

ЛАБОРАТОРНАЯ РАБОТА 21. ИСТОЧНИКИ ПРИВЯЗКИ ПРОИЗВОЛЬНОГО ТИПА

21.1. Цель и содержание

Цель лабораторной работы: научиться использовать механизм привязки с объектами-источниками пользовательского типа.

Задачами лабораторной работы являются:

- научиться осуществлять привязку элементов к источнику, который является объектом пользовательского класса;
- научиться разрабатывать многослойные (multi-tier) приложения WPF.

21.2. Теоретическое обоснование

21.2.1. Многомодульные приложения

При разработке реальных информационных программных систем очень редко используются одномодульные приложения (в виде единственного исполняемого файла .exe). Оптимальным вариантом является разбиение приложения на отдельные уровни, каждый из которых представляется отдельной сборкой (модулем). На рис. 21.1 представлена стандартная схема многоуровневого приложения (multi tier application), которая включает: уровень доступа к данным (Data tier), уровень бизнес-логики (Logic tier) и уровень презентаций (Presentation Tier).

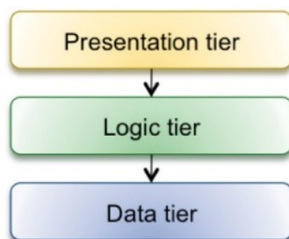


Рисунок 21.1 – Multy tier Application

Такое разделение на отдельные слои позволяет осуществлять многопользовательскую параллельную разработку сложного приложения.

21.2.2. Привязка к произвольным объектам

При реализации механизма привязки необходимо помнить, что целевое свойство должно быть свойством зависимости. Но на объект-источник ограничения отсутствуют. Это означает, что программист может привязать к свойству зависимости любое другое свойство любого другого объекта, независимо от природы его создания.

В предыдущих лабораторных работах рассматривались простые механизмы привязки, основанные на отношении между двумя элементами управления WPF, в котором один элемент – целевой объект привязки, другой – источник привязки.

Но гораздо чаще программисту приходится извлекать данные не из визуального элемента управления, а из объекта определенного бизнес-класса (то есть класса, описанного программистом и предоставляющего необходимые данные, а также реализующего определенный набор действий). В данном механизме необходимо учитывать ограничение – данные для привязки должны находиться в общедоступных свойствах.

При привязке к объекту, который не является элементом управления WPF, пользоваться свойством привязки `ElementName` не представляется возможным. Вместо него программист должен использовать одно из следующих свойств:

1. `Source`. Ссылка, указывающая на исходный объект; другими словами, это объект, поставляющий данные. Существует несколько подходов: извлечение объекта источника из ресурса, программная генерация объекта источника или получение источника от поставщика данных.

2. `RelativeSource`. Указывает на исходный объект, основываясь на отношениях между исходным и целевым объектом. То есть, можно привязать целевой объект к самому себе (сам целевой объект используется в качестве объекта-источника), к родительскому элементу или к определенному элементу, отстоящему от целевого на *N* уровней в дереве элементов.

3. DataContext. Если источник не был указан с помощью свойства Source или RelativeSource, то среда WPF производит поиск в дереве элементов, начиная с текущего элемента. Она проверяет свойство DataContext каждого элемента и использует первый из них, который не равен null. Свойство DataContext исключительно полезно, когда нужно привязать несколько свойств одного объекта к разным элементам, потому что можно установить свойство DataContext высокоуровневого объекта контейнера, вместо его установки непосредственно на целевой элемент.

В рамках данной лабораторной работы будет изучен механизм использования свойства DataContext, как наиболее простого и широко применяемого.

21.3. Методика и порядок выполнения работы

21.3.1. Учебное задание

Необходимо написать приложение для обработки информации о товарах. О каждом товаре необходимо обрабатывать информацию:

- код;
- наименование;
- цена;
- количество;
- описание.

Для освоения методики привязки к объектам произвольного класса, а также получения навыков разработки многомодульных приложений, необходимо реализовать три этапа:

1. Разработать слой доступа к данным.
2. Разработать слой бизнес-логики.
3. Разработать слой презентаций.

Каждый слой должен быть реализовываться в отдельном проекте.

Для решения задачи выполните следующую последовательность действий:

1. В окне Visual Studio создайте пустое решение с именем «Shop» (не проект!). Для этого выберите тип проекта «Visual Studio Solution» (Blank Solution) (рис. 21.1).

