационной безопасности Иркутского ИВЦ структурного подразделения ГВЦ – филиала ОАО «РЖД» **И.В. Гомич**

Кашковский В.В.

К 31 Архитектура вычислительных систем : лабораторный практикум. – Иркутск : ИрГУПС, 2011. – 88 с.

Лабораторный практикум содержит теоретический материал по разработке программного обеспечения для выполнения лабораторных работ по дисциплине «Архитектура вычислительных систем». Помимо выполнения лабораторных работ по дисциплине «Архитектура вычислительных систем» лабораторный практикум предполагается использовать при выполнении курсового проектирования по дисциплине «Информационные системы на железнодорожном транспорте». Предназначен для студентов технических вузов, обучающихся по специальности «Программное обеспечение вычислительной техники и автоматизированных систем».

УДК 004.2
ББК 32.973-04

© Кашковский В.В., 2011
© Иркутский государственный университет
путей сообщения, 2011

ОГЛАВЛЕНИЕ

| | |
|--|----|
| Введение | 4 |
| Лабораторная работа № 1. Исследование работы вычислительной системы с архитектурой «клиент – сервер» | 5 |
| Лабораторная работа № 2. Исследование работы мультизадачных приложений | 60 |
| Лабораторная работа № 3. Исследование работы мультипрограммных приложений | 66 |
| Лабораторная работа № 4. Исследование работы ассоциативных вычислительных систем | 79 |

Введение

Архитектура вычислительных систем является одной из наиболее динамично развивающихся научных дисциплин. Сложность изучения данной дисциплины состоит в том, что существует ряд важных практических вопросов, связанных с необходимостью более полного использования в программном обеспечении особенностей архитектуры IBM PC-совместимых компьютеров, которые невозможно охватить в рамках краткого лекционного курса. Поэтому был разработан предлагаемый лабораторный практикум, который содержит теоретический материал по разработке программного обеспечения для лабораторных работ по дисциплине «Архитектура вычислительных систем». Форма представления данного пособия как электронного выбрана по двум причинам:

1. Электронный лабораторный практикум удобнее для выполнения лабораторных работ, чем тот же самый материал на бумажном носителе, поскольку позволяет осуществлять быстрый поиск нужного материала, а главное, дает возможность копирования приведенных примеров для переноса в среду программирования.

2. В условиях изменения учебных программ печатные учебные пособия быстро устаревают и не могут адаптироваться к изменяющимся формам обучения.

В качестве языка программирования при выполнении лабораторных работ выбрана Turbo Delphi, как наиболее доступная и простая для изучения среда и язык программирования.

Лабораторный практикум охватывает четыре темы, связанные с организацией параллельных вычислений и работы ассоциативных вычислительных систем. Особенностью лабораторного практикума является то, что лабораторную работу № 1 «Исследование работы вычислительной системы с архитектурой «клиент – сервер» предполагается использовать не только для изучения дисциплины «Архитектура вычислительных систем», но и в качестве учебной базы данных и клиентского приложения, построенного на технологии ADO, при выполнении курсового проектирования по дисциплине «Информационные системы на железнодорожном транспорте».

На каждую лабораторную работу отводится 4 академических часа.

Лабораторная работа № 1

ИССЛЕДОВАНИЕ РАБОТЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ С АРХИТЕКТУРОЙ «КЛИЕНТ – СЕРВЕР»

Цель работы

Разработать приложение с архитектурой «клиент – сервер» в Turbo Delphi и исследовать его работу.

Указания к работе

1. Разработать в Turbo Delphi приложение по предлагаемому образцу.
2. Исследовать работу ВС с архитектурой «клиент – сервер».
3. Оформить отчет.

Пример использования ADO для работы с БД

Создадим приложение, которое позволяло бы редактировать и просматривать информацию, содержащуюся в базе данных Market.mdb. Указанная база данных создана с помощью Microsoft Access и содержит сведения о товарах, имеющихся в ассортименте некоторого магазина, торгующего комплектующими к вычислительной технике, а также данные о заказах, сделанных покупателями.

В этой базе данных созданы три таблицы.

Таблица Products содержит сведения о товарах, которые имеются в магазине. В этой таблице представлены такие поля:

- идентификационный номер товара «ТоварID» (тип Счетчик; первичный ключ);
- наименование товара «Товар» (тип Строковый (50));
- «Изготовитель» (тип Строковый (50));
- «Модель» (тип Строковый (50), индексированное поле, совпадения не допускаются);
- «Технические характеристики» (тип Строковый (150));
- «Цена» (тип Денежный);
- количество единиц товара на складе «Наличие на складе» (тип Целый);
- количество заказываемых единиц товара «Количество» (тип Целый).

Последнее из указанных полей играет вспомогательную роль. Далее это поле будет использовано только с целью указания количества приобретаемого товара при оформлении нового заказа (по умолчанию данное поле будет всегда равно 1). Поле «Количество» введено в рассматриваемом примере для того, чтобы, с одной стороны, сделать вполне работоспособное при-

ложение, с другой стороны, не слишком усложнять пример и, наконец, с третьей стороны, проиллюстрировать некоторые особенности использования технологии ADO, такие как, например, фильтрация по закладкам.

| Products : таблица | | |
|--------------------|----------------------------|------------|
| | Имя поля | Тип данных |
| 🔑 | ТоварID | Счетчик |
| | Товар | Текстовый |
| | Изготовитель | Текстовый |
| | Модель | Текстовый |
| | Технические характеристики | Текстовый |
| | Цена | Денежный |
| | Наличие на складе | Числовой |
| | Количество | Числовой |

Таблица «Products»

В таблицу Orders вносятся данные о заказах, сделанных клиентами. Эта таблица включает в себя следующие поля:

- номер записи «RecID» (тип *Счетчик*; первичный ключ);
- номер заказа «Номер» (тип *Длинное целое*);
- «Дата заказа» (тип *Дата*);
- заказчик «Customer» (тип *Строковый (70)*);
- идентификационный номер товара «ТоварID» (тип *Длинное целое*);
- идентификационный номер заказчика «ПокупательID» (тип *Длинное целое*);
- количество единиц заказанного товара «Получено» (тип *Целый*).

| Orders : таблица | | |
|------------------|--------------|------------|
| | Имя поля | Тип данных |
| 🔑 | RecID | Счетчик |
| | Номер | Числовой |
| | Дата заказа | Дата/время |
| | ТоварID | Числовой |
| | Получено | Числовой |
| | ПокупательID | Числовой |

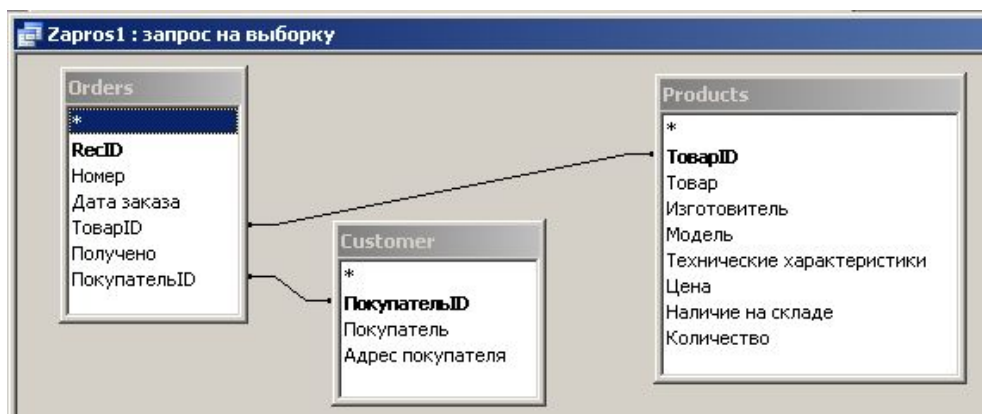
Таблица «Orders»

- В таблицу «Customer» вносятся данные о заказчике (покупателе):
- номер заказчика «ПокупательID» (тип *Счетчик*; первичный ключ);
 - заказчик «Покупатель» (тип *Строковый (30)*, индексированное поле, совпадения не допускаются);
 - адрес заказчика «Адрес покупателя» (тип *Строковый (30)*).

| Customer : таблица | | |
|--------------------|------------------|------------|
| | Имя поля | Тип данных |
| 🔑 | ПокупательID | Счетчик |
| | Покупатель | Текстовый |
| | Адрес покупателя | Текстовый |

Таблица «Customer»

Связь между таблицами иллюстрирует схема запроса «Zapros1»:



При наличии этой базы данных в Access перейдем непосредственно к разработке приложения Delphi, которое будет обеспечивать управление этой БД.

Первый этап заключается в проектировании главной формы приложения.

Учет заказов

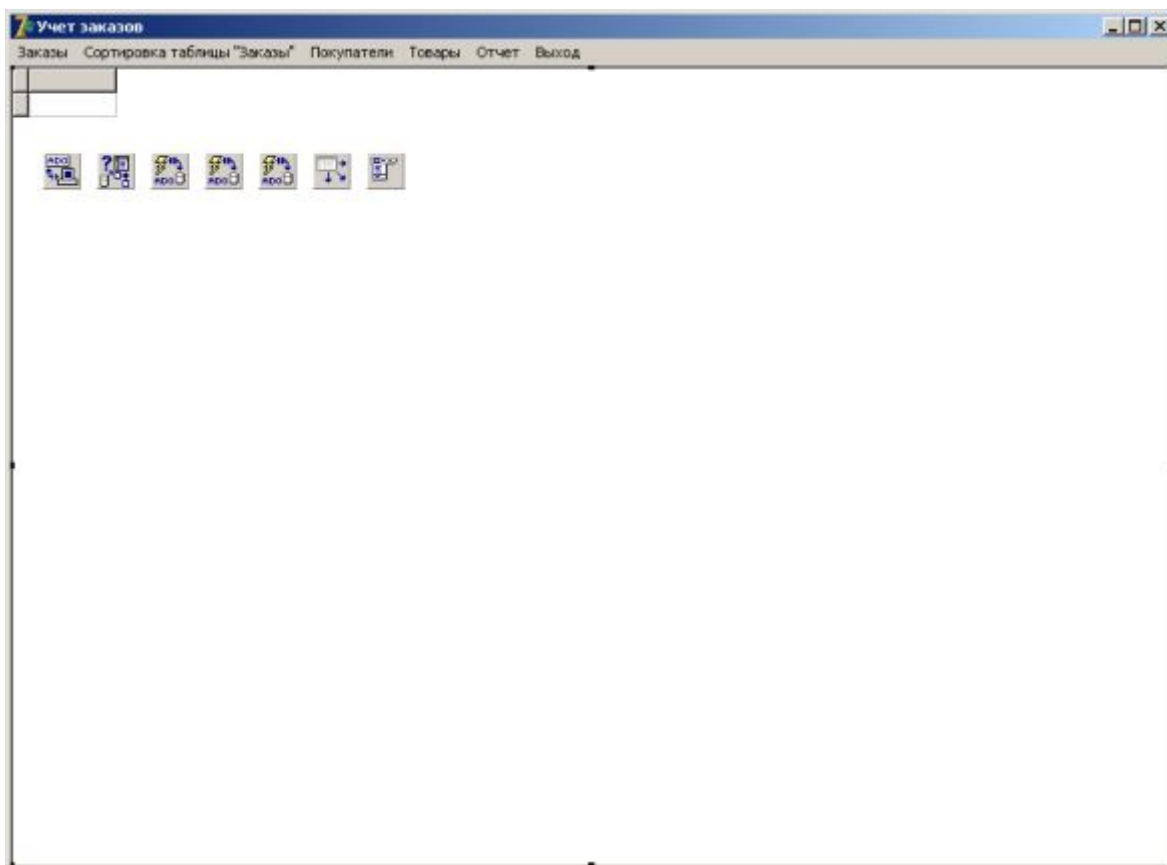
Заказы Сортировка таблицы "Заказы" Покупатели Товары Отчет Выход

| Код записи | Код заказа | Дата заказа | Заказчик | Адрес заказчика | Товар |
|------------|------------|-------------|-----------------------------|--|--------|
| 102 | 1 | 18.06.2008 | ЗАО Афродита | г. Иркутск ул. Александра Невского 39 | Диван |
| 104 | 1 | 18.06.2008 | ЗАО Афродита | г. Иркутск ул. Александра Невского 39 | Мат |
| 105 | 1 | 18.06.2008 | ЗАО Афродита | г. Иркутск ул. Александра Невского 39 | Мозги |
| 106 | 1 | 18.06.2008 | ЗАО Афродита | г. Иркутск ул. Александра Невского 39 | Привод |
| 108 | 2 | 03.02.2009 | ЗАО Петров-Водкин и сыновья | г. Иркутск ул. Донская 57 | Кулер |
| 109 | 2 | 03.02.2009 | ЗАО Петров-Водкин и сыновья | г. Иркутск ул. Донская 57 | Мат |
| 110 | 2 | 03.02.2009 | ЗАО Петров-Водкин и сыновья | г. Иркутск ул. Донская 57 | Мозги |
| 111 | 2 | 03.02.2009 | ЗАО Петров-Водкин и сыновья | г. Иркутск ул. Донская 57 | Привод |
| 112 | 3 | 05.03.2009 | ОАО Инклюзия | г. Иркутск ул. Рабочего штаба 75 | Диван |
| 113 | 3 | 05.03.2009 | ОАО Инклюзия | г. Иркутск ул. Рабочего штаба 75 | Кулер |
| 114 | 3 | 05.03.2009 | ОАО Инклюзия | г. Иркутск ул. Рабочего штаба 75 | Мат |
| 115 | 3 | 05.03.2009 | ОАО Инклюзия | г. Иркутск ул. Рабочего штаба 75 | Мозги |
| 117 | 4 | 22.05.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Диван |
| 118 | 4 | 22.05.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Кулер |
| 119 | 4 | 22.05.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Мат |
| 120 | 4 | 22.05.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Мозги |
| 121 | 4 | 22.05.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Привод |
| 122 | 5 | 17.10.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Диван |
| 123 | 5 | 17.10.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Кулер |
| 125 | 5 | 17.10.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Мозги |
| 126 | 5 | 17.10.2009 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 | Привод |
| 127 | 6 | 08.01.2010 | Сидоров индустри | г. Иркутск ул. Советская д. 125 оф. 17 | Диван |
| 129 | 6 | 08.01.2010 | Сидоров индустри | г. Иркутск ул. Советская д. 125 оф. 17 | Мат |
| 130 | 6 | 08.01.2010 | Сидоров индустри | г. Иркутск ул. Советская д. 125 оф. 17 | Мозги |
| 131 | 6 | 08.01.2010 | Сидоров индустри | г. Иркутск ул. Советская д. 125 оф. 17 | Привод |
| 132 | 7 | 08.01.2010 | ЧП Адама Горидзе | г. Иркутск ул. Трилиссера 15 | Диван |
| 133 | 7 | 08.01.2010 | ЧП Адама Горидзе | г. Иркутск ул. Трилиссера 15 | Кулер |
| 134 | 7 | 08.01.2010 | ЧП Адама Горидзе | г. Иркутск ул. Трилиссера 15 | Мат |
| 135 | 7 | 08.01.2010 | ЧП Адама Горидзе | г. Иркутск ул. Трилиссера 15 | Мозги |
| 136 | 7 | 08.01.2010 | ЧП Адама Горидзе | г. Иркутск ул. Трилиссера 15 | Привод |

Главная форма приложения «Учет заказов»

Эта форма будет служить для того, чтобы пользователь имел возможность просмотреть полный список заказов и отредактировать его в случае необходимости.