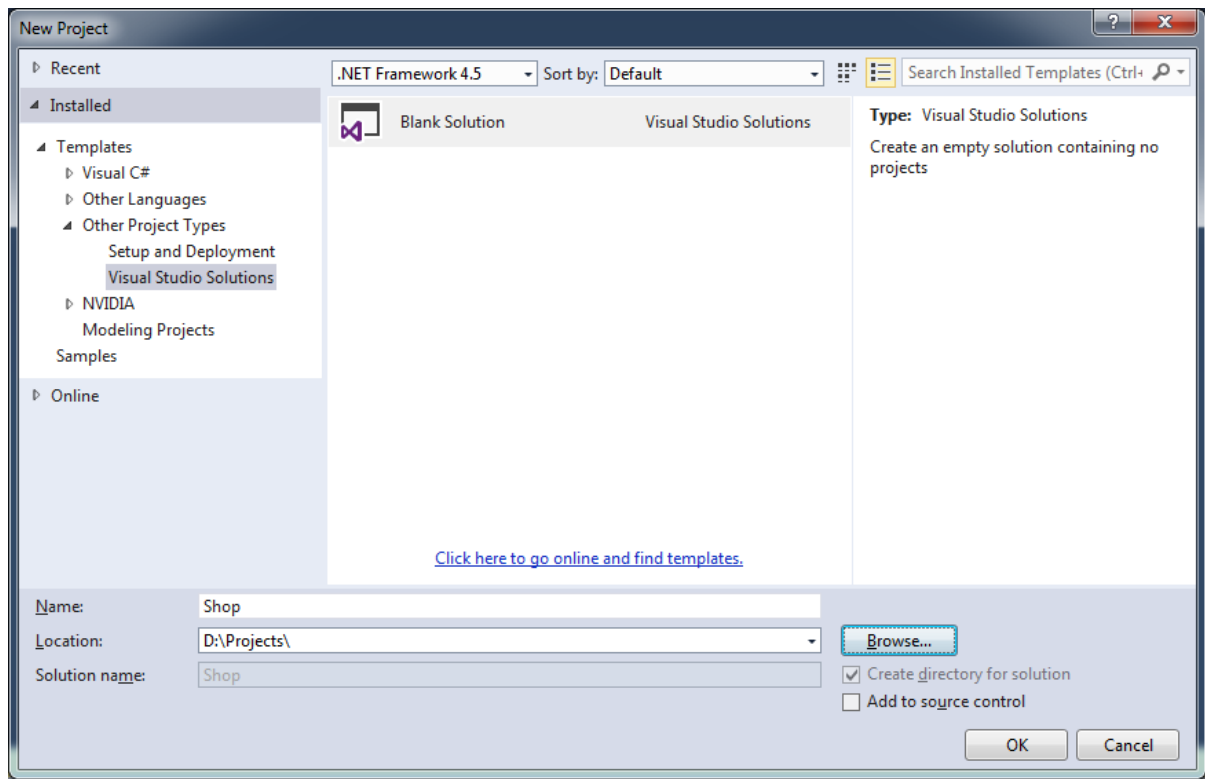


Рисунок 21.1 – Создание пустого решения Visual Studio

2. В обозревателе проектов появится новое пустое решение с именем «Shop» (рисунок 21.2).

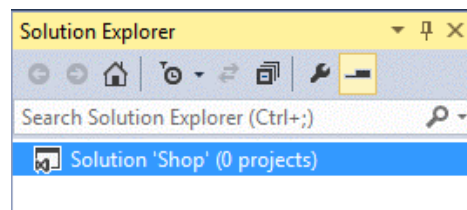


Рисунок 21.2 – Вид обозревателя проекта после создания пустого решения «Shop»

3. Добавьте в решение «Shop» три проекта (таблица 21.1).

Таблица 21.1 – Проекты, добавляемые в решение «Shop»

Имя проекта	Тип проекта	Назначение
DataTier	C# → Windows → Class Library	Библиотека классов для работы с данными. Данный проект представляет уровень доступа к данным в трехуровневой архитектуре.

LogicTier	C# → Windows → Class Library	Библиотека классов, представляющая все необходимые данные и методы для пользовательского интерфейса. Проект представляет уровень бизнес-логики в трехуровневой архитектуре.
PresentationTier	C# → Windows → WPF Application	Проект оконного приложения Windows Presentation Foundation. Данный проект представляет собой уровень презентаций и содержит интерфейс программы.

На рисунке 21.3 показан внешний вид обозревателя проектов после добавления новых проектов.

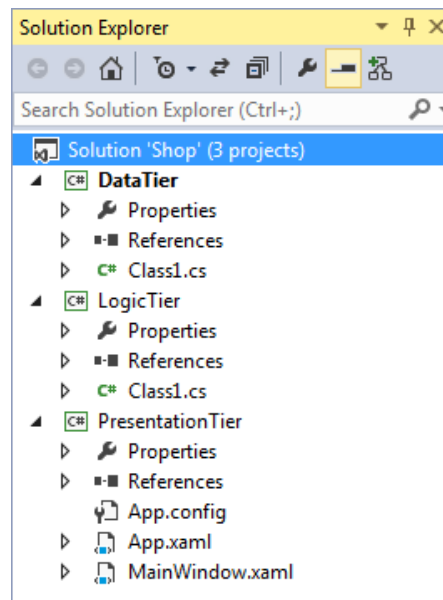


Рисунок 21.3 – Вид обозревателя проектов после добавления отдельных проектов

Наиболее простой способ решения данной задачи – реализовать уровни (tiers) в следующей последовательности с целью избежания архитектурных ошибок разработки:

1) Программирование уровня DataTier. Легче всего реализовать данный уровень первым, т.к. он в большей степени зависит от данных (в данном случае от предметной области из условия задания), чем от Windows-приложения.

2) Программирование уровня PresentationTier. Реализация данного уровня не требует знаний в области программирования, т.к. презентация WPF – это внешний

вид оконного приложения Windows. Данный уровень предполагает, в основном, работу дизайнера.

3) Программирование уровня LogicTier. Цель данного слоя приложения – связать уровни DataTier и PresentationTier.

4. Проект DataTier будет содержать классы, необходимые для представления данных (таблица 21.2).

Таблица 21.2 – Классы проекта «DataTier»

Имя проекта	Назначение
Товар	Представляет отдельный товар; содержит свойства, определяемые заданием
ВсеТовары	Представляет список товаров; содержит методы сохранения и загрузки товаров на носителе

На рис. 21.4 представлен листинг класса «Товар».

```

7 namespace DataTier
8 {
9     11 references
10     public class Товар
11     {
12         7 references
13         public String Код { get; set; }
14         7 references
15         public String Наименование { get; set; }
16         8 references
17         public float Цена { get; set; }
18         7 references
19         public int Количество { get; set; }
20         6 references
21         public String Описание { get; set; }
22     }
23 }

```

Рисунок 21.4 – Определение класса «Товар»

На рис. 21.5 представлен листинг класса «ВсеТовары». Класс «ВсеТовары» содержит два метода:

- метод ПолучитьВсеТовары(), возвращающий список товаров;
- метод СохранитьВсеТовары() для сохранения списка товаров на внешнем носителе.

```

7 namespace DataTier
8 {
9     1reference
10    public class ВсеТовары
11    {
12        1reference
13        public static List<Товар> ПолучитьВсеТовары()
14        {
15            List<Товар> list = new List<Товар>();
16
17            list.Add(
18                new Товар()
19                {
20                    Код = "001", Наименование = "ОС Windows 8", Количество = 10, Цена = 40.99f,
21                    Описание = "Современная операционная система. Версия 8"
22                });
23
24            list.Add(
25                new Товар()
26                {
27                    Код = "002", Наименование = "3D Max", Количество = 2, Цена = 500.99f,
28                    Описание = "Система визуализации и рендеринга от Autodesk Corp."
29                });
30
31            list.Add(
32                new Товар()
33                {
34                    Код = "003", Наименование = "Total Commander 1.00",
35                    Количество = 100, Цена = 0.5f, Описание = " - "
36                });
37
38            list.Add(
39                new Товар()
40                {
41                    Код = "004-001", Наименование = "MS SQL Server",
42                    Количество = 5, Цена = 150.00f, Описание = "СУБД от Microsoft Corp."
43                });
44
45            return list;
46        }
47
48        0 references
49        public static void СохранитьВсеТовары(List<Товар> товары)
50        {
51        }
52    }
53 }

```

Рисунок 21.5 – Определение класса «ВсеТовары»

Классы, представленные на рис. 21.4 и 21.5, являются минимальной абстракцией объектов предметной области. Метод ПолучитьВсеТовары() должен считывать список товаров из файла или получать его из базы данных, но в данном примере осуществляется построение списка в коде. Метод СохранитьВсеТовары() оставим пустым. Все операции сохранения и загрузки данных, взаимодействие с внешними носителями, осуществляются через слой DataTier.