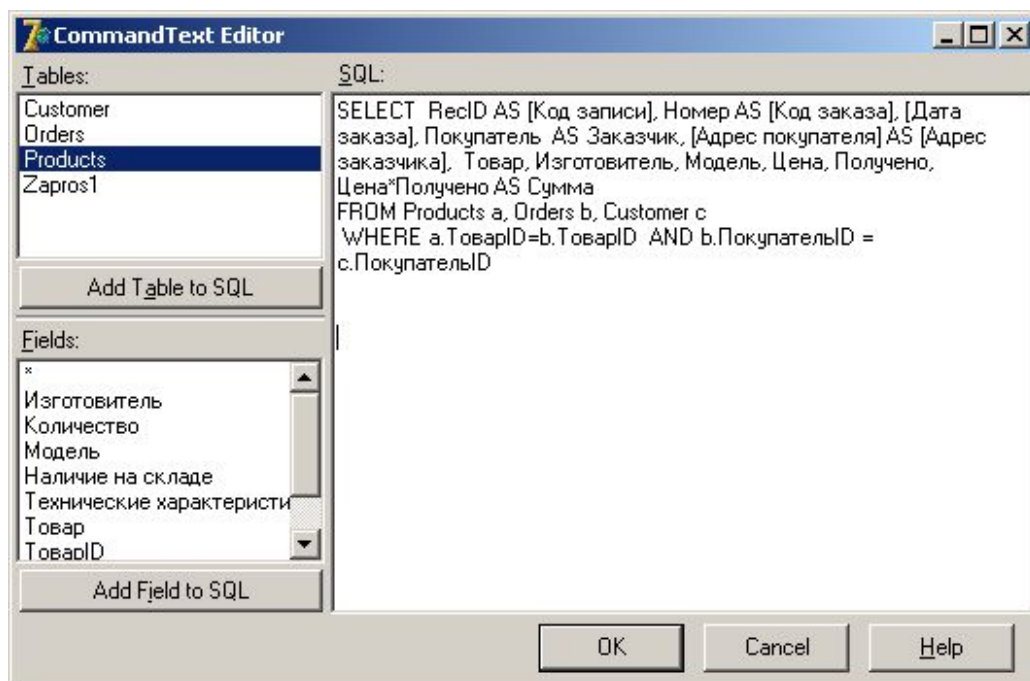
На форме должны быть размещены следующие компоненты:



1. Компонент соединения ADOConnection1. Для него необходимо выполнить настройку свойства ConnectionString, как это было описано при изучении компонента *ADOConnection*:

```
Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Data Source=Market.mdb;Mode=Share Deny None;Extended Properties="";Jet OLEDB:System database="";Jet OLEDB:Registry Path="";Jet OLEDB:Database Password="";Jet OLEDB:Engine Type=5;Jet OLEDB:Database Locking Mode=1;Jet OLEDB:Global Partial Bulk Ops=2;Jet OLEDB:Global Bulk Transactions=1;Jet OLEDB:New Database Password="";Jet OLEDB:Create System Database=False;Jet OLEDB:Encrypt Database=False;Jet OLEDB:Don't Copy Locale on Compact=False;Jet OLEDB:Compact Without Replica Repair=False;Jet OLEDB:SFP=False
```

2. Набор данных ADODataset1. В свойстве Connection данного компонента устанавливается ссылка на компонент ADOConnection1. В редакторе свойства CommandText записывает следующий текст запроса:



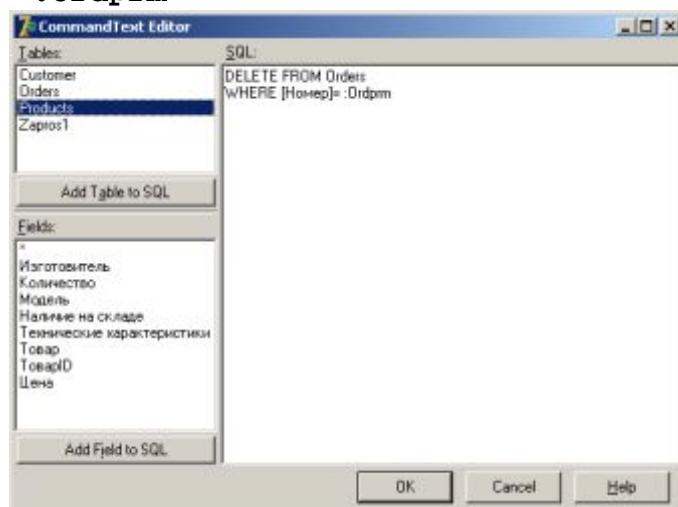
Редактирование набора данных ADODataset1

```
SELECT  RecID AS [Код записи], Номер AS [Код заказа], [Дата
заказа], Покупатель AS Заказчик, [Адрес покупателя] AS [Адрес
заказчика], Товар, Изготовитель, Модель, Цена, Получено, Це-
на*Получено AS Сумма
FROM Products a, Orders b, Customer c
WHERE a.ТоварID=b.ТоварID AND b.ПокупательID = c.ПокупательID
```

3. Три компонента ADOCommand1, ADOCommand2 и ADOCommand3.

С помощью ADOCommand1 будет обеспечиваться аннулирование (удаление) текущего заказа. В свойство CommandText этого компонента следует записать такой текст запроса:

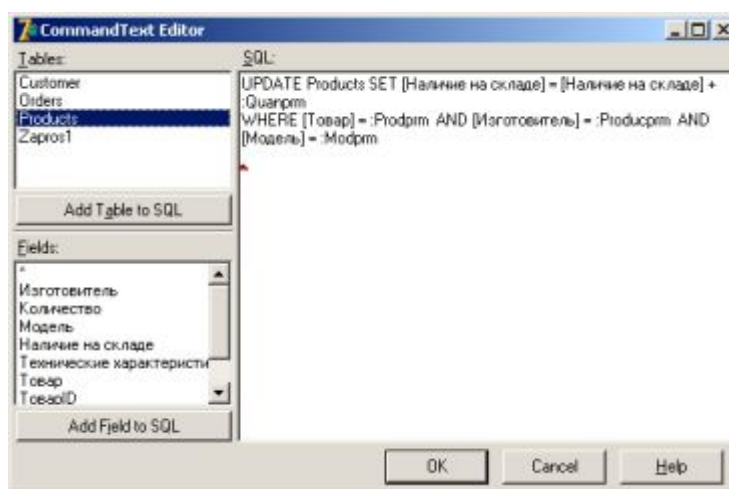
```
DELETE FROM Orders
WHERE [Номер]= :Ordprm
```



Редактирование компонента ADOCommand1

Компонент ADOCommand2 служит для восстановления прежнего значения поля **[Наличие на складе]** таблицы Products (т. е. количества товара на складе) при удалении заказа. Это делается при помощи запроса такого вида:

```
UPDATE Products SET [Наличие на складе] = [Наличие на складе] +
:Quanprm
WHERE [Товар] = :Prodprm AND [Изготовитель] = :Producprm AND
[Модель] = :Modprm
```



Редактирование компонента ADOCommand2

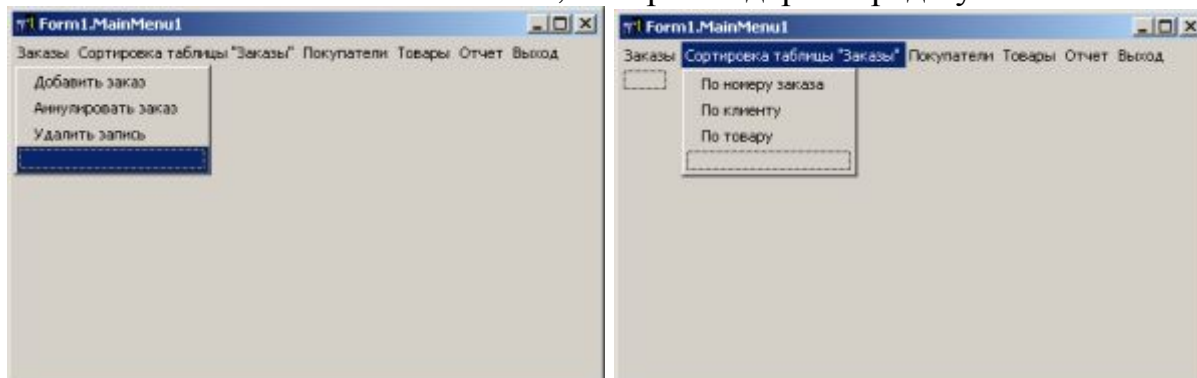
Компонент ADOCommand3 служит для удаления выбранной записи таблицы Orders. Это делается при помощи запроса такого вида:

```
DELETE FROM Orders
WHERE RecID = :Ordprm
```

4. Источник данных DataSource1 из раздела Data Access, свойству DataSet которого присваивается значение ADODataset1.

5. Сетка DBGrid1 из раздела Data Controls. Свойство DataSource данного компонента должно иметь значение DataSource1. Приложение не предусматривает возможность прямого редактирования данных, поэтому свойство Readonly компонента DBGrid1 устанавливается в true, что делает отображаемые в сетке записи доступными только для чтения.

6. Главное меню MainMenu1, которое содержит ряд пунктов:



Пункты главного меню:

- *Заказы*;
- *Сортировка таблицы «Заказы»*;
- *Покупатели* (обработчик события TForm1.N9Click);
- *Товары* (обработчик события TForm1.N10Click);
- *Отчет* (обработчик события TForm1.N12Click);
- *Выход* (Обработчик события TForm1.N3Click).

В свою очередь пункт *Заказы* включает в себя три подпункта:

- *Добавить* (добавить новый заказ, обработчик события TForm1.N4Click);
- *Аннулировать заказ* (аннулирует выбранный заказ, т. е. удаляет все записи с таким номером заказа в таблице *Orders* и возвращает число приобретенных товаров в соответствующие записи поля [Наличие на складе] таблицы *Products*, обработчик события TForm1.N5Click);
- *Удалить запись* (удаляет выбранную запись, обработчик события TForm1.N11Click).

Пункт *Сортировка* содержит три подпункта:

- *по номеру заказа* (обработчик события TForm1.N6Click);
- *по клиенту* (обработчик события TForm1.N7Click);
- *по товару* (обработчик события TForm1.N8Click).

При выборе пункта меню *Заказ | Добавить заказ* на экране будет открываться окно формирования нового заказа.

Добавление нового заказа

Покупатель

| Код покупателя | Покупатель | Адрес покупателя |
|----------------|-----------------------------|---------------------------------------|
| 39 | ЗАО Агроинте | г. Иркутск ул. Александра Невского 39 |
| 35 | ЗАО Петров-Водкин и сыновья | г. Иркутск ул. Дамская 57 |
| 37 | ОАО Инкозния | г. Иркутск ул. Рабочего штаба 75 |
| 33 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 |

Новый заказчик

Новый товар

Номер заказа: 12

Сохранить

Список товаров

| Код товара | Товар | Изготовитель | Модель | Технические характеристики | Цена | Наличие на складе |
|------------|-----------|--------------|-------------|----------------------------|---------|-------------------|
| 3 | Двигатель | Хитачи | FGH-456 | Самый крутой | 1254.52 | 693 |
| 5 | Кулер | Интел | B3P-321 | Ваще | 454.52 | 247 |
| 4 | Мать | Сименс | RHJ-386 | Облаженные | 2560.25 | 603 |
| 6 | Мозги | Сони | DDR-1024.56 | Мрак | 932.78 | 597 |
| 1 | Привод | Томат | MHP-362 | Хороший | 1430.18 | 365 |

Состав заказа

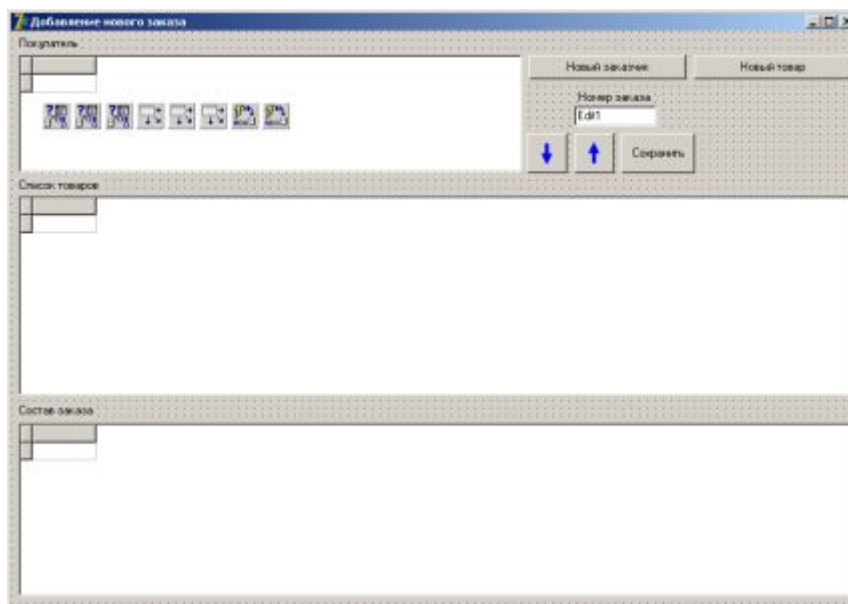
| Код товара | Товар | Изготовитель | Модель | Технические характеристики | Цена | Наличие на складе | Количество |
|------------|-----------|--------------|-------------|----------------------------|---------|-------------------|------------|
| 3 | Двигатель | Хитачи | FGH-456 | Самый крутой | 1254.52 | 693 | 2 |
| 5 | Кулер | Интел | B3P-321 | Ваще | 454.52 | 247 | 6 |
| 4 | Мать | Сименс | RHJ-386 | Облаженные | 2560.25 | 603 | 8 |
| 6 | Мозги | Сони | DDR-1024.56 | Мрак | 932.78 | 597 | 3 |

Форма добавления нового заказа

На эту форму следует поместить следующие компоненты.

1. Три набора данных ADODataset1, ADODataset2 и ADODataset3. Первый из этих наборов данных представляет собой полный список товаров, имеющих на складе. Он формируется с помощью запроса следующего вида:

```
SELECT ТоварID AS [Код товара], Товар, Изготовитель, Модель,
[Технические характеристики], Цена, [Наличие на складе] FROM
Products
WHERE [Наличие на складе]>0
ORDER BY Товар, Изготовитель, Модель
```



Второй набор данных ADODataset2 представляет собой список товаров, выбранных пользователем для текущего заказа. Он формируется аналогичным запросом:

```
SELECT ТоварID AS [Код товара], Товар, Изготовитель, Модель,
[Технические характеристики], Цена, [Наличие на складе],
Количество FROM Products
WHERE [Наличие на складе]>0
ORDER BY Товар, Изготовитель, Модель
```

Поле **Количество** пользователь может скорректировать вручную, задав требуемое количество товара.

Отличие набора данных ADODataset2 от ADODataset1 состоит в том, что на него накладывается фильтр по закладкам с помощью метода FilterOnBookmarks (см. листинг). Закладки формируются по записям, выбранным пользователем в списке товаров.

Третий набор данных ADODataset3 позволяет выбрать покупателя (заказчика):

```
SELECT ПокупательID AS [Код покупателя], Покупатель, [Адрес
покупателя]
FROM Customer
ORDER BY Покупатель
```

2. Три источника данных DataSource1, DataSource2, и DataSource3, связанных соответственно с наборами данных ADODataset1, ADODataset2 и ADODataset3.

3. Два компонента ADOCommand1 и ADOCommand2. С помощью первого из этих компонентов осуществляется добавление записей, относящихся к новому заказу, в таблицу Orders. Для этого используется запрос следующего вида:

```
INSERT INTO Orders(Номер, [Дата заказа], ПокупательID, ТоварID,
[Получено]) VALUES(:OIDprm, :ODateprm, :Custprm, :PIDprm,
:Quanprm)
```

Второй компонент ADOCommand2 служит для корректировки значений полей в таблице *Products*. Запрос, хранящийся в свойстве CommandText данного компонента, имеет такой вид:

```
UPDATE Products SET [Наличие на складе] = [Наличие на складе] -
:Quanprm, Количество = 1
WHERE ТоварID = :PIDprm
```

С помощью данного запроса значение поля **[Наличие на складе]** уменьшается на количество заказанных единиц товара (оно хранится в поле **Количество**), а затем в самом поле **Количество** вновь устанавливается исходное значение 1.

Необходимо отметить, что использование двух компонентов *ADOCommand*, как для первой, так и для второй формы, носит чисто иллюстративный характер, ради обеспечения большей наглядности. На самом же деле более правильным будет использование одного такого компонента и динамическое изменение текста запроса в его свойстве CommandText в процессе выполнения приложения.

4. Три сетки DBGrid1, DBGrid2 и DBGrid3, связанные соответственно с DataSource1, DataSource2 и DataSource3, а через них с набором данных ADODataset1, ADODataset2 и ADODataset3.

В DBGrid1 будет отображаться список всех товаров. Записи, отображаемые в этой сетке, должны быть доступны только для чтения, поэтому свойство Readonly компонента DBGrid1 задается равным true. Кроме того, должна обеспечиваться возможность выбора сразу нескольких товаров, которые необходимо включить в состав заказа. Для этого к свойству Options сетки DBGrid1 добавляется опция dgMultiSelect.

Вторая сетка DBGrid2 выводит состав заказа (необходимые товары, выбранные из списка, отображаемого в DBGrid1). Записи, содержащиеся в

DBGrid2, доступны для редактирования – именно здесь пользователь может указать требуемое количество каждого товара.

Третья сетка DBGrid3 выводит название организации-заказчика товаров и ее адрес.

В редакторе Edit1 содержится номер формируемого заказа. Этот номер задается автоматически и вычисляется как максимальный номер из всех существующих заказов, увеличенный на единицу. Изменение этого номера вручную запрещено, поэтому свойство Readonly данного редактора устанавливается в true.

5. Две быстрые кнопки SpeedButton1 и SpeedButton2. Первая кнопка с изображением стрелки, направленной вниз, реализует перенос выделенных в списке товаров во второй список, который определяет состав заказа. По сути дела, при нажатии кнопки SpeedButton1 для набора данных ADO-DataSet2 устанавливается, а затем активизируется фильтр по закладкам, сформированным на основе записей, выделенных пользователем в списке товаров. После этого отфильтрованный таким образом набор данных активизируется, причем в нем отображаются только те записи, которые были ранее выбраны пользователем в DBGrid1.

Вторая кнопка со стрелкой, направленной вверх, очищает список заказанных товаров (фактически отменяет фильтрацию и закрывает набор данных ADODataset2).

6. Кнопка Button1 с заголовком *Сохранить*. По нажатию этой кнопки выполняется добавление новых записей в таблицу заказов Orders и корректировка значений полей [**Наличие на складе**] и **Количество** в таблице Products, после чего данная форма закрывается и пользователю снова становится доступным список заказов, содержащий в том числе и вновь внесенную информацию.

7. Кнопка Button2 с заголовком *Новый заказчик*. Вызывает форму *form3* для корректировки записей в таблице *Customer*.