5. Уровни DataTier и LogicTier реализованы в виде библиотек, в которых отсутствует метод Main(). То есть данные проекты не предполагают запуска.

Библиотеки LogicTier и DataTier после выполнения построения будут оформлены в сборки *.dll. На данном шаге решения задачи уровень DataTier реализован, откомпилируйте данный проект и исправьте ошибки, если они есть.

6. Для взаимодействия с пользователем необходимо реализовать уровень презентаций. Проект «PresentationTier» будет содержать главную форму оконного приложения.

На рис. 21.6 представлен внешний вид формы MainWindow.xaml в режиме проектирования.

The image shows a design view of a WPF window titled 'MainWindow.xaml'. The window has a light blue background. It contains a form with several input fields and a large text area. The fields are labeled: 'Код:', 'Наименование:', 'Цена:', 'Количество:', 'Описание:', 'Всего по товару:', 'Суммарная стоимость:', and 'Всего товаров:'. The 'Описание:' field is a large text area. The 'Всего по товару:', 'Суммарная стоимость:', and 'Всего товаров:' fields are at the bottom. The design view shows various handles for resizing and moving elements.

Рисунок 21.6 – Дизайн MainWindow.xaml

Рисунок 21.7 иерархию элементов управления в главном окне.

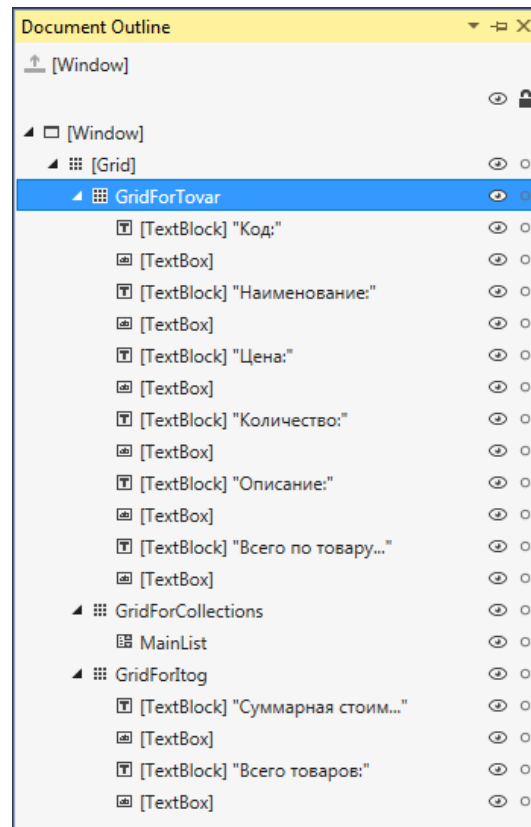


Рисунок 21.7 – Иерархическая структура элементов в окне MainWindow.xaml

Из рисунка 21.7 видно, что окно Window является корневым элементом в визуальном дереве. На втором уровне иерархии расположен элемент Grid без установленного свойства Name, который вмещает в себя все остальные элементы. На третьем уровне расположены три элемента типа Grid: GridForTovar (содержит поля для отображения информации о конкретном товаре), GridForCollections (содержит поле MainList типа ListBox для отображения списка товаров), GridForItog (область для вывода агрегирующих показателей).