8. Кнопка Button3 с заголовком *Новый товар*. Вызывает форму *form4* для корректировки записей в таблице *Products*.

Form3 выглядит следующим образом:

| Код покупателя | Покупатель | Адрес покупателя |
|----------------|-----------------------------|--|
| 39 | ЗАО Афродита | г. Иркутск ул. Александра Невского 39 |
| 35 | ЗАО Петров-Водкин и сыновья | г. Иркутск ул. Донская 57 |
| 37 | ОАО Инклюзия | г. Иркутск ул. Рабочего штаба 75 |
| 33 | ОАО Феникс | г. Иркутск ул. Баррикадная 35 |
| 29 | Сидоров индустриал | г. Иркутск ул. Советская д. 125 оф. 17 |
| 31 | ЧП Адама Горидзе | г. Иркутск ул. Трилиссера 15 |
| 36 | Шушарин энд компани | г. Иркутск ул. Батарейная 48 |

Форма содержит два набора данных `ADODataset1` и `ADODataset2`. Первый из этих наборов данных представляет собой список покупателей и их адресов. Он формируется с помощью запроса следующего вида:

```
SELECT ПокупательID AS [Код покупателя], Покупатель, [Адрес покупателя]
FROM Customer
ORDER BY Покупатель
```

Второй набор данных `ADODataset2` предназначен для проверки записей таблицы `Customer`, исключающей возможность ввода одинаковых значений в поле `Покупатель`:

```
SELECT ПокупательID, Покупатель
FROM Customer
WHERE Покупатель = :Custprm AND ПокупательID <> :CustIDprm
```

Сетка `DBGrid1` связана с `DataSource1`, а через него с набором данных `ADODataset1`. В `DBGrid1` будет отображаться список всех покупателей. Записи, отображаемые в этой сетке, должны быть доступны только для чтения, поэтому свойство `ReadOnly` компонента `DBGrid1` задается равным `true`.

Редактирование текущей записи или ввод новой записи осуществляется через редакторы `Edit1` и `Edit2`. Для этого используются кнопки

- Button1 «Изменить»;
- Button2 «Выход»;
- Button3 «Добавить»;
- Button4 «Удалить».

С кнопками связаны четыре `ADOCommand`.

Первый `ADOCommand1` предназначен для редактирования текущей записи:

```
UPDATE Customer SET Покупатель = :Cuprm, [Адрес покупателя] = :Adrprm
WHERE ПокупательID = :CuIDprm
```

Второй *ADOCommand2* предназначен для ввода новой записи:

```
INSERT INTO Customer(Покупатель, [Адрес покупателя])  
VALUES (:Cuprm, :Adrprm)
```

Третий и четвертый *ADOCommand3* и *ADOCommand4* предназначены для удаления записей покупателя в таблицах **Orders** и **Customer**:

```
DELETE FROM Orders  
WHERE ПокупательID= :CuIDdprm
```

```
DELETE FROM Customer  
WHERE ПокупательID= :CuIDdprm
```

Form4 выглядит следующим образом:

Редактирование таблицы "Товар"

Товар: Edit1 Изготовитель: Edit2

Модель: Edit3 Технические характеристики: Edit4

Наличие на складе: Edit5 Цена: Edit6

Изменить

Добавить

Удалить

Выход

Редактирование таблицы "Товар"

Товар: Дивядюк Изготовитель: Хитачи

Модель: FGH-456 Технические характеристики: Самый крутой

Наличие на складе: 693 Цена: 1254.52

| Код товара | Товар |
|------------|---------|
| 3 | Дивядюк |
| 5 | Кулер |
| 4 | Мать |
| 6 | Мозги |
| 1 | Привод |

Изменить

Добавить

Удалить

Выход

Данная форма устроена и работает точно так же, как *Form3*, но выполняет операции с таблицей *Products*. Поскольку в таблице *Products* больше полей, чем в таблице *Orders*, *Form4* содержит шесть редакторов Edit.

Form5 выглядит следующим образом:

7' Отчет о закупке товаров за последние пять лет

Отчет по фирмам через набор данных | Отчет по моделям через набор данных | Отчет по фирмам через SQL запросы

Общие заказы | По моделям | DDR-1024.56 | Выход

Закуплено единиц товара

| Покупатель | Всего | 2010 год | 2009 год | 2008 год | 2007 год | 2006 год |
|------------------------|-------|----------|----------|----------|----------|----------|
| ЗАО Афродита | 382 | 343 | | 39 | | |
| ЗАО Петров-Водкин и сы | 28 | | 28 | | | |
| ОАО Инклюзия | 396 | 355 | 41 | | | |
| ОАО Феникс | 110 | 74 | 36 | | | |
| Сидоров индастриз | 8 | 8 | | | | |
| ЧП Адама Горидзе | 25 | 25 | | | | |
| Шушарин энд компани | 27 | 27 | | | | |
| Итого | 976 | 832 | 105 | 39 | | |

Сумма (рублей)

| Покупатель | Всего | 2010 год | 2009 год | 2008 год | 2007 год | 2006 год |
|------------------------|------------|------------|-----------|----------|----------|----------|
| ЗАО Афродита | 542652.70 | 488066.29 | | 54586.41 | | |
| ЗАО Петров-Водкин и сы | 38543.21 | | 38543.21 | | | |
| ОАО Инклюзия | 517434.30 | 459128.85 | 58305.45 | | | |
| ОАО Феникс | 115851.07 | 79353.37 | 36497.70 | | | |
| Сидоров индастриз | 12355.46 | 12355.46 | | | | |
| ЧП Адама Горидзе | 33161.25 | 33161.25 | | | | |
| Шушарин энд компани | 28087.76 | 28087.76 | | | | |
| Итого | 1288085.75 | 1100152.98 | 133346.36 | 54586.41 | | |

Данная форма предназначена для демонстрации различных технологий формирования сложных запросов. Форма содержит семь компонент *ADODataset*. Соответствующие SQL-запросы данных компонент приведены в листинге программы.

Все компоненты *ADODataset* и *ADOCommand* всех пяти форм подключаются к базе данных через компонент соединения *ADOConnection1*, ссылка на который устанавливается в свойстве *Connection* каждого из компонентов. Для того чтобы компоненты второй формы получили доступ к компоненту *ADOConnection1*, размещенному на первой форме, необходимо предварительно в разделе *uses* каждого модуля (*EditWin – form2*; *Colump – form3*; *Product – form4* и *Reliz – form5*) приложения сослаться на главный модуль (например, *MainBDADO*) и сохранить изменения.

Для подключения данных модулей к головному в нем после служебного слова *implementation* необходимо вставить следующую строку:

```
uses EditWin, Colump, Product, Reliz;
```

Листинг программы содержит программный код создаваемого приложения (в частности, код его главного модуля, позволяющего просматривать список заказов, и код модуля, реализующего возможность добавления новых заказов).

```

// Код головного модуля

unit MainBDADO;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, Menus, Grids, DBGrids, DB, ADODB;

type
  TForm1 = class(TForm)
    ADOConnection1: TADOConnection;
    ADODataSet1: TADODataSet;
    ADOCommand1: TADOCommand;
    ADOCommand2: TADOCommand;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    N11: TMenuItem;
    ADOCommand3: TADOCommand;
    N12: TMenuItem;
    procedure N4Click(Sender: TObject);
    procedure N5Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N7Click(Sender: TObject);
    procedure N8Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure ADODataSet1AfterOpen(DataSet: TDataSet);
    procedure FormShow(Sender: TObject);
    procedure N9Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure N11Click(Sender: TObject);
    procedure N12Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
end;

```

```

var
  Form1: TForm1;
  ADF      : array [0..255] of Char;

implementation

uses EditWin, Colump, Product, Reliz;

{$R *.dfm}

{ Переход к формированию нового заказа }

procedure TForm1.N4Click(Sender: TObject);
var st: string; buf: integer;
begin
  { Автоматическая генерация номера нового заказа }
  st:= ADODataset1.Sort;
  buf:= ADODataset1.RecNo;
  ADODataset1.Sort:= '[Код заказа]';
  ADODataset1.Last;
  if ADODataset1.RecordCount = 0 then Form2.Edit1.Text:='1'
    else Form2.Edit1.Text:=
ADODataset1.FieldValues['Код заказа'] + 1;
  ADODataset1.Sort:= st;
  ADODataset1.RecNo:= buf; { Открытие формы добавления нового
заказа }
  Form2.ShowModal;
end;

{ Удаление записи }

procedure TForm1.N5Click(Sender: TObject);
begin

  StrPCopy(ADF, 'Вы уверены, что необходимо аннулировать данный
заказ?');

  if Application.MessageBox(ADF,'Внимание: выполняется операция
аннулирования заказа!',
    MB_OKCANCEL + MB_DEFBUTTON1) <> IDOK then exit;

  { Передача Номера текущей записи в команды ADO }

  ADOCommand1.Parameters.ParamByName('Ordprm').Value
:=ADODataset1.FieldValues['Код заказа'];
  ADOCommand2.Parameters.ParamByName('Quanprm').Value
:=ADODataset1.FieldValues['Получено'];
  ADOCommand2.Parameters.ParamByName('Prodprm').Value
:=ADODataset1.FieldValues['Модель'];

  ADODataset1.Close;

{

```

```

    Выполнение команд ADO (удаление записей, относящихся
    к текущему заказу, и восстановление значений полей таблицы
    товаров)
}

    ADOCommand1.Execute;
    ADOCommand2.Execute;
    ADODataset1.Open;
end;

{ Сортировка по номеру заказа }

procedure TForm1.N6Click(Sender: TObject);
begin
    ADODataset1.Sort:= '[Код заказа]';
end;

{ Сортировка по клиенту }

procedure TForm1.N7Click(Sender: TObject);
begin
    ADODataset1.Sort:= 'Заказчик';
end;

{ Сортировка по товару }

procedure TForm1.N8Click(Sender: TObject);
begin
    ADODataset1.Sort:= 'Товар, Изготовитель, Модель';
end;

{ Выход }

procedure TForm1.N3Click(Sender: TObject);
begin
    Close;
end;

{ Изначальная сортировка по Номеру заказа }

procedure TForm1.ADODataSet1AfterOpen(DataSet: TDataSet);
begin
    ADODataset1.Sort:= '[Код заказа]';
end;

procedure TForm1.FormShow(Sender: TObject);
begin

{ Открытие соединения и набора данных при отображении главной
формы }

```

```

    ADOConnection1.Open;
    ADODataSet1.Open;
end;

procedure TForm1.N9Click(Sender: TObject);
begin
    Form3.ShowModal;
end;

procedure TForm1.N10Click(Sender: TObject);
begin
    Form4.ShowModal;
end;

procedure TForm1.N11Click(Sender: TObject);
begin
    StrPCopy(ADF, 'Вы уверены, что необходимо удалить данную запись?');

    if Application.MessageBox(ADF, 'Внимание: выполняется операция удаления записи!',
        MB_OKCANCEL + MB_DEFBUTTON1) <> IDOK then exit;

    { Передача Номера текущей записи в команду ADO }

    ADOCom-
mand3.Parameters.ParamByName('Ordprm').Value:=ADODataSet1.Fiel
dValues['Код записи'];

    ADODataSet1.Close;

    { Выполнение команды ADO }

    ADOCommand3.Execute;

    ADODataSet1.Open;
end;

procedure TForm1.N12Click(Sender: TObject);
begin
    Form5.Show;
end;

end.

// Код модуля, обеспечивающего добавление нового заказа

unit EditWin;

interface

```

```

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, Buttons, Grids, DBGrids, ADODB, DB, DBCtrls;

type
  TForm2 = class(TForm)
    ADODataSet1: TADODataSet;
    ADODataSet2: TADODataSet;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    ADOCommand1: TADOCommand;
    ADOCommand2: TADOCommand;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    Button1: TButton;
    ADODataSet3: TADODataSet;
    DataSource3: TDataSource;
    DBGrid3: TDBGrid;
    Button2: TButton;
    Button3: TButton;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;
  BM: array of TVarRec; { Массив закладок }

implementation

uses MainBDADO, Colump, Product;

{$R *.dfm}

```



```

{ Фильтрация по закладкам (вывод товаров, включенных
  в состав заказа) }

procedure TForm2.SpeedButton1Click(Sender: TObject);
var i: integer;
begin
  if DBGrid1.SelectedRows.Count > 0 then
  begin
    { Сохранение закладок по записям, выделенным в DBGrid1,
      в динамическом массиве BM }

    SetLength(BM, DBGrid1.SelectedRows.Count);
    for i:= 0 to DBGrid1.SelectedRows.Count - 1 do
    begin
      BM[i].VType:= vtPointer;
      BM[i].VPointer:= pointer(DBGrid1.SelectedRows[i]);
    end;

    { Открытие второго набора данных ADODataset2
      и включение для него фильтрации по закладкам }

    ADODataset2.Open;
    ADODataset2.FilterOnBookmarks(BM);
    end else
      MessageDlg('Ни один товар не выбран!', mtError, [mbOK], 0);
    end;

    { Отключение фильтрации (очистка заказа) }

  procedure TForm2.SpeedButton2Click(Sender: TObject);
  begin
    ADODataset2.Filtered:= False;
    SetLength(BM, 0);
    ADODataset2.Close;
  end;

  { Добавление нового заказа в базу данных }

  procedure TForm2.Button1Click(Sender: TObject);
  var i,j,k: integer;
  begin
    if ADODataset3.FieldValues['Покупатель'] = '' then
    begin
      MessageDlg('Не указан покупатель!', mtError, [mbOK], 0);
      exit;
    end;

    Form1.ADODataset1.Close;
    if ADODataset2.Active then

```

```

begin
  ADODataSet2.First;
  i:=1;
  while i <= ADODataSet2.RecordCount do
    begin
      { Передача параметров в команды ADO }

      j:=ADODataSet2.FieldValues['Наличие на складе'];
      k:=ADODataSet2.FieldValues['Количество'];      { Заказано }
      if k>j then
        begin
          MessageDlg('В наличии только '+IntToStr(j)+' штук товара
''+ADODataSet2.FieldValues['Товар']+'"', mtError,[mbOK], 0);
          k:=j;
        end;
        if          j=0          then          MessageDlg('Товар
''+ADODataSet2.FieldValues['Товар']+''' отсутствует на складе!',
mtError,[mbOK], 0) else
          begin
            ADOCommand1.Parameters.ParamByName('OIDprm').Value      :=
StrToInt(Edit1.Text);
            ADOCommand1.Parameters.ParamByName('ODateprm').Value:=
Date;
            ADOCommand1.Parameters.ParamByName('Custprm').Value      :=
ADODataSet3.FieldValues['Код покупателя'];
            ADOCommand1.Parameters.ParamByName('PIDprm').Value      :=
ADODataSet2.FieldValues['Код товара'];
            ADOCommand1.Parameters.ParamByName('Quanprm').Value := k;
            ADOCommand2.Parameters.ParamByName('Quanprm').Value := k;
            ADOCommand2.Parameters.ParamByName('PIDprm').Value      :=
ADODataSet2.FieldValues['Код товара'];
            { Выполнение команд ADO (добавление новых записей в таблицу
заказов и корректировка значений полей таблицы товаров) }
            ADOCommand1.Execute;
            ADOCommand2.Execute;
          end;
          ADODataSet2.Next;
          i:=i+1;
        end;
        end
        else MessageDlg('Не выбраны товары для
заказа!', mtError,[mbOK], 0);
        Form1.ADODataSet1.Open;
        Close;
      end;

      { Открытие набора данных при отображении формы }

procedure TForm2.FormShow(Sender: TObject);
begin
  ADODataSet1.Open;

```

```

    ADODataSet3.Open;
end;

{ Заккрытие наборов данных при закрытии формы }

procedure TForm2.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
    ADODataSet1.Close;
    ADODataSet2.Close;
    ADODataSet3.Close;
    SetLength(BM, 0);
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
    Form3.ShowModal;
end;

procedure TForm2.Button3Click(Sender: TObject);
begin
    Form4.ShowModal;
end;

end.

// Код модуля для редакции записей таблицы "Заказы" (Orders)

unit Colume;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms,
    Dialogs, StdCtrls, DB, ADODB, Grids, DBGrids;

type
    TForm3 = class(TForm)
        Label4: TLabel;
        DBGrid1: TDBGrid;
        ADODataSet1: TADODataSet;
        DataSource1: TDataSource;
        Button1: TButton;
        Button2: TButton;
        Edit1: TEdit;
        Label1: TLabel;
        Edit2: TEdit;
        ADOCommand1: TADOCommand;
        ADOCommand2: TADOCommand;
        Button3: TButton;
    end;

```

```

    Button4: TButton;
    ADOCommand3: TADOCommand;
    ADOCommand4: TADOCommand;
    ADODataSet2: TADODataSet;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure DBGrid1MouseUp(Sender: TObject; Button:
TMouseButton;
        Shift: TShiftState; X, Y: Integer);
    procedure DBGrid1KeyUp(Sender: TObject; var Key: Word;
        Shift: TShiftState);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form3: TForm3;

implementation

uses MainBDADO, EditWin;

{$R *.dfm}

procedure TForm3.Button1Click(Sender: TObject);
var i,k:Integer; j:integer;
begin

    { Редактирование записи "Покупатель" }

    if ADODataSet1.FieldValues['Код покупателя']=0 then exit;
    if form3.Edit1.Text='' then
        begin
            MessageDlg('Заполнение поля "Покупатель" обязательно!',
mtError,[mbOK], 0);
            exit;
        end;

    i:=ADODataSet1.FieldValues['Код покупателя'];

    { Поиск повторяющейся записи поля "Покупатель" }

Запрос ADODataSet1:

```

```

SELECT ПокупательID AS [Код покупателя], Покупатель, [Адрес
покупателя]
FROM Customer
ORDER BY Покупатель

```

Запрос ADODataSet2:

```

SELECT ПокупательID, Покупатель
FROM Customer
WHERE Покупатель = :Custprm AND ПокупательID <> :CustIDprm

```

Запрос ADOCommand1:

```

UPDATE Customer SET Покупатель = :Cuprm, [Адрес покупателя] =
:Adrprm
WHERE ПокупательID = :CuIDprm

```

```

}

```

```

ADODataSet2.Parameters.ParamByName('CustIDprm').Value := i;
ADODataSet2.Parameters.ParamByName('Custprm').Value :=
form3.Edit1.Text;
ADODataSet2.Open;
j:=ADODataSet2.RecordCount;
ADODataSet2.Close;

```

```

if j>0 then
begin
    MessageDlg('Покупатель "'+form3.Edit1.Text+'" уже есть в базе
данных!', mtError,[mbOK], 0);
    exit;
end;

```

```

StrPCopy(ADF, 'Вы уверены, что необходимо редактировать запись
покупателя "'+ADODataSet1.FieldValues['Покупатель']+'"?');

```

```

if Application.MessageBox(ADF,'Внимание: выполняется операция
редактирования записи!',
    MB_OKCANCEL + MB_DEFBUTTON1) <> IDOK then exit;

```

```

ADOCommand1.Parameters.ParamByName('CuIDprm').Value :=
ADODataSet1.FieldValues['Код покупателя'];
ADOCommand1.Parameters.ParamByName('Cuprm').Value :=
form3.Edit1.Text;
ADOCommand1.Parameters.ParamByName('Adrprm').Value :=
form3.Edit2.Text;

```

```

ADODataSet1.Close;
form1.ADODataSet1.Close;
form2.ADODataSet3.Close;

```

```

{ Выполнение команды ADO }

ADOCommand1.Execute;

{ Обновление таблиц }

ADODataset1.Open;
form1.ADODataset1.Open;
form2.ADODataset3.Open;

{ Установка указателя на запись, равную i }

ADODataset1.First;
while not ADODataset1.Eof and (i<>ADODataset1.FieldValues['Код
покупателя']) do ADODataset1.Next;

end;

procedure TForm3.FormShow(Sender: TObject);
begin
  ADODataset1.Open;
  form3.Edit1.Text:=ADODataset1.FieldValues['Покупатель'];
  form3.Edit2.Text:=ADODataset1.FieldValues['Адрес покупателя'];
end;

procedure TForm3.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  ADODataset1.Close;
end;

procedure TForm3.Button2Click(Sender: TObject);
begin
  Close;
end;

procedure TForm3.Button3Click(Sender: TObject);
var j:integer;
begin

  { Новый покупатель }

  if form3.Edit1.Text='' then
  begin
    MessageDlg('Заполнение поля "Покупатель" обязательно!',
mtError,[mbOK], 0);
    exit;
  end;

  { Поиск повторяющейся записи поля "Покупатель" }

```

```

ADODataset2.Parameters.ParamByName('CustIDprm').Value := -1;
ADODataset2.Parameters.ParamByName('Custprm').Value :=
form3.Edit1.Text;
ADODataset2.Open;
j:=ADODataset2.RecordCount;
ADODataset2.Close;

if j>0 then
begin
    MessageDlg('Покупатель "'+form3.Edit1.Text+'" уже есть в базе
данных!', mtError,[mbOK], 0);
    exit;
end;

{
Запрос ADOCommand2:

INSERT INTO Customer(Покупатель, [Адрес покупателя])
VALUES(:Cuprm, :Adrprm)

}

ADODataset2.Parameters.ParamByName('Cuprm').Value :=
form3.Edit1.Text;
ADODataset2.Parameters.ParamByName('Adrprm').Value :=
form3.Edit2.Text;;

ADODataset1.Close;
form2.ADODataSet3.Close;

{ Выполнение команды ADO }

ADODataset2.Execute;

{ Обновление таблицы }

ADODataset1.Open;
form2.ADODataSet3.Open;

{ Установка указателя на последнюю запись }

ADODataset1.Last;

end;

procedure TForm3.Button4Click(Sender: TObject);
begin
{ Удаление связанных записей }

if ADODataset1.FieldValues['Код покупателя']=0 then exit;

```

```

StrPCopy(ADF, 'Вы уверены, что необходимо удалить все записи
покупателя "' + ADODataSet1.FieldValues['Покупатель'] + '"?');

if Application.MessageBox(ADF, 'Внимание: выполняется операция
удаления записей!',
    MB_OKCANCEL + MB_DEFBUTTON1) <> IDOK then exit;
{
Запрос ADOCommand3:

DELETE FROM Orders
WHERE ПокупательID = :CuIDdprm

Запрос ADOCommand4:

DELETE FROM Customer
WHERE ПокупательID = :CuIDdprm

}

ADOCommand3.Parameters.ParamByName('CuIDdprm').Value
:=ADODataSet1.FieldValues['Код покупателя'];
ADOCommand4.Parameters.ParamByName('CuIDdprm').Value
:=ADODataSet1.FieldValues['Код покупателя'];

ADODataSet1.Close;
form1.ADODataSet1.Close;
form2.ADODataSet3.Close;

{ Выполнение команд ADO }

ADOCommand3.Execute;
ADOCommand4.Execute;

{ Обновление таблиц }

ADODataSet1.Open;
form1.ADODataSet1.Open;
form2.ADODataSet3.Open;

end;

procedure TForm3.DBGrid1MouseUp(Sender: TObject; Button:
TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    form3.Edit1.Text:=ADODataSet1.FieldValues['Покупатель'];
    form3.Edit2.Text:=ADODataSet1.FieldValues['Адрес покупателя'];
end;

procedure TForm3.DBGrid1KeyUp(Sender: TObject; var Key: Word;

```



```

    Shift: TShiftState);
begin
    form3.Edit1.Text:=ADODataset1.FieldValues['Покупатель'];
    form3.Edit2.Text:=ADODataset1.FieldValues['Адрес покупателя'];
end;

// Код модуля для редактирования записей в таблице "Товары"
(Products)

unit Product;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms,
    Dialogs, ADODB, StdCtrls, DB, Grids, DBGrids;

type
    TForm4 = class(TForm)
        Label4: TLabel;
        DBGrid1: TDBGrid;
        ADODataset1: TADODataset;
        DataSource1: TDataSource;
        Button1: TButton;
        Button2: TButton;
        Edit1: TEdit;
        Label1: TLabel;
        Edit2: TEdit;
        ADOCommand1: TADOCommand;
        ADOCommand2: TADOCommand;
        Button3: TButton;
        Button4: TButton;
        ADOCommand3: TADOCommand;
        ADOCommand4: TADOCommand;
        Label2: TLabel;
        Label3: TLabel;
        Edit3: TEdit;
        Edit4: TEdit;
        Label5: TLabel;
        Label6: TLabel;
        Edit5: TEdit;
        Edit6: TEdit;
        ADODataset2: TADODataset;
        procedure FormShow(Sender: TObject);
        procedure FormClose(Sender: TObject; var Action:
TCloseAction);
        procedure Button2Click(Sender: TObject);
        procedure Button1Click(Sender: TObject);
        procedure DBGrid1MouseUp(Sender: TObject; Button:
TMouseButton;

```

```

        Shift: TShiftState; X, Y: Integer);
    procedure DBGrid1KeyUp(Sender: TObject; var Key: Word;
        Shift: TShiftState);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form4: TForm4;

implementation

uses MainBDADO, EditWin;

{$R *.dfm}

procedure TForm4.FormShow(Sender: TObject);
begin
    {

Запрос ADODataset1:

SELECT ТоварID AS [Код товара], Товар, Изготовитель, Модель,
[Технические характеристики], [Наличие на складе], Цена,
Количество
FROM Products
ORDER BY Товар, Изготовитель, Модель
}

    ADODataset1.Open;

    form4.Edit1.Text:=ADODataset1.FieldValues['Товар'];
    form4.Edit2.Text:=ADODataset1.FieldValues['Изготовитель'];
    form4.Edit3.Text:=ADODataset1.FieldValues['Модель'];
    form4.Edit4.Text:=ADODataset1.FieldValues['Технические
характеристики'];
    form4.Edit5.Text:=ADODataset1.FieldValues['Наличие на складе'];
    form4.Edit6.Text:=ADODataset1.FieldValues['Цена'];
end;

procedure TForm4.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
    ADODataset1.Close;
end;

```

```

procedure TForm4.Button2Click(Sender: TObject);
begin
  Close;
end;

procedure TForm4.Button1Click(Sender: TObject);
var i,j,k : integer; a,b:extended; ttt:ShortString;  i1:Int64;
begin

  { Редактирование записи Товар }

  if ADODataSet1.FieldValues['Код товара']=0 then exit;
  if form4.Edit3.Text='' then
  begin
    MessageDlg('Заполнение поля "Модель" обязательно!',
mtError,[mbOK], 0);
    exit;
  end;

  i1:=ADODataSet1.FieldValues['Код товара'];

  { Поиск повторяющейся записи поля "Модель"

Запрос ADODataSet2:

SELECT ТоварID, Модель
FROM Products
WHERE Модель = :Modprm AND ТоварID <> :ModIDprm

}

  ADODataSet2.Parameters.ParamByName('ModIDprm').Value := i1;
  ADODataSet2.Parameters.ParamByName('Modprm').Value :=
form4.Edit3.Text;
  ADODataSet2.Open;
  j:=ADODataSet2.RecordCount;
  ADODataSet2.Close;

  if j>0 then
  begin
    MessageDlg('Модель "'+form4.Edit3.Text+'" уже есть в базе
данных!', mtError,[mbOK], 0);
    exit;
  end;

  val(form4.Edit5.Text,i,k);

  if (k<>0) or (i<0) then i:=0;

  ttt:=form4.Edit6.Text;

```

```

for j:=1 to ORD(ttt[0]) do if ttt[j]=',' then ttt[j]:='.';

val(ttt,a,k);

if (k<>0) or (a<0) then a:=0;

StrPCopy(ADF, 'Вы уверены, что необходимо редактировать запись
модели "' + ADODataset1.FieldValues['Модель'] + '"?');

if Application.MessageBox(ADF, 'Внимание: выполняется операция
редактирования записи!',
    MB_OKCANCEL + MB_DEFBUTTON1) <> IDOK then exit;

{

Запрос ADODataset1:

SELECT ТоварID AS [Код товара], Товар, Изготовитель, Модель,
[Технические характеристики], [Наличие на складе], Цена,
Количество
FROM Products
ORDER BY Товар, Изготовитель, Модель

Запрос ADOCommand1:

UPDATE Products SET Товар = :Prodprm, Изготовитель = :Prprm,
Модель = :Modprm,
[Технические характеристики] = :Thprm, [Наличие на складе] =
:Nprm, Цена = :Znprm, Количество = 1
WHERE [Код товара] = :ProdIDprm

}

ADOCommand1.Parameters.ParamByName('ProdIDprm').Value      :=
ADODataset1.FieldValues['Код товара'];
ADOCommand1.Parameters.ParamByName('Prodprm').Value        :=
form4.Edit1.Text;
ADOCommand1.Parameters.ParamByName('Prprm').Value          :=
form4.Edit2.Text;
ADOCommand1.Parameters.ParamByName('Modprm').Value          :=
form4.Edit3.Text;
ADOCommand1.Parameters.ParamByName('Thprm').Value           :=
form4.Edit4.Text;
ADOCommand1.Parameters.ParamByName('Nprm').Value            := i;
ADOCommand1.Parameters.ParamByName('Znprm').Value           := a;

ADODataset1.Close;
form1.ADODataset1.Close;
form2.ADODataset1.Close;

{ Выполнение команды ADO }

```

```

ADOCCommand1.Execute;

{ Обновление таблиц }

ADODataset1.Open;
form1.ADODataSet1.Open;
form2.ADODataSet1.Open;

{ Установка указателя на запись, равную i1 }

ADODataset1.First;
while not ADODataset1.Eof and (i1<>ADODataset1.FieldValues['Код
товара']) do ADODataset1.Next;

end;

procedure TForm4.DBGrid1MouseUp(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  form4.Edit1.Text:=ADODataset1.FieldValues['Товар'];
  form4.Edit2.Text:=ADODataset1.FieldValues['Изготовитель'];
  form4.Edit3.Text:=ADODataset1.FieldValues['Модель'];
  form4.Edit4.Text:=ADODataset1.FieldValues['Технические
характеристики'];
  form4.Edit5.Text:=ADODataset1.FieldValues['Наличие на складе'];
  form4.Edit6.Text:=ADODataset1.FieldValues['Цена'];
end;

procedure TForm4.DBGrid1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  form4.Edit1.Text:=ADODataset1.FieldValues['Товар'];
  form4.Edit2.Text:=ADODataset1.FieldValues['Изготовитель'];
  form4.Edit3.Text:=ADODataset1.FieldValues['Модель'];
  form4.Edit4.Text:=ADODataset1.FieldValues['Технические
характеристики'];
  form4.Edit5.Text:=ADODataset1.FieldValues['Наличие на складе'];
  form4.Edit6.Text:=ADODataset1.FieldValues['Цена'];
end;

procedure TForm4.Button3Click(Sender: TObject);
var i,j,k : integer; a,b:extended; ttt:ShortString;
begin

  { Новый товар }

  if form4.Edit3.Text='' then
    begin

```

```

    MessageDlg('Заполнение поля "Модель" обязательно!',
mtError,[mbOK], 0);
    exit;
end;

{ Поиск повторяющейся записи поля "Модель"

Запрос ADODataSet2:

SELECT ТоварID, Модель
FROM Products
WHERE Модель = :Modprm AND ТоварID <> :ModIDprm

}

ADODataSet2.Parameters.ParamByName('ModIDprm').Value := -1;
ADODataSet2.Parameters.ParamByName('Modprm').Value :=
form4.Edit3.Text;
ADODataSet2.Open;
j:=ADODataSet2.RecordCount;
ADODataSet2.Close;

if j>0 then
begin
    MessageDlg('Модель "'+form4.Edit3.Text+'" уже есть в базе
данных!', mtError,[mbOK], 0);
    exit;
end;

val(form4.Edit5.Text,i,k);

if (k<>0) or (i<0) then i:=0;

ttt:=form4.Edit6.Text;

for j:=1 to ORD(ttt[0]) do if ttt[j]=',' then ttt[j]:='.';

val(ttt,a,k);

if (k<>0) or (a<0) then a:=0;

{
Запрос ADOCommand2:

INSERT INTO Products(Товар, Изготовитель, Модель, [Технические
характеристики] , [Наличие на складе], Цена, Количество)
VALUES(:Prodprm, :Prprm, :Modprm, :Thprm, :Nprm, :Znprm, :Colprm)
}
ADOCommand2.Parameters.ParamByName('Prodprm').Value :=
form4.Edit1.Text;

```

```

    ADOCommand2.Parameters.ParamByName('Prprm').Value           :=
form4.Edit2.Text;
    ADOCommand2.Parameters.ParamByName('Modprm').Value         :=
form4.Edit3.Text;
    ADOCommand2.Parameters.ParamByName('Thprm').Value          :=
form4.Edit4.Text;
    ADOCommand2.Parameters.ParamByName('Nprm').Value            := i;
    ADOCommand2.Parameters.ParamByName('Znprm').Value           := a;
    ADOCommand2.Parameters.ParamByName('Colprm').Value          := 1;

    ADODataset1.Close;
    form2.ADODataset1.Close;

    { Выполнение команды ADO }

    ADOCommand2.Execute;

    { Обновление таблицы }

    ADODataset1.Open;
    form2.ADODataset1.Open;

    { Установка указателя на последнюю запись }

    ADODataset1.Last;

end;

procedure TForm4.Button4Click(Sender: TObject);
begin
    { Удаление связанных записей }

    if ADODataset1.FieldValues['Код товара']=0 then exit;

    StrPCopy(ADF, 'Вы уверены, что необходимо удалить все записи
товара "'+ADODataset1.FieldValues['Товар']+"?");

    if Application.MessageBox(ADF,'Внимание: выполняется операция
удаления записей!',
        MB_OKCANCEL + MB_DEFBUTTON1) <> IDOK then exit;

    {
Запрос ADOCommand3:

DELETE FROM Orders
WHERE ТоварID= :CuIDdprm

Запрос ADOCommand4:

DELETE FROM Products