На рис. 21.8 представлен код XAML для представленного на рисунках 21.6, 21.7 окна.

```

1 <Window x:Class="PresentationTier.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     Title="{Binding НаименованиеМагазина}" Height="400" Width="800"
5     WindowStartupLocation="CenterScreen" Background="LightCyan">
6     <Grid>
7         <Grid.RowDefinitions>
8             <RowDefinition Height="Auto"></RowDefinition>
9             <RowDefinition></RowDefinition>
10            <RowDefinition Height="Auto"></RowDefinition>
11        </Grid.RowDefinitions>
12        <Grid Grid.Row="0" Name="GridForTovar"
13            DataContext="{Binding ElementName=MainList, Path=SelectedItem}">
14            <Grid.ColumnDefinitions>
15                <ColumnDefinition Width="Auto"></ColumnDefinition>
16                <ColumnDefinition></ColumnDefinition>
17            </Grid.ColumnDefinitions>
18            <Grid.RowDefinitions>
19                <RowDefinition/>
20                <RowDefinition/>
21                <RowDefinition/>
22                <RowDefinition/>
23                <RowDefinition/>
24                <RowDefinition/>
25                <RowDefinition/>
26            </Grid.RowDefinitions>
27            <TextBlock Margin="5" Grid.Row="0" Text="Код:"/>
28            <TextBox Margin="5" Grid.Row="0" Grid.Column="1"
29                Text="{Binding КодТовара}"/>
30            <TextBlock Margin="5" Grid.Row="1" Text="Наименование:"/>
31            <TextBox Margin="5" Grid.Row="1" Grid.Column="1"
32                Text="{Binding НаименованиеТовара}"/>
33            <TextBlock Margin="5" Grid.Row="2" Text="Цена:"/>
34            <TextBox Margin="5" Grid.Row="2" Grid.Column="1"
35                Text="{Binding ЦенаТовара}"/>
36            <TextBlock Margin="5" Grid.Row="3" Text="Количество:"/>
37            <TextBox Margin="5" Grid.Row="3" Grid.Column="1"
38                Text="{Binding КоличествоТовара}"/>
39            <TextBlock Margin="5" Grid.Row="4" Text="Описание:"/>
40            <TextBox Margin="5" Grid.Row="5"
41                Grid.Column="0" Grid.ColumnSpan="2"
42                Text="{Binding ОписаниеТовара}"/>
43            <TextBlock Margin="5" Grid.Row="6" Text="Всего по товару:"/>
44            <TextBox Margin="5" Grid.Row="6" Grid.Column="1"
45                Text="{Binding СуммарнаяСтоимостьПозиции, Mode=OneWay}"/>
46        </Grid>

```

```

47 <Grid Grid.Row="1" Name="GridForCollections">
48     <ListBox Name="MainList" ItemsSource="{Binding СписокТоваров, Mode=OneWay}"
49         DisplayMemberPath="ПредставлениеТовара" Background="Azure"
50         Margin="10"/>
51 </Grid>
52 <Grid Grid.Row="2" Name="GridForItog">
53     <Grid.ColumnDefinitions>
54         <ColumnDefinition/>
55         <ColumnDefinition/>
56         <ColumnDefinition/>
57         <ColumnDefinition/>
58     </Grid.ColumnDefinitions>
59     <TextBlock Margin="5" Text="Суммарная стоимость:" Grid.Column="0"
60         HorizontalAlignment="Right"/>
61     <TextBox Margin="5" Grid.Column="1"
62         Text="{Binding Path=СуммарнаяСтоимость, Mode=OneWay}" />
63     <TextBlock Margin="5" Text="Всего товаров:" Grid.Column="2"
64         HorizontalAlignment="Right"/>
65     <TextBox Margin="5" Grid.Column="3"
66         Text="{Binding Path=СуммарноеКоличество, Mode=OneWay}" />
67 </Grid>
68 </Grid>
69 </Window>

```

Рисунок 21.8 – Код XAML главного окна

Как и в случае использования привязки к элементам управления в данном коде также присутствуют привязки. В некоторых элементах управления устанавливаются различные свойства с использованием {Binding ...}. В связи с этим необходимо дать некоторые пояснения по коду XAML, представленному на рисунке 21.8:

Таблица 21.3 – Особенности кода XAML, связанные с выполнением привязки данных

Номер строки	Исходный код и пояснения
4	Title="{Binding НаименованиеМагазина}" Производится установка заголовка окна. Значение свойства Title привязывается к полю данных НаименованиеМагазина. Такого поля на данном этапе проектирования приложения не существует, поля данных НаименованиеМагазина будет реализовано позже.
29	Text="{Binding КодТовара}" Значение свойства Text элемента управления TextBox привязывается к полю данных КодТовара. Такого поля на данном этапе проектирования приложения не существует, поля данных КодТовара будет реализовано позже.