```

```

WHERE ТоварID= :CuIDdprm
}

ADOCommand3.Parameters.ParamByName('CuIDdprm').Value
:=ADODataset1.FieldValues['Код товара'];
ADOCommand4.Parameters.ParamByName('CuIDdprm').Value
:=ADODataset1.FieldValues['Код товара'];

ADODataset1.Close;
form1.ADODataSet1.Close;
form2.ADODataSet1.Close;

{ Выполнение команд ADO }

ADOCommand3.Execute;
ADOCommand4.Execute;

{ Обновление таблиц }

ADODataset1.Open;
form1.ADODataSet1.Open;
form2.ADODataSet1.Open;

end;

end.

// Код модуля для формирования отчетов

unit Reliz;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, Grids, DBGrids, DateUtils,
  ComCtrls;

type
  TForm5 = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Label5: TLabel;
    Label1: TLabel;
    Button2: TButton;
    StringGrid1: TStringGrid;
    StringGrid2: TStringGrid;
    ComboBox1: TComboBox;
    ADODataset1: TADODataset;
  end;

```



```

    ADODataSet2: TADODataSet;
    Button1: TButton;
    Button3: TButton;
    Label2: TLabel;
    Label3: TLabel;
    Button4: TButton;
    StringGrid3: TStringGrid;
    StringGrid4: TStringGrid;
    TabSheet3: TTabSheet;
    Label4: TLabel;
    StringGrid5: TStringGrid;
    Label6: TLabel;
    StringGrid6: TStringGrid;
    ADODataSet3: TADODataSet;
    ADODataSet4: TADODataSet;
    ADODataSet5: TADODataSet;
    ADODataSet6: TADODataSet;
    ADODataSet7: TADODataSet;
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form5          : TForm5;
    ttt,ttt1       : ShortString;
    k1,k2          : integer;
    A1             : array of Integer;

implementation

uses MainBDADO;

{$R *.dfm}

procedure Itogo;
var i,j,k,m:integer; y: extended;
    n : array [1..6] of integer;
    x : array [1..6] of extended;
begin

    { Заполнение графы "Итого" в form5.StringGrid1 и form5.StringGrid2
}

```

```

for j:=1 to 6 do
begin
  n[j]:=0;
  x[j]:=0;
end;

for j:=1 to 6 do
for i:=1 to form5.StringGrid1.RowCount-1 do
begin
  ttt:=form5.StringGrid1.Cells[j,i];
  Val(ttt,k,m);
  if m<>0 then k:=0;
  n[j]:=n[j]+k;

  ttt:=form5.StringGrid2.Cells[j,i];
  Val(ttt,y,m);
  if m<>0 then y:=0;
  x[j]:=x[j]+y;

end;

form5.StringGrid1.RowCount:=form5.StringGrid1.RowCount+1;
form5.StringGrid2.RowCount:=form5.StringGrid1.RowCount;
i:=form5.StringGrid1.RowCount-1;
form5.StringGrid1.Cells[0,i]:='Итого';
form5.StringGrid2.Cells[0,i]:='Итого';
for j:=1 to 6 do
begin

  if n[j]>0 then STR(n[j]:10,ttt)
    else ttt:='';
  form5.StringGrid1.Cells[j,i]:=ttt;

  if x[j]>0 then STR(x[j]:10:2,ttt)
    else ttt:='';
  form5.StringGrid2.Cells[j,i]:=ttt;
end;
end;

procedure Itogol;
var i,j,k,m:integer; y: extended;
    n : array [1..6] of integer;
    x : array [1..6] of extended;
begin

  { Заполнение графы "Итого" в form5.StringGrid3 и form5.StringGrid4
}

  for j:=1 to 6 do
    begin

```

```

    n[j]:=0;
    x[j]:=0;
end;

for j:=1 to 6 do
for i:=1 to form5.StringGrid3.RowCount-1 do
begin
    ttt:=form5.StringGrid3.Cells[j,i];
    Val(ttt,k,m);
    if m<>0 then k:=0;
    n[j]:=n[j]+k;

    ttt:=form5.StringGrid4.Cells[j,i];
    Val(ttt,y,m);
    if m<>0 then y:=0;
    x[j]:=x[j]+y;

end;

form5.StringGrid3.RowCount:=form5.StringGrid3.RowCount+1;
form5.StringGrid4.RowCount:=form5.StringGrid3.RowCount;
i:=form5.StringGrid3.RowCount-1;
form5.StringGrid3.Cells[0,i]:='Итого';
form5.StringGrid4.Cells[0,i]:='Итого';
for j:=1 to 6 do
begin

    if n[j]>0 then STR(n[j]:10,ttt)
        else ttt:='';
    form5.StringGrid3.Cells[j,i]:=ttt;

    if x[j]>0 then STR(x[j]:10:2,ttt)
        else ttt:='';
    form5.StringGrid4.Cells[j,i]:=ttt;
end;
end;

procedure Itogo2;
var i,j,k,m:integer; y: extended;
    n : array [1..6] of integer;
    x : array [1..6] of extended;
begin

{ Заполнение графы "Итого" в form5.StringGrid3 и form5.StringGrid4
}

for j:=1 to 6 do
begin
    n[j]:=0;
    x[j]:=0;

```

```

end;

for j:=1 to 6 do
for i:=1 to form5.StringGrid5.RowCount-1 do
begin
  ttt:=form5.StringGrid5.Cells[j,i];
  Val(ttt,k,m);
  if m<>0 then k:=0;
  n[j]:=n[j]+k;

  ttt:=form5.StringGrid6.Cells[j,i];
  Val(ttt,y,m);
  if m<>0 then y:=0;
  x[j]:=x[j]+y;

end;

form5.StringGrid5.RowCount:=form5.StringGrid5.RowCount+1;
form5.StringGrid6.RowCount:=form5.StringGrid5.RowCount;
i:=form5.StringGrid5.RowCount-1;
form5.StringGrid5.Cells[0,i]:='Итого';
form5.StringGrid6.Cells[0,i]:='Итого';
for j:=1 to 6 do
begin

  if n[j]>0 then STR(n[j]:10,ttt)
    else ttt:='';
  form5.StringGrid5.Cells[j,i]:=ttt;

  if x[j]>0 then STR(x[j]:10:2,ttt)
    else ttt:='';
  form5.StringGrid6.Cells[j,i]:=ttt;
end;
end;

procedure Zapros;
var i,k,m,n,Year:integer; x,y:extended; j:Int64; AObject: TObject;
begin

{ Для использования функции YearOf(Date) присоединить библиотеку
DateUtils или
использовать функцию DecodeDate(Date, Year, Month, Day: Word); }

Year:=YearOf(Date);

for i:=0 to form5.StringGrid1.RowCount-1 do
for k:=0 to form5.StringGrid1.ColCount-1 do
begin
  form5.StringGrid1.Cells[k,i]:='';
  form5.StringGrid2.Cells[k,i]:='';

```

```

    form5.StringGrid3.Cells[k,i]:='';
    form5.StringGrid4.Cells[k,i]:='';
end;

form5.StringGrid1.RowCount:=2;
form5.StringGrid2.RowCount:=2;
form5.StringGrid3.RowCount:=2;
form5.StringGrid4.RowCount:=2;

form5.StringGrid3.Cells[0,0]:='  Модель';
form5.StringGrid3.Cells[1,0]:='  Всего';
form5.StringGrid3.Cells[2,0]:=' '+IntToStr(Year)+' год';
form5.StringGrid3.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
form5.StringGrid3.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid3.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid3.Cells[6,0]:=' '+IntToStr(Year-4)+' год';

form5.StringGrid4.Cells[0,0]:='  Модель';
form5.StringGrid4.Cells[1,0]:='  Всего';
form5.StringGrid4.Cells[2,0]:=' '+IntToStr(Year)+' год';
form5.StringGrid4.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
form5.StringGrid4.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid4.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid4.Cells[6,0]:=' '+IntToStr(Year-4)+' год';

{
Запрос ADODataset2:

SELECT ТоварID, Модель, Цена
FROM Products
ORDER BY Модель
}

form5.ADODataset2.Open;

k2:=form5.ADODataset2.RecordCount+1;
SetLength(A1,k2);

i:=0;
form5.ComboBox1.Clear;
form5.ADODataset2.First;
while not form5.ADODataset2.Eof do
begin
    i:=i+1;
    if form5.StringGrid1.RowCount<=i then
    begin
        form5.StringGrid3.RowCount:=i+1;
        form5.StringGrid4.RowCount:=i+1;
    end;
end;

```

```

ttd:=form5.ADODataSet2.FieldValues['Модель'];
form5.ComboBox1.AddItem(ttd,AObject);

form5.StringGrid3.Cells[0,i]:=ttd;
form5.StringGrid4.Cells[0,i]:=ttd;

A1[i]:=form5.ADODataSet2.FieldValues['ТоварID'];

form5.ADODataSet2.Next;
end;

form5.ComboBox1.ItemIndex:=0;

form5.ADODataSet2.Close;

form5.StringGrid1.Cells[0,0]:=' Покупатель';
form5.StringGrid1.Cells[1,0]:=' Всего';
form5.StringGrid1.Cells[2,0]:=' '+IntToStr(Year)+' год';
form5.StringGrid1.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
form5.StringGrid1.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid1.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid1.Cells[6,0]:=' '+IntToStr(Year-4)+' год';

form5.StringGrid2.Cells[0,0]:=' Покупатель';
form5.StringGrid2.Cells[1,0]:=' Всего';
form5.StringGrid2.Cells[2,0]:=' '+IntToStr(Year)+' год';
form5.StringGrid2.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
form5.StringGrid2.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid2.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid2.Cells[6,0]:=' '+IntToStr(Year-4)+' год';
n:=Year;

{

Запрос ADODataSet1:

SELECT RecID, Номер, [Дата заказа], b.ПокупательID, Покупатель,
b.ТоварID, Модель, Цена, Получено
FROM Products a, Orders b, Customer c
WHERE a.ТоварID=b.ТоварID AND b.ПокупательID = c.ПокупательID
ORDER BY Покупатель

}

i:=0;
form5.ADODataSet1.First;
j:=-1;
while not form5.ADODataSet1.Eof do
begin

```

```

{ Вывод покупателя }

if j<>form5.ADODataSet1.FieldValues['ПокупательID'] then
begin
  i:=i+1;
  if form5.StringGrid1.RowCount<=i then
  begin
    form5.StringGrid1.RowCount:=i+1;
    form5.StringGrid2.RowCount:=i+1;
  end;

form5.StringGrid1.Cells[0,i]:=form5.ADODataSet1.FieldValues['Покупатель'];

form5.StringGrid2.Cells[0,i]:=form5.ADODataSet1.FieldValues['Покупатель'];
  j:=form5.ADODataSet1.FieldValues['ПокупательID'];
end;

Year:=n-YearOf(form5.ADODataSet1.FieldValues['Дата заказа'])+2;

{ Число закупок }

Val(form5.StringGrid1.Cells[Year,i],k,m);
if m<>0 then k:=0;

k:=k+form5.ADODataSet1.FieldValues['Получено'];
STR(k:10,ttt);
form5.StringGrid1.Cells[Year,i]:=ttt;

Val(form5.StringGrid1.Cells[1,i],k,m);
if m<>0 then k:=0;

k:=k+form5.ADODataSet1.FieldValues['Получено'];
STR(k:10,ttt);
form5.StringGrid1.Cells[1,i]:=ttt;

{ Сумма закупок }

Val(form5.StringGrid2.Cells[Year,i],x,m);
if m<>0 then x:=0;

y:=form5.ADODataSet1.FieldValues['Получено']*form5.ADODataSet1.FieldValues['Цена'];
x:=x+y;
STR(x:10:2,ttt);
form5.StringGrid2.Cells[Year,i]:=ttt;

```

```

Val(form5.StringGrid2.Cells[1,i],x,m);
if m<>0 then x:=0;

x:=x+y;
STR(x:10:2,ttt);
form5.StringGrid2.Cells[1,i]:=ttt;

        { По моделям }

{ Поиск номера строки в динамическом массиве }

m:=form5.ADODataSet1.FieldValues['ТоварID'];

for k1:=1 to k2 do if A1[k1]=m then Break;

{ Число закупок }

Val(form5.StringGrid3.Cells[Year,k1],k,m);
if m<>0 then k:=0;

k:=k+form5.ADODataSet1.FieldValues['Получено'];
STR(k:10,ttt);
form5.StringGrid3.Cells[Year,k1]:=ttt;

Val(form5.StringGrid3.Cells[1,k1],k,m);
if m<>0 then k:=0;

k:=k+form5.ADODataSet1.FieldValues['Получено'];
STR(k:10,ttt);
form5.StringGrid3.Cells[1,k1]:=ttt;

{ Сумма закупок }

Val(form5.StringGrid4.Cells[Year,k1],x,m);
if m<>0 then x:=0;

x:=x+y;
STR(x:10:2,ttt);
form5.StringGrid4.Cells[Year,k1]:=ttt;

Val(form5.StringGrid4.Cells[1,k1],x,m);
if m<>0 then x:=0;

x:=x+y;
STR(x:10:2,ttt);
form5.StringGrid4.Cells[1,k1]:=ttt;

form5.ADODataSet1.Next;
end;
Itogo;
Itogol;

```



```

end;

procedure Zaprosl;
var i,k,m,n,Year:integer; x,y:extended; j:Int64; AObject: TObject;
begin

{ Для использования функции YearOf(Date) присоединить библиотеку
DateUtils или
использовать функцию DecodeDate(Date, Year, Month, Day: Word); }

Year:=YearOf(Date);

for i:=0 to form5.StringGrid1.RowCount-1 do
for k:=0 to form5.StringGrid1.ColCount-1 do
begin
form5.StringGrid1.Cells[k,i]:='';
form5.StringGrid2.Cells[k,i]:='';
end;

form5.StringGrid1.RowCount:=2;
form5.StringGrid2.RowCount:=2;

form5.StringGrid1.Cells[0,0]:=' Покупатель';
form5.StringGrid1.Cells[1,0]:=' Всего';
form5.StringGrid1.Cells[2,0]:=' '+IntToStr(Year)+' год';
form5.StringGrid1.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
form5.StringGrid1.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid1.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid1.Cells[6,0]:=' '+IntToStr(Year-4)+' год';

form5.StringGrid2.Cells[0,0]:=' Покупатель';
form5.StringGrid2.Cells[1,0]:=' Всего';
form5.StringGrid2.Cells[2,0]:=' '+IntToStr(Year)+' год';
form5.StringGrid2.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
form5.StringGrid2.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid2.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid2.Cells[6,0]:=' '+IntToStr(Year-4)+' год';
n:=Year;

{

Запрос ADODataset1:

SELECT RecID, Номер, [Дата заказа], b.ПокупательID, Покупатель,
b.ТоварID, Модель, Цена, Получено
FROM Products a, Orders b, Customer c
WHERE a.ТоварID=b.ТоварID AND b.ПокупательID = c.ПокупательID
ORDER BY Покупатель

}

```

```

i:=0;
form5.ADODataSet1.First;
j:=-1;
while not form5.ADODataSet1.Eof do
begin

    { Вывод покупателя }

    if j<>form5.ADODataSet1.FieldValues['ПокупательID'] then
    begin
        i:=i+1;
        if form5.StringGrid1.RowCount<=i then
        begin
            form5.StringGrid1.RowCount:=i+1;
            form5.StringGrid2.RowCount:=i+1;
        end;

form5.StringGrid1.Cells[0,i]:=form5.ADODataSet1.FieldValues['Покупатель'];

form5.StringGrid2.Cells[0,i]:=form5.ADODataSet1.FieldValues['Покупатель'];
        j:=form5.ADODataSet1.FieldValues['ПокупательID'];
    end;

    if
form5.ADODataSet1.FieldValues['Модель']=form5.ComboBox1.Text
then
    begin
        Year:=n-YearOf(form5.ADODataSet1.FieldValues['Дата заказа'])+2;

        { Число закупок }

        Val(form5.StringGrid1.Cells[Year,i],k,m);
        if m<>0 then k:=0;

        k:=k+form5.ADODataSet1.FieldValues['Получено'];
        STR(k:10,ttt);
        form5.StringGrid1.Cells[Year,i]:=ttt;

        Val(form5.StringGrid1.Cells[1,i],k,m);
        if m<>0 then k:=0;

        k:=k+form5.ADODataSet1.FieldValues['Получено'];
        STR(k:10,ttt);
        form5.StringGrid1.Cells[1,i]:=ttt;

        { Сумма закупок }

        Val(form5.StringGrid2.Cells[Year,i],x,m);

```

```

        if m<>0 then x:=0;

y:=form5.ADODataSet1.FieldValues['Получено']*form5.ADODataSet1.FieldValues['Цена'];
        x:=x+y;
        STR(x:10:2,ttt);
        form5.StringGrid2.Cells[Year,i]:=ttt;

        Val(form5.StringGrid2.Cells[1,i],x,m);
        if m<>0 then x:=0;

        x:=x+y;
        STR(x:10:2,ttt);
        form5.StringGrid2.Cells[1,i]:=ttt;

    end;

    form5.ADODataSet1.Next;
end;
Itogo;
end;

procedure Zapros2;
var i,j,k,m,n,Year:integer; x:extended;
begin

    Year:=YearOf(Date);

    for i:=0 to form5.StringGrid5.RowCount-1 do
    for k:=0 to form5.StringGrid5.ColCount-1 do
    begin
        form5.StringGrid5.Cells[k,i]:='';
        form5.StringGrid6.Cells[k,i]:='';
    end;

    form5.StringGrid5.RowCount:=2;
    form5.StringGrid6.RowCount:=2;

    form5.StringGrid5.Cells[0,0]:=' Покупатель';
    form5.StringGrid5.Cells[1,0]:=' Всего';
    form5.StringGrid5.Cells[2,0]:=' '+IntToStr(Year)+' год';
    form5.StringGrid5.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
    form5.StringGrid5.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
    form5.StringGrid5.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
    form5.StringGrid5.Cells[6,0]:=' '+IntToStr(Year-4)+' год';

    form5.StringGrid6.Cells[0,0]:=' Покупатель';
    form5.StringGrid6.Cells[1,0]:=' Всего';
    form5.StringGrid6.Cells[2,0]:=' '+IntToStr(Year)+' год';
    form5.StringGrid6.Cells[3,0]:=' '+IntToStr(Year-1)+' год';

```

```

form5.StringGrid6.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid6.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid6.Cells[6,0]:=' '+IntToStr(Year-4)+' год';

{

Запрос ADODataset5:

SELECT  Покупатель, SUM(Получено) AS Получено, SUM(Получено*Цена)
AS Сумма
FROM Orders a, Products b, Customer c
WHERE a.ПокупательID = c.ПокупательID
AND a.ТоварID = b.ТоварID
GROUP BY Покупатель

}

form5.ADODataSet5.Open;
i:=0;
form5.ADODataSet5.First;
while not form5.ADODataSet5.Eof do
begin

    { Вывод покупателя хотя бы с одной покупкой }

    i:=i+1;

    if form5.StringGrid5.RowCount<=i then
    begin
        form5.StringGrid5.RowCount:=i+1;
        form5.StringGrid6.RowCount:=i+1;
    end;

form5.StringGrid5.Cells[0,i]:=form5.ADODataSet5.FieldValues['Покупатель'];

form5.StringGrid6.Cells[0,i]:=form5.ADODataSet5.FieldValues['Покупатель'];

    form5.ADODataSet5.Next;
end;

form5.ADODataSet5.Close;

{

Запрос ADODataset3:

```

```

SELECT  Покупатель, SUM(Получено) AS Получено, SUM(Получено*Цена)
AS Сумма
FROM Orders a, Products b, Customer c
WHERE a.ПокупательID = c.ПокупательID
AND a.ТоварID = b.ТоварID
AND b.Модель = :Prodprm
GROUP BY  Покупатель
}

```

```

form5.ADODataSet3.Parameters.ParamByName('Prodprm').Value:=form5.C
omboBox1.Text;
form5.ADODataSet3.Open;
form5.ADODataSet3.First;
while not form5.ADODataSet3.Eof do
begin

    ttt:=form5.ADODataSet3.FieldValues['Покупатель'];

    for n:=1 to i do if ttt = form5.StringGrid5.Cells[0,n] then
Break;

    { Вывод "Всего получено" }

    m:=form5.ADODataSet3.FieldValues['Получено'];
    STR(m:10,ttt);
    form5.StringGrid5.Cells[1,n]:=ttt;

    x:=form5.ADODataSet3.FieldValues['Сумма'];
    STR(x:10:2,ttt);
    form5.StringGrid6.Cells[1,n]:=ttt;

    form5.ADODataSet3.Next;
end;

form5.ADODataSet3.Close;

{

```

Запрос ADODataSet4:

```

SELECT      Покупатель,      SUM(a.Получено)      AS      Получено,
SUM(a.Получено*Цена) AS Сумма
FROM Orders a, Products b, Customer c
WHERE a.ПокупательID = c.ПокупательID
AND a.ТоварID=b.ТоварID
AND b.Модель = :Prodprm
AND YEAR([Дата заказа]) = :Yearprm
GROUP BY  Покупатель

```

```

}

For j:=0 to 4 do
begin

form5.ADODataSet4.Parameters.ParamByName('Prodprm').Value:=form5.C
omboBox1.Text;

form5.ADODataSet4.Parameters.ParamByName('Yearprm').Value:=Year-j;
form5.ADODataSet4.Open;
k:=2+j;
form5.ADODataSet4.First;
while not form5.ADODataSet4.Eof do
begin

ttt:=form5.ADODataSet4.FieldValues['Покупатель'];

for n:=1 to i do if ttt = form5.StringGrid5.Cells[0,n] then
Break;

{ Вывод "Получено по годам" }

m:=form5.ADODataSet4.FieldValues['Получено'];
STR(m:10,ttt);
form5.StringGrid5.Cells[k,n]:=ttt;

x:=form5.ADODataSet4.FieldValues['Сумма'];
STR(x:10:2,ttt);
form5.StringGrid6.Cells[k,n]:=ttt;

form5.ADODataSet4.Next;
end;
form5.ADODataSet4.Close;
end;

Itogo2;
end;

procedure Zapro3;
var i,j,k,m,n,Year:integer; x:extended;
begin

Year:=YearOf(Date);

for i:=0 to form5.StringGrid5.RowCount-1 do
for k:=0 to form5.StringGrid5.ColCount-1 do
begin
form5.StringGrid5.Cells[k,i]:='';
form5.StringGrid6.Cells[k,i]:='';
end;
end;

```

```

form5.StringGrid5.RowCount:=2;
form5.StringGrid6.RowCount:=2;

form5.StringGrid5.Cells[0,0]:=' Покупатель';
form5.StringGrid5.Cells[1,0]:=' Всего';
form5.StringGrid5.Cells[2,0]:=' '+IntToStr(Year)+' год';
form5.StringGrid5.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
form5.StringGrid5.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid5.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid5.Cells[6,0]:=' '+IntToStr(Year-4)+' год';

form5.StringGrid6.Cells[0,0]:=' Покупатель';
form5.StringGrid6.Cells[1,0]:=' Всего';
form5.StringGrid6.Cells[2,0]:=' '+IntToStr(Year)+' год';
form5.StringGrid6.Cells[3,0]:=' '+IntToStr(Year-1)+' год';
form5.StringGrid6.Cells[4,0]:=' '+IntToStr(Year-2)+' год';
form5.StringGrid6.Cells[5,0]:=' '+IntToStr(Year-3)+' год';
form5.StringGrid6.Cells[6,0]:=' '+IntToStr(Year-4)+' год';

{

Запрос ADODataset5:

SELECT Покупатель, SUM(Получено) AS Получено, SUM(Получено*Цена)
AS Сумма
FROM Orders a, Products b, Customer c
WHERE a.ПокупательID = c.ПокупательID
AND a.ТоварID = b.ТоварID
GROUP BY Покупатель

}

form5.ADODataSet5.Open;
i:=0;
form5.ADODataSet5.First;
while not form5.ADODataSet5.Eof do
begin

    { Вывод покупателя хотя бы с одной покупкой }

    i:=i+1;

    if form5.StringGrid5.RowCount<=i then
    begin
        form5.StringGrid5.RowCount:=i+1;
        form5.StringGrid6.RowCount:=i+1;
    end;

```

```

form5.StringGrid5.Cells[0,i]:=form5.ADODataSet5.FieldValues['Покупатель'];

form5.StringGrid6.Cells[0,i]:=form5.ADODataSet5.FieldValues['Покупатель'];

    form5.ADODataSet5.Next;
end;

form5.ADODataSet5.Close;

{
Запрос ADODataSet6:

SELECT  Покупатель, SUM(Получено) AS Получено, SUM(Получено*Цена)
AS Сумма
FROM Orders a, Products b, Customer c
WHERE a.ПокупательID = c.ПокупательID
AND a.ТоварID = b.ТоварID
GROUP BY Покупатель
}

form5.ADODataSet6.Open;
form5.ADODataSet6.First;
while not form5.ADODataSet6.Eof do
    begin

        ttt:=form5.ADODataSet6.FieldValues['Покупатель'];

        for n:=1 to i do if ttt = form5.StringGrid5.Cells[0,n] then
Break;

        { Вывод "Всего получено" }

        m:=form5.ADODataSet6.FieldValues['Получено'];
        STR(m:10,ttt);
        form5.StringGrid5.Cells[1,n]:=ttt;

        x:=form5.ADODataSet6.FieldValues['Сумма'];
        STR(x:10:2,ttt);
        form5.StringGrid6.Cells[1,n]:=ttt;

        form5.ADODataSet6.Next;
end;

form5.ADODataSet6.Close;

```



```

{
Запрос ADODataset7:

SELECT      Покупатель,      SUM(a.Получено)      AS      Получено,
SUM(a.Получено*Цена) AS Сумма
FROM Orders a, Products b, Customer c
WHERE a.ПокупательID = c.ПокупательID
AND a.ТоварID=b.ТоварID
AND YEAR([Дата заказа]) = :Yearprm
GROUP BY Покупатель
}

For j:=0 to 4 do
begin

form5.ADODataSet7.Parameters.ParamByName('Yearprm').Value:=Year-j;
form5.ADODataSet7.Open;
k:=2+j;
form5.ADODataSet7.First;
while not form5.ADODataSet7.Eof do
begin

ttt:=form5.ADODataSet7.FieldValues['Покупатель'];

for n:=1 to i do if ttt = form5.StringGrid5.Cells[0,n] then
Break;

{ Вывод "Получено по годам" }

m:=form5.ADODataSet7.FieldValues['Получено'];
STR(m:10,ttt);
form5.StringGrid5.Cells[k,n]:=ttt;

x:=form5.ADODataSet7.FieldValues['Сумма'];
STR(x:10:2,ttt);
form5.StringGrid6.Cells[k,n]:=ttt;

form5.ADODataSet7.Next;
end;
form5.ADODataSet7.Close;
end;

Itogo2;
end;

procedure TForm5.FormShow(Sender: TObject);
var Year:integer;
begin

```

```

form5.PageControl1.ActivePageIndex:=0;

{ Настройка параметров StringGrid }

form5.StringGrid1.Font.Name:='Courier New';
form5.StringGrid1.Font.Size:=9;
form5.StringGrid1.FixedCols:=0;
form5.StringGrid1.DefaultColWidth:=75;
form5.StringGrid1.ColWidths[0]:=155;

form5.StringGrid1.Options:=[goFixedVertLine,goFixedHorzLine,goVert
Line,goHorzLine,goRangeSelect,goColSizing];

form5.StringGrid2.Font.Name:='Courier New';
form5.StringGrid2.Font.Size:=9;
form5.StringGrid2.FixedCols:=0;
form5.StringGrid2.DefaultColWidth:=75;
form5.StringGrid2.ColWidths[0]:=155;

form5.StringGrid2.Options:=[goFixedVertLine,goFixedHorzLine,goVert
Line,goHorzLine,goRangeSelect,goColSizing];

form5.StringGrid3.Font.Name:='Courier New';
form5.StringGrid3.Font.Size:=9;
form5.StringGrid3.FixedCols:=0;
form5.StringGrid3.DefaultColWidth:=75;
form5.StringGrid3.ColWidths[0]:=155;

form5.StringGrid3.Options:=[goFixedVertLine,goFixedHorzLine,goVert
Line,goHorzLine,goRangeSelect,goColSizing];

form5.StringGrid4.Font.Name:='Courier New';
form5.StringGrid4.Font.Size:=9;
form5.StringGrid4.FixedCols:=0;
form5.StringGrid4.DefaultColWidth:=75;
form5.StringGrid4.ColWidths[0]:=155;

form5.StringGrid4.Options:=[goFixedVertLine,goFixedHorzLine,goVert
Line,goHorzLine,goRangeSelect,goColSizing];

form5.StringGrid5.Font.Name:='Courier New';
form5.StringGrid5.Font.Size:=9;
form5.StringGrid5.FixedCols:=0;
form5.StringGrid5.DefaultColWidth:=75;
form5.StringGrid5.ColWidths[0]:=155;

form5.StringGrid5.Options:=[goFixedVertLine,goFixedHorzLine,goVert
Line,goHorzLine,goRangeSelect,goColSizing];

form5.StringGrid6.Font.Name:='Courier New';
form5.StringGrid6.Font.Size:=9;

```

```

form5.StringGrid6.FixedCols:=0;
form5.StringGrid6.DefaultColWidth:=75;
form5.StringGrid6.ColWidths[0]:=155;

form5.StringGrid6.Options:=[goFixedVertLine,goFixedHorzLine,goVert
Line,goHorzLine,goRangeSelect,goColSizing];

ADODataset1.Open;

Zapros;

Zapros3;

end;

procedure TForm5.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  ADODataset1.Close;
  ADODataset3.Close;
end;

procedure TForm5.Button2Click(Sender: TObject);
begin
  Close;
end;

procedure TForm5.Button1Click(Sender: TObject);
var i,j:integer;
begin
  Zapros;
  Zapros3;
end;

procedure TForm5.Button3Click(Sender: TObject);
begin
  Zapros1;
  Zapros2;
end;

end.

```

Представленный листинг снабжен достаточно подробными комментариями, поэтому в детальном его описании необходимости не возникает. Имеет смысл только дополнительно пояснить несколько моментов.

Генерация номера нового заказа производится путем отыскания максимального номера среди уже существующих заказов и последующего увеличения его на единицу. Для поиска максимального номера заказа в таблице выполняется сортировка по полю Номер и переход к последней записи

(именно она и будет содержать заказ с максимальным номером). После этого осуществляется возврат к прежнему порядку сортировки, а курсор устанавливается на запись, которая была текущей до выполнения поиска.

Порядок сортировки записей определяется свойством *Sort*, где задается список полей, по которым выполняется упорядочивание записей.

Фильтрация записей в наборе данных *ADODataset2* второй формы выполняется методом *FilterOnBookmarks*. Предварительно пользователь отмечает интересующие его записи в сетке *DBGrid1*, связанной с набором данных *ADODataset1*, который, так же как и *ADODataset2*, строится по записям таблицы *Products*. После нажатия кнопки [↓] в массив *BM* передаются закладки, созданные в свойстве *SelectedRows* компонента *DBGrid1*. Затем этот массив используется в качестве параметра метода *FilterOnBookmarks*, который и выполняет фильтрацию по закладкам. В результате этого в набор данных *ADODataset2* включаются (и отображаются в сетке *DBGrid2*) только те записи, которые были выбраны пользователем в общем списке.

Добавление и удаление записей производится при помощи компонентов команд *ADOCommand*, в которых реализованы соответствующие запросы SQL. Передача параметров в эти команды осуществляется по их именам через подсвойство *ParamByName* свойства *Parameters* соответствующих компонентов *ADOCommand*. Выполнение запросов производится методом *Execute*.

Таким образом, в процессе реализации данного примера было создано приложение, которое позволяет вести простейший учет заказов (добавление, удаление, просмотр и сортировку) в магазине или торговой фирме.

Лабораторная работа № 2

ИССЛЕДОВАНИЕ РАБОТЫ МУЛЬТИЗАДАЧНЫХ ПРИЛОЖЕНИЙ

Цель работы

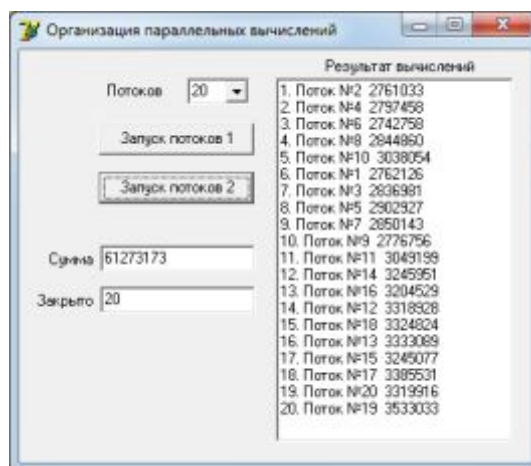
Исследовать работу мультизадачного приложения.

Указания к работе

1. Разработать мультизадачное приложение по предлагаемому образцу.
2. Выполнить исследование работы мультизадачного приложения.
3. Оформить отчет.