32, 35, 38, 42	Значение свойства Text элемента управления TextBox привязывается к полю данных, значение которого будет отображено в данном TextBox. Аналогично с механизмом привязки, продемонстрированным в строке 29.
45	Text="{Binding СуммарнаяСтоимостьПозиции, Mode=OneWay}" Значение свойства Text элемента управления TextBox привязывается к полю данных СуммарнаяСтоимостьПозиции в режиме OneWay. Аналогичен предыдущим привязкам, но не предполагает обновления источника привязки при изменении целевого свойства.
48, 49	ItemsSource="{Binding СписокТоваров, Mode=OneWay}" DisplayMemberPath="ПредставлениеТовара" Элемент ListBox отображает не единственное значение, а целый список элементов. Поэтому для привязки к источнику этих элементов применяется свойство ItemsSource. Очевидно, что если нам необходимо выводить список товаров, то нужно указать, какое именно поле будет выводиться в этом списке пользователю; для этого устанавливает свойство DisplayMemberPath. Поле данных «ПредставлениеТовара» на данном этапе не реализовано.
62, 66	Text="{Binding Path=СуммарнаяСтоимость, Mode=OneWay}" Text="{Binding Path=СуммарноеКоличество, Mode=OneWay}" Механизм привязки аналогичен продемонстрированному в строке 45.
13	DataContext="{Binding ElementName=MainList, Path=SelectedItem}" Фактически, выполняется привязка одного элемента управления к другому. Производится привязка свойства DataContext элемента Grid к элементу MainList типа ListBox. Но не ко всему списку, а только к объекту, заданному свойством Path, то есть к свойству ListBox.SelectedItem (текущий выбранный элемент).

После верстки главного окна с использование инструментария XAML приложение WPF может быть запущено. Внешний вид созданного приложения представлен на рис. 21.9.

Рисунок 21.9 – Главное окно приложения в режиме выполнения

На данном этапе приложение функционирует, но нет связи с данными. Несмотря на то, что осуществлялась привязка к несуществующим данным, приложение все равно работает. Теперь в нашем приложении две сборки: DataTier.dll и PresentationTier.exe. Реализуем связующее звено – LogicTier.dll.

7. В проекте LogicTier создадим классы, представленные в таблице 21.4.

Таблица 21.4 – Особенности кода XAML, связанные с выполнением привязки данных

Имя класса	Описание
ТоварнаяПозиция	Это класс, который «оборачивает» класс «Товар» из проекта DataTier. Класс «ТоварнаяПозиция» дополняет «сырой» класс данных всем необходимым для вышестоящих уровней приложения.
Магазин	Данный класс «оборачивает» класс «ВсеТовары» из проекта DataTier.

На рисунке 21.10 представлен листинг класса «Товарная позиция».

```

8 namespace LogicTier
9 {
10     public class ТоварнаяПозиция
11     {
12         private Товар _товар;
13
14         public ТоварнаяПозиция(Товар p)
15         {
16             _товар = p;
17         }
18         public String КодТовара
19         {
20             get { return _товар.Код; }
21             set { _товар.Код = value; }
22         }
23
24         public String НаименованиеТовара
25         {
26             get { return _товар.Наименование; }
27             set { _товар.Наименование = value; }
28         }
29
30         public float ЦенаТовара
31         {
32             get { return _товар.Цена; }
33             set { _товар.Цена = value; }
34         }
35
36         public int КоличествоТовара
37         {
38             get { return _товар.Количество; }
39             set { _товар.Количество = value; }
40         }
41
42         public String ОписаниеТовара
43         {
44             get { return _товар.Описание; }
45             set { _товар.Описание = value; }
46         }
47
48         public float СуммарнаяСтоимостьПозиции
49         {
50             get { return _товар.Цена * _товар.Количество; }
51         }
52
53         public String ПредставлениеТовара
54         {
55             get { return _товар.Код + " : " + _товар.Наименование
56                 + " (" + _товар.Цена.ToString("C") + ")"; }
57         }
58     }
59 }

```

Рисунок 21.10 – Код класса «ТоварнаяПозиция»

В представленном коде производится связь между уровнем презентаций и уровнем данных: для каждой привязки к единичному товару в коде XAML

(таблица 21.3) в данном классе создается одноименное поле. Если привязка производилась в режиме OneWay, поле может быть объявлено только для чтения (например, поле ПредставлениеТовара).

В свою очередь, каждое поле из класса «ТоварнаяПозиция» является оберткой над соответствующим полем класса «Товар» (т.е. представляет это поле или вычисляется на основе нескольких полей).