Исследование работы мультизадачного приложения

В данной лабораторной работе демонстрируется способ синхронизации потоков с помощью критических секций. Приложение позволяет запускать до 20 потоков, которые за отведенное им время должны выполнить какое-то количество операций инкрементации. Для вывода результатов инкрементации потоков на главную форму служит соответствующая процедура. При отсутствии критических секций в потоках они начинают конфликтовать за общий ресурс, в результате чего может произойти потеря данных. Результат работы приложения показан на рисунке:



```
unit MainPotok;  
  
interface  
  
uses  
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  
    Forms,  
    Dialogs, StdCtrls;  
  
type
```

```

TForm1 = class(TForm)
  ComboBox1: TComboBox;
  Label1: TLabel;
  Edit1: TEdit;
  Button4: TButton;
  Edit2: TEdit;
  Memo1: TMemo;
  Label8: TLabel;
  Label9: TLabel;
  Label13: TLabel;
  Button5: TButton;
  procedure FormCreate(Sender: TObject);
  procedure Button4Click(Sender: TObject);
  procedure Button5Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

TMessThread1 = class(TThread)
private
  NN:integer;
protected
  procedure Execute; override;
public
  constructor Create(Number:integer);
end;

TMessThread2 = class(TThread)
private
  NN:integer;
protected
  procedure Execute; override;
public
  constructor Create(Number:integer);
end;

const
  TimStop=5;
  NThread=20;

var
  Form1      : TForm1;
  Start      : boolean;
  DoneFlags  : integer;
  Sum        : Int64;
  Threads1   : array [1..NThread] of TMessThread1;
  Threads2   : array [1..NThread] of TMessThread2;
  CS         : TRTLCriticalSection;

implementation

{$R *.dfm}

{$R+ }

```

```

procedure Nuli;
begin
  form1.Edit1.Text:='0';
  form1.Edit2.Text:='0';
  form1.Memo1.Clear;
  DoneFlags:=0;
  Sum:=0;
  form1.Refresh;
end;

procedure ThreadsDone1(k:Int64;NN:integer);
begin
  inc(DoneFlags);
  form1.Edit2.Text:=IntToStr(DoneFlags);
  Start:=DoneFlags>=form1.ComboBox1.ItemIndex+1;
  form1.Memo1.Lines.Add(IntToStr(DoneFlags)+' . Поток №'+IntToStr(NN)+' '+IntToStr(k));
  Sum:=Sum+k;
  form1.Edit1.Text:=IntToStr(Sum);
end;

procedure ThreadsDone2(k:Int64; NN:integer);
begin
  inc(DoneFlags);
  form1.Edit2.Text:=IntToStr(DoneFlags);
  Start:=DoneFlags>=form1.ComboBox1.ItemIndex+1;
  form1.Memo1.Lines.Add(IntToStr(DoneFlags)+' . Поток №'+IntToStr(NN)+' '+IntToStr(k));
  Sum:=Sum+k;
  form1.Edit1.Text:=IntToStr(Sum);
  if Start then DeleteCriticalSection(CS);
end;

constructor TMessThread1.Create(Number:integer);
begin
  NN := Number;
  inherited Create(False);
end;

procedure TMessThread1.Execute;
var i      : Int64;
    Tim,Tim1 : Double;
    Hour,Min,Sec,MSec : Word;
begin
  Tim:=Time;
  Tim1 } { Текущее системное время в
  Hour:= 0;      { 0..23 }
  Min := 0;      { 0..59 }
  Sec := TimStop; { 0..59 }
  MSec:= 0;      { 0..999 }
  Tim1:=Tim+EncodeTime(Hour,Min,Sec,MSec);
  i:=0;
  repeat
    i:=i+1;

```

```

    Tim:=Time;                                { Текущее системное время в
Tim }
    until Tim>=Tim1;
    ThreadsDone1(i,NN);
end;

constructor TMessThread2.Create(Number:integer);
begin
    NN := Number;
    inherited Create(False);
end;

procedure TMessThread2.Execute;
var i      : Int64;
    Tim,Tim1 : Double;
    Hour,Min,Sec,MSec : Word;
begin
    Tim:=Time;                                { Текущее системное время в
Tim1 }
    Hour:= 0;                                { 0..23 }
    Min := 0;                                { 0..59 }
    Sec := TimStop;                          { 0..59 }
    MSec:= 0;                                { 0..999 }
    Tim1:=Tim+EncodeTime(Hour,Min,Sec,MSec);
    i:=0;
    repeat
        i:=i+1;
        Tim:=Time;                            { Текущее системное время в
Tim }
        until Tim>=Tim1;
        EnterCriticalSection(CS);
        ThreadsDone2(i,NN);
        LeaveCriticalSection(CS);
    end;

procedure TForm1.FormCreate(Sender: TObject);
var i:integer;
begin
    Start:=true;
    Nuli;

    for i:=1 to NThread do form1.ComboBox1.Items.Add(IntToStr(i));
    form1.ComboBox1.ItemIndex:=NThread-1;

end;

procedure TForm1.Button4Click(Sender: TObject);
var i:integer;
begin
    if not Start then exit; { Защита от двойного клика }
    Start := false;
    Nuli;

    for i:=1 to form1.ComboBox1.ItemIndex+1 do
    Threads1[i]:=TMessThread1.Create(i);
end;

```



```

procedure TForm1.Button5Click(Sender: TObject);
var i:integer;
begin
  if not Start then exit; { Защита от двойного клика }
  Start := false;

  Nuli;

  InitializeCriticalSection(CS);

  for i:=1 to form1.ComboBox1.ItemIndex+1 do
    Threads2[i]:=TMessThread2.Create(i);
  end;

end.

```

Лабораторная работа № 3

ИССЛЕДОВАНИЕ РАБОТЫ МУЛЬТИПРОГРАММНЫХ ПРИЛОЖЕНИЙ

Цель работы

Разработать программу на Turbo-Delphi для демонстрации мультипрограммного приложения, организованного по принципу «хозяин – работник (master – slave)» и исследовать его работу.

Указания к работе

1. Используя среду Turbo-Delphi, разработать приложение-«хозяин» (программа Zap).
2. Используя среду Turbo-Delphi разработать приложение-«работник» (программа Zadatcha).
3. Отладить организацию мультипрограммных вычислений по принципу «хозяин – работник».
4. Оформить отчет.

Листинги программ

```
unit Zapusk;  
  
interface  
  
uses  
    Windows, Messages, SysUtils, Variants, Classes, Graphics,  
    Controls, Forms,  
    Dialogs, StdCtrls, ExtCtrls;  
  
type  
    TForm1 = class(TForm)  
        Button1: TButton;  
        Timer1: TTimer;  
        Edit1: TEdit;  
        Label1: TLabel;  
        Edit2: TEdit;  
        Label2: TLabel;  
        Edit3: TEdit;  
        Label3: TLabel;  
        Button2: TButton;  
        Button3: TButton;  
        Button4: TButton;  
        Button5: TButton;  
        Button6: TButton;  
        Memo1: TMemo;
```

```

Label4: TLabel;
Button7: TButton;
Button8: TButton;
Button9: TButton;
procedure Button1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure WMCopyData(var Msg: TWMCopyData); message
WM_COPYDATA;
procedure Button7Click(Sender: TObject);
procedure Edit3Db1Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  Tim,Tim1,Tim2,Tim3,Tim4 : Double;      { Переменные времени.
Отсчет системного времени начинается с 30.12.1899 0:0:0:0 }
  Start                      : Boolean;   { Логические переменные }
  k,n                        : integer;   { Число запущенных
процессов и сообщений }
  ttt                        : ShortString; { Рабочая строковая }
  ADF                        : array [0..255] of Char;

implementation

{$R *.dfm}

Procedure execWait(const ComLine:String);
Var Si : TStartupInfo;
    p : TProcessInformation;
Begin

  FillChar(Si,SizeOf(Si),0);
  With Si Do
    Begin
      cb:=SizeOf(Si);
      dwFlags:=StartF_UseShowWindow;
      WShowWindow:=4;
    End;

```

```

    CreateProcess(nil,PChar(ComLine),nil,nil,False,Create_default_error_mode,nil,nil,Si,p);
{   WaitForSingleObject(p.hProcess,0);}

End;

procedure TForm1.Button1Click(Sender: TObject);
var i,j:integer;
begin

{*****
  Запуск процессов командной строкой с четырьмя параметрами:
    - служебное слово "Процесс";
    - номер процесса;
    - Top;
    - Left.
  Разделение параметров в командной строке через пробел.
  *****/}

  for j:=0 to 2 do { Строки из форм процессов }
  for i:=0 to 1 do { Столбцы из форм процессов }
  begin
    k:=k+1;
    ExecWait('Zadatcha.exe Процесс '+IntToStr(k)+' '
{ № процесса }
                                     +IntToStr(110+j*285)+' '
{ Top                               }
                                     +IntToStr(280+((k-1) div
6)*560+i*280)); { Left           }
    end;
    form1.Edit3.Text:=IntToStr(k);
    if k>n then n:=k;
  end;

procedure TForm1.Timer1Timer(Sender: TObject);
var i,j,k,X1,Y1:integer; Hour,Min,Sec,MSec:Word;
    B : Boolean;
begin

{ Работа с системным временем }

    Tim1:=Now;
    время в Tim1      } { Текущее системное
    if Start then     } { Секундомер }
    Begin
        Tim:=Tim1;    } { Время начала отсчета
    }
    Start:=False;
  end;

```

```

    form1.Edit1.Text:=DateTimeToStr(Tim1); { Системное
время }

    if Tim>0 then form1.Edit2.Text:=TimeToStr(Tim1-Tim); {
Секундомер }

    Hour:= 0; { 0..23 }
    Min := 0; { 0..59 }
    Sec := 5; { 0..59 }
    MSec:= 0; { 0..999 }
    Tim2:= Tim+EncodeTime(Hour,Min,Sec,MSec); { Пример: интервал Tim2
равен Tim + 5 секунд }

end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    k:=0;
    n:=0;
    form1.Edit2.Text:='';
    form1.Edit3.Text:='';
    form1.Memo1.Clear;
    Start:=False; { Выключение секундомера }
    Tim:=0;
end;

procedure Komanda(m:integer; Handle:HWND);
var aCopyData : TCopyDataStruct;
    hTargetWnd : HWND;
    i : integer;
    S1 : AnsiString;
begin

    S1:=IntToStr(m);

    with aCopyData do
    begin
        dwData := 0;
        cbData := StrLen(PChar(S1)) + 1; { StrLen - Возвращает число
символов в строке с нуль-окончанием S }
        lpData := PChar(S1);
    end;

    for i:=1 to k do
    begin

        { Поиск процесса по заголовку его окна "Работник i" }

```

```

    hTargetWnd := FindWindowEx(0, 0, nil, PChar('Работник
'+IntToStr(i)));

    if hTargetWnd <> 0 then SendMessage(hTargetWnd, WM_COPYDATA,
Longint(Handle), Longint(@aCopyData))
        else execWait('Mess.exe '+IntToStr(i));
    end;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    if k=0 then exit;
    Komanda(1,Handle);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    if k=0 then exit;
    Komanda(2,Handle);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
    if k=0 then exit;
    Start:=True; { Включение секундомера }
    Komanda(3,Handle);
    form1.Memo1.Clear;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
    if k=0 then exit;
    Komanda(4,Handle);
    Start:=False; { Выключение секундомера }
    Tim:=0;
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
    if k=0 then exit;
    Komanda(5,Handle);
    k:=0;
    form1.Edit3.Text:='0';
end;

procedure TForm1.WMCopyData(var Msg: TWMCopyData);
var aCopyData: TCopyDataStruct;
begin
    { Извлекает текст из параметра lpData от программы-"работника" }

```

```

    ttt:=StrPas(Msg.CopyDataStruct.lpData);

    form1.Memo1.Lines.Add('Принято от работника '+ttt);

end;

procedure TForm1.Button7Click(Sender: TObject);
begin
    if form1.Caption<>'Программа-"хозяин"' then
    begin
        form1.Caption:='Программа-"хозяин"';
        form1.Button7.Caption:='Супостат';
    end
    else
    begin
        form1.Caption:='Программа Супостат I';
        form1.Button7.Caption:='Хозяин';
    end;
end;

procedure TForm1.Edit3DbClick(Sender: TObject);
var i,j:integer;
begin
    Val(form1.Edit3.Text,i,j);
    if j=0 then k:=i;
    if k>n then n:=k;
end;

procedure TForm1.Button8Click(Sender: TObject);
var aCopyData : TCopyDataStruct;
    hTargetWnd : HWND;
    i           : integer;
    S1          : AnsiString;
begin

    if n=0 then exit;

    S1:='Stop';

    with aCopyData do
    begin
        dwData := 0;
        cbData := StrLen(PChar(S1)) + 1; { StrLen - Возвращает число
символов в строке с нуль-окончанием S }
        lpData := PChar(S1);
    end;

    for i:=1 to n do
    begin

        { Поиск процесса по заголовку его окна "Работник i сачкует!" }

```

```

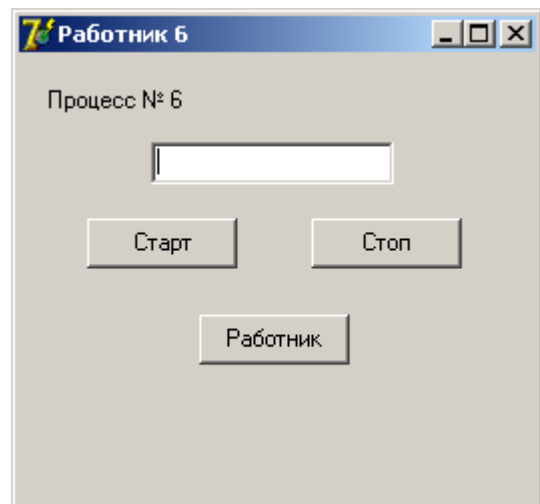
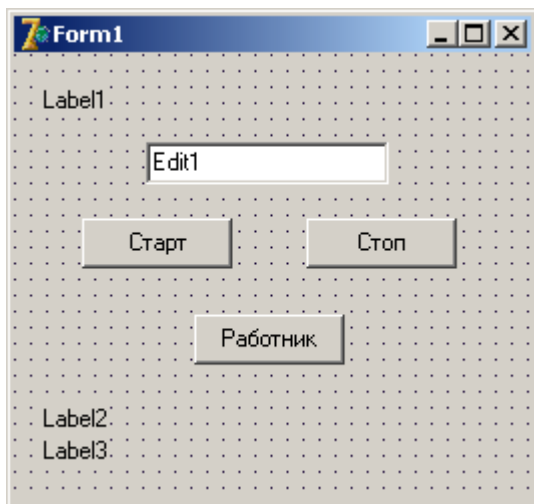
    hTargetWnd := FindWindowEx(0, 0, nil, PChar('Работник
'+IntToStr(i)+ ' сачкует!'));

    if hTargetWnd <> 0 then SendMessage(hTargetWnd, WM_COPYDATA,
Longint(Handle), Longint(@aCopyData));
    end;
end;

end.

```

Головная форма и листинг программы-«хозяина»



Главная форма и листинг программы-«работника»

Модуль программы MainKonv

```
unit MainKonv;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Button3: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure WMCopyData(var Msg: TWMCopyData); message
WM_COPYDATA;
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```

TMessThread = class(TThread)
private
    { Private declarations }
protected
    procedure Execute; override;
end;

var
    Form1: TForm1;
    Start, Stop: boolean;
    ttt: ShortString;
    Left1, Left2, Top1, Top2, k: integer;
    Sender1, Sender2: TObject;
    NewThread: TMessThread;

implementation

{$R *.dfm}

{$R+}

procedure TMessThread.Execute;
var i          : Int64;
    aCopyData  : TCopyDataStruct;
    hTargetWnd : HWND;
    S1         : AnsiString;
begin
    { Place thread code here }

    i:=0;
    repeat
        i:=i+1;
        form1.Edit1.Text:=IntToStr(i);
    until Stop;

    S1:=IntToStr(k)+'/'+IntToStr(i);

    with aCopyData do
        begin
            dwData := 0;
            cbData := StrLen(PChar(S1)) + 1; { StrLen - Возвращает число
симболов в строке с нуль-окончанием S }
            lpData := PChar(S1);
        end;

        { Поиск окна по его заголовку 'Программа-"хозяин"' }

        hTargetWnd := FindWindowEx(0, 0, nil, PChar('Программа-"хозяин"'));

        if hTargetWnd <> 0 then

```

```

begin
    SendMessage(hTargetWnd, WM_COPYDATA, Longint(Handle),
Longint(@aCopyData));
    form1.Label3.Caption:='';
end
else form1.Label3.Caption:='Ой беда!!! Барина
пропал!';

end;

procedure TForm1.Button1Click(Sender: TObject);
begin
    Stop:=false;
    NewThread := TMessThread.Create(False);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    Stop:=True;
end;

procedure TForm1.FormCreate(Sender: TObject);
var i,j:integer;
begin
    k:=0;
    Left1:=280;
    Top1:=110;
    form1.Label1.Caption:='';
    form1.Label2.Caption:='';
    form1.Label3.Caption:='';
    form1.Edit1.Text:='';
    if paramstr(1)<>' ' then ttt:=paramstr(1);
    if paramstr(2)<>' ' then
    begin
        Val(paramstr(2),k,j);
        if j=0 then
            begin
                ttt:=ttt+' № '+IntToStr(k);
                form1.Caption:='Работник '+IntToStr(k);
            end
        else
            begin
                ttt:=ttt+' Error';
                k:=0;
            end;
        end;
    form1.Label1.Caption:=ttt;
    if paramstr(3)<>' ' then ttt:=paramstr(3);
    Val(ttt,i,j);
    if j=0 then form1.Top:=i;
    Top2:=form1.Top;
    if paramstr(4)<>' ' then ttt:=paramstr(4);
    Val(ttt,i,j);

```

```

    if j=0 then form1.Left:=i;
    Left2:=form1.Left;
    Sender1:=Sender;
    Start:=true;
end;

procedure TForm1.WMCopyData(var Msg: TWMCopyData);
var aCopyData: TCopyDataStruct;
    i,j:integer;
    NewThread:TMessThread;
begin
{ Извлекает текст из параметра lpData от "Программа-хозяин" }

    ttt:=StrPas(Msg.CopyDataStruct.lpData);

    label2.Caption := 'Принята команда '+ttt;

    Val(ttt,i,j);

    if j<>0 then exit;

    Case i of
        1:begin
            form1.Top:=Top1+3*k;
            form1.Left:=Left1+3*k;
            end;
        2:begin
            form1.Top:=Top2;
            form1.Left:=Left2;
            end;
        3:begin
            Stop:=false;
            NewThread := TMessThread.Create(False);
            end;
        4:Stop:=true;
        5:Halt;
    end;

end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    if form1.Caption<>'Работник '+IntToStr(k) then
        begin
            form1.Caption:='Работник '+IntToStr(k);
            form1.Button3.Caption:='Лодырь';
        end
        else
        begin
            form1.Caption:='Лодырь '+IntToStr(k);
            form1.Button3.Caption:='Работник';
        end
    end;
end;

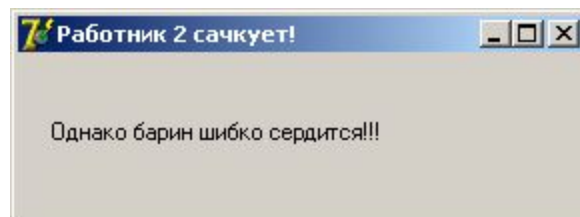
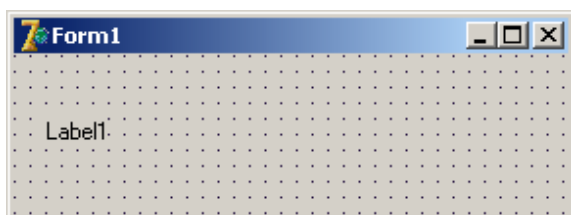
```

```

    end;
    MessageBeep(0);
end;

end.

```



Головная форма и листинг программы «сообщение» (MainMess)

Модуль программы MainMess

```

unit MainMess;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure WMCopyData(var Msg: TWMCopyData); message
WM_COPYDATA;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
var i,j:integer;
begin
  i:=0;

```

```

if paramstr(1)<>' ' then
begin
  form1.Caption:='Работник '+paramstr(1)+ ' сачкует!';
  val(paramstr(1),i,j);
  if j<>0 then i:=0;
  i:=i*3;
  end
  else form1.Caption:='Проба пера';
form1.Label1.Caption:='Однако барин шибко сердится!!!';
form1.Top:=form1.Top+i;
form1.Left:=form1.Left+i;
end;

procedure TForm1.WMCopyData(var Msg: TWMCopyData);
var aCopyData : TCopyDataStruct;
    ttt        : ShortString;
begin

{ Извлекает текст из параметра lpData от "Программа-хозяин" }

  ttt:=StrPas(Msg.CopyDataStruct.lpData);
  if ttt='Stop' then Halt;

end;

end.

procedure TMainForm.PostMessButtonClick(Sender: TObject);
{ Отправка форме сообщения без требования предоставления
результатов обработки }
begin
  PostMessage(Handle, SX_MYMESSAGE, 1, 0);
end;

procedure TMainForm.SendMessButtonClick(Sender: TObject);
{ Отправка форме сообщения с требованием предоставления
результатов обработки }
begin
  SendMessage(Handle, SX_MYMESSAGE, 0, 0);
end;

```

Лабораторная работа № 4

ИССЛЕДОВАНИЕ РАБОТЫ АССОЦИАТИВНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

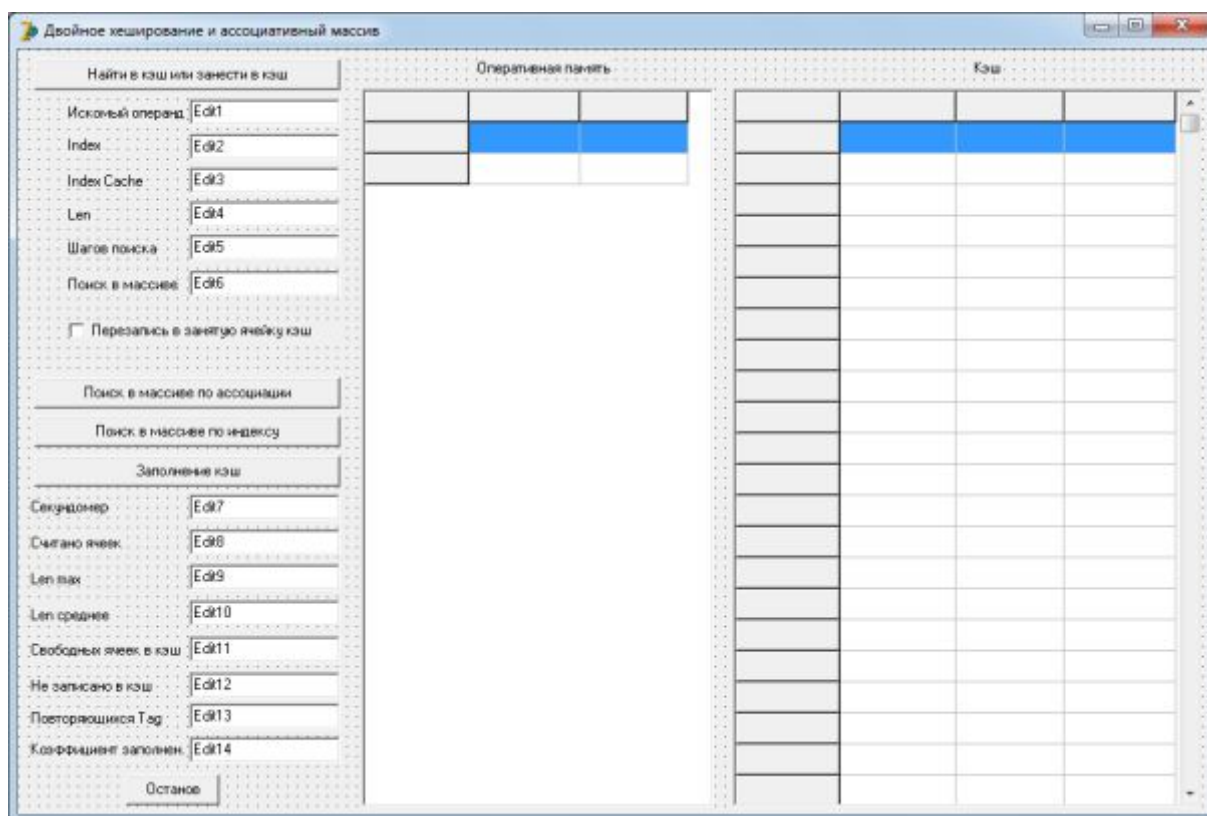
Цель работы

Разработать на Turbo-Delphi приложение, демонстрирующее работу ассоциативного запоминающего устройства. Используя данное приложение, исследовать работу кэш и ассоциативного массива.

Указания к работе

1. Используя среду Turbo-Delphi, разработать приложение, демонстрирующее работу ассоциативного запоминающего устройства.
2. Используя данное приложение, исследовать работу кэш и ассоциативного массива.
3. Оформить отчет.

Внешний вид приложения



Листинг программы ассоциативного запоминающего устройства

```
unit MainHesh;  
  
interface
```

```

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, Math, Grids;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    StringGrid1: TStringGrid;
    Label1: TLabel;
    Label2: TLabel;
    StringGrid2: TStringGrid;
    Label3: TLabel;
    Label4: TLabel;
    Button2: TButton;
    Button3: TButton;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Button4: TButton;
    Edit6: TEdit;
    Edit7: TEdit;
    Label8: TLabel;
    Label9: TLabel;
    Edit8: TEdit;
    Button5: TButton;
    Edit9: TEdit;
    CheckBox1: TCheckBox;
    Edit10: TEdit;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Edit11: TEdit;
    Label13: TLabel;
    Edit12: TEdit;
    Label14: TLabel;
    Edit13: TEdit;
    Label15: TLabel;
    Edit14: TEdit;
    Label16: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);

```



```

    procedure StringGrid1SelectCell(Sender: TObject; ACol, ARow:
Integer;
    var CanSelect: Boolean);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

hash=record
    key : ShortString;
    id : boolean;
end;

const P=30011; { P должно быть простым числом! }
var
    Form1: TForm1;
    H      : array [0..P] of hash;
    len1,len2,Sh : integer;
    DataField   : TStrings;
    Stop        : boolean;

implementation

{$R *.dfm}
{$R+ }

{
    * функция get_id возвращает индекс слова key в массиве H.
    * При этом происходит добавление слова key в словарь H, если его
там нет.
}
function get_id(key:ShortString):integer;
var h1, h2, Ps : int64; i,j:integer;
begin

    { Перевод строки в число }

    if key<>'' then for i:=1 to ORD(key[0]) do
Ps:=Ps+ORD(key[i])*Round(Power(256,ORD(key[0])-i))
        else Ps:=0;

    { Вычисляем две хеш-функции: }

    h1:=13*Ps mod P;                                     { Хеш-функция вычисления
начального адреса от 0 до P-1 }

```

```

    h2:=17*Ps mod P; if h2=0 then h2:=1;    { Хеш-функция шага
поиска свободного адреса от 1 до P-1 }
    len1:=1;    { Число шагов поиска = 1 }

{ Проверим совпадение key =Tag и H[h1].key по hash адресу h1 }

    if key=H[h1].key then
    begin
        get_id:=h1;    { Ячейка найдена }
        exit;
    end;

{ Если ячейка h1 занята, то проверим наличие свободных ячеек }

    if H[h1].id then
    repeat
        h1:=(h1+h2) mod P;    { Вычисление нового шага поиска }
        len1:=len1+1;    { Число шагов поиска +1 }
        if key=H[h1].key then
        begin
            get_id:=h1;    { Ячейка найдена }
            exit;
        end;
    until not H[h1].id or (len1>Sh);

{ Запись при наличии свободной ячейки h1 или принудительная
перезапись в h1 }

    if form1.CheckBox1.Checked and H[h1].id or not H[h1].id then
    begin
        H[h1].key:=key;

form1.StringGrid2.Cells[1,h1+1]:=IntToStr(form1.StringGrid1.Row-
1);
        form1.StringGrid2.Cells[2,h1+1]:=key;
        form1.StringGrid2.Cells[3,h1+1]:=DataField.Values[key];
    end
    else
len2:=len2+1;
    H[h1].id:=true;    { Ячейка занята }
    get_id:=h1;
end;

procedure TForm1.Button1Click(Sender: TObject);
var j:integer;
begin

    { Перезапись из ОЗУ в кэш }

    val(form1.Edit5.Text,Sh,j);    { Допустимое число шагов
поиска свободной ячейки }
    if (j<>0) or (Sh<3) then Sh:=10;

```

```

if Sh>P+1          then Sh:=P+1;
form1.Edit5.Text:=IntToStr(Sh);

j:=get_id(form1.Edit1.Text);          { Перезапись из ОЗУ в кэш }

form1.StringGrid2.Row:=j+1;          { Вывод результатов }
form1.Edit2.Text:=IntToStr(form1.StringGrid1.Row-1);
form1.Edit3.Text:=IntToStr(j);
form1.Edit4.Text:=IntToStr(len1);
end;

procedure TForm1.Button2Click(Sender: TObject);
var i: Integer;
begin

{ Поиск в ассоциативном массиве }

form1.Edit6.Text:=DataField.Values[form1.Edit1.Text];
end;

procedure TForm1.FormCreate(Sender: TObject);
var i,j,k:integer;  ttt:ShortString;
begin
  randomize;
  form1.StringGrid1.RowCount:=P+2;
  form1.StringGrid1.Cells[0,0]:='Index';
  form1.StringGrid1.Cells[1,0]:='Tag';
  form1.StringGrid1.Cells[2,0]:='Data';
  form1.StringGrid2.RowCount:=P+2;
  form1.StringGrid2.Cells[0,0]:='Index hash';
  form1.StringGrid2.Cells[1,0]:='Index';
  form1.StringGrid2.Cells[2,0]:='Tag';
  form1.StringGrid2.Cells[3,0]:='Data';
  for i:=0 to P do
    begin
      H[i].id :=false;    { Ячейка свободна }
      H[i].key:='';
    end;
  DataField := TStringList.Create;
  try
    for i:=0 to form1.StringGrid1.RowCount do
      begin
        form1.StringGrid1.Cells[0,i+1]:=IntToStr(i);
        form1.StringGrid2.Cells[0,i+1]:=IntToStr(i);
        ttt:='';
        for j:=1 to 5 do
          begin
            repeat
              k:=random(100)+48;
            until (Chr(k)>='0') and (Chr(k)<='9') or (Chr(k)>='A') and
(Chr(k)<='Z');

```

```

        ttt:=ttt+Chr(k);
    end;
    form1.StringGrid1.Cells[1,i+1]:=ttt;
    ttt:='';
    for j:=1 to 5 do
        begin
            repeat
                k:=random(100)+48;
                until (Chr(k)>='0') and (Chr(k)<='9') or (Chr(k)>='A') and
(Chr(k)<='Z');
                ttt:=ttt+Chr(k);
            end;
            form1.StringGrid1.Cells[2,i+1]:=ttt;
            DataField.Add(Format('%s=%s',
[form1.StringGrid1.Cells[1,i+1],form1.StringGrid1.Cells[2,i+1]]));
        end;
    finally
    end;
end;

```

```

    form1.Edit1.Text:=form1.StringGrid1.Cells[1,1]; { Поиск по
умолчанию }
    form1.Edit2.Text:='';
    form1.Edit3.Text:='';
    form1.Edit4.Text:='';
    form1.Edit5.Text:='10';
    form1.Edit6.Text:='';
    form1.Edit7.Text:='';
    form1.Edit8.Text:='';
    form1.Edit9.Text:='';
    form1.Edit10.Text:='';
    form1.Edit11.Text:='';
    form1.Edit12.Text:='';
    form1.Edit13.Text:='';
    form1.Edit14.Text:='';
    form1.StringGrid1.Row:=1;
end;

```

```

procedure TForm1.FormClose(Sender: TObject; var Action:
TCloseAction);
begin

```

```

    { Очистка динамической памяти при закрытии }

```

```

    DataField.Free;

```

```

end;

```

```

procedure TForm1.StringGrid1SelectCell(Sender: TObject; ACol,
    ARow: Integer; var CanSelect: Boolean);
begin

```

```

{ Выбор ячейки из ОЗУ }

form1.Edit1.Text:=form1.StringGrid1.Cells[1,ARow];
form1.Edit2.Text:=IntToStr(ARow-1);

end;

procedure TForm1.Button3Click(Sender: TObject);
var i,j:integer;
begin

{ Поиск в ассоциативном массиве по индексу }

val(form1.Edit2.Text,i,j);
if j<>0 then exit;

form1.Edit6.Text:=DataField.Names[i] + ' : ' +
DataField.ValueFromIndex[i];
form1.StringGrid1.Row:=i+1;
end;

procedure TForm1.Button4Click(Sender: TObject);
var i,j,m:integer;    ttt:ShortString;
    k:single;         Tim1,Tim:Double;
begin

{ Перезапись ячеек из ОЗУ в кэш }

j:=0;
k:=0;
Stop:=false;
Tim1:=Now;           { Время начала отсчета }
}
len2:=0;

for i:=0 to P do
begin
form1.Edit1.Text:=form1.StringGrid1.Cells[1,i+1];  { Выбор
ячейки из ОЗУ }
form1.StringGrid1.Row:=i+1;
Form1.Button1Click(Sender);                        { Запись в
кэш }

k:=k+len1;
Tim:=Now;                                           { Текущее системное
время в Tim1 }
form1.Edit7.Text:=TimeToStr(Tim-Tim1);             { Секундомер }

form1.Edit8.Text:=IntToStr(i);                      { Переписано ячеек }
if j<len1 then j:=len1;
form1.Edit9.Text:=IntToStr(j);                     { Len1 max }

```

```

    form1.Edit12.Text:=IntToStr(Len2);      { Len2 не переписано
ячеек из ОЗУ (адреса заняты) }
    form1.Refresh;                          { Вывод данных на
экран }

    Application.ProcessMessages;            { Проверка прерываний
(событий) }

    if Stop then break;                     { Принудительный выход
из цикла }

end;

    { Вывод итоговых данных на экран }

        { Len1 среднее }

k:=k/(P+1);
STR(k:3:1,ttt);
form1.Edit10.Text:=ttt;

if i=0 then
begin
    form1.Edit13.Text:='';
    form1.Edit14.Text:='';
    exit;
end;

    { Число свободных ячеек }

m:=0;
for j:=0 to P do if form1.StringGrid2.Cells[2,j+1]='' then
m:=m+1;
form1.Edit11.Text:=IntToStr(m);

    { Коэффициент заполнения кэш }

k:=(i-m)/i;
STR(k:6:4,ttt);
form1.Edit14.Text:=ttt;

    { Число повторов }

form1.Edit13.Text:=IntToStr(m-Len2);

end;

procedure TForm1.Button5Click(Sender: TObject);
begin
    Stop:=true; { Принудительный выход из цикла }

```

```
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
  if Form1.CheckBox1.Checked then
  begin
    form1.Edit13.Visible:=false;
    form1.Label16.Visible:=false;
  end
  else
  begin
    form1.Edit13.Visible:=true;
    form1.Label16.Visible:=true;
  end;
end;

end.
```

Учебное издание

Кашковский Виктор Владимирович

Архитектура вычислительных систем

Лабораторный практикум

Редактор *В.С. Смирнова*

Компьютерный набор *В.В. Кашковского*

Подписано в печать 12.10.2011.
Формат 60×84 ¹/₁₆. Печать офсетная.
Усл. печ. л. 5,5. Уч.-изд. л. 5,95.
План 2011 г.

Типография ИрГУПС, г. Иркутск, ул. Чернышевского, 15