Листинг класса «Магазин» представлен на рисунок 21.11.

```

8 namespace LogicTier
9 {
10     public class Магазин
11     {
12         private List<ТоварнаяПозиция> _товары = new List<ТоварнаяПозиция>();
13
14         public Магазин()
15         {
16             List<Товар> tmp = ВсеТовары.ПолучитьВсеТовары();
17             foreach (var t in tmp)
18             {
19                 _товары.Add(new ТоварнаяПозиция(t));
20             }
21         }
22
23         public List<ТоварнаяПозиция> СписокТоваров
24         {
25             get { return _товары; }
26         }
27
28         public String НаименованиеМагазина {
29             get { return "Наш магазин"; }
30         }
31
32         public float СуммарнаяСтоимость {
33             get
34             {
35                 return _товары.Sum(p => p.СуммарнаяСтоимостьПозиции);
36             }
37         }
38
39         public float СуммарноеКоличество
40         {
41             get
42             {
43                 return _товары.Sum(p => p.КоличествоТовара);
44             }
45         }
46     }
47 }

```

Рисунок 21.11 – Определение класса «Магазин»

Класс «Магазин» – это «обертка» над классом «ВсеТовары».

После компиляции и исправления ошибок получим три сборки: DataTier.dll, LogicTier.dll, PresentationTier.exe.

8. Завершающий этап разработки приложения: связь класса MainWindow и класса LogicTier. Для выполнения данной связи необходимо задать свойство DataContext главного окна. На рисунок 21.12 представлен измененный код конструктора MainWindow.

```

17 namespace PresentationTier
18 {
19     /// <summary>
20     /// Interaction logic for MainWindow.xaml
21     /// </summary>
22     public partial class MainWindow : Window
23     {
24         public MainWindow()
25         {
26             Магазин logicTier = new Магазин();
27             this.DataContext = logicTier;
28             InitializeComponent();
29         }
30     }
31 }

```

Рисунок 21.12 – Конструктор MainWindow

Из листинга видно, что создается объект класса «Магазин» и устанавливается в качестве источника привязки через свойство DataContext для окна MainWindow. После этого все привязки начинают получать данные. Внешний вид окна полученного приложения представлен на рисунок 21.13.

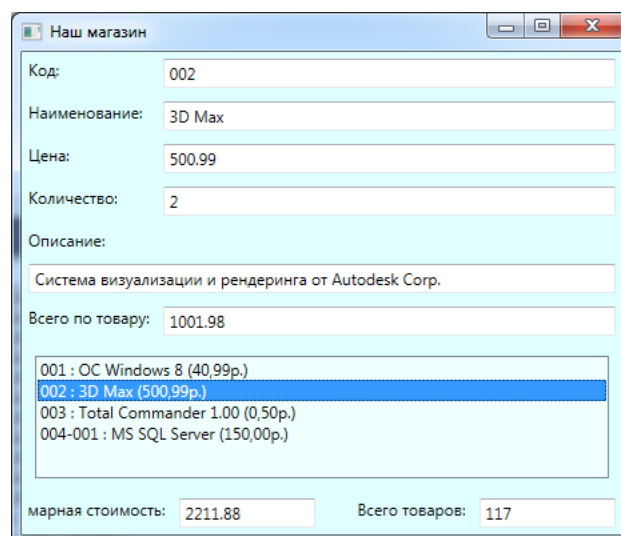


Рисунок 21.13 – Результат решения задачи

9. Выполните задание в соответствии с индивидуальным вариантом.

21.3.2. Индивидуальное задание

В каждом варианте необходимо реализовать многоуровневое приложение для обработки данных из файла. Формат текстового файла представлен в каждом варианте. В отличие от рассмотренного примера, приложение должно считывать информацию из файла. Приложение должно быть многомодульным (как в реализованном примере).

Вариант	Задание
1	Исходный файл содержит список преподавателей в формате «Товар, цена, количество, магазин» (произвольное количество строк): <pre> 1 Рис 1 кг. / 120.11 / 54 / Магнит 2 Ручка шариковая / 23.00 / 15 / ТЦ Карандаш 3 Тетрадь 12 л. / 15.00 / 200 / ГеоКнига 4 Тетрадь 60 л. / 90.70 / 300 / ГеоКнига 5 Торт 1 кг. / 450 / 4 / Магнит </pre> Необходимо вычислить: суммарную стоимость товаров в каждом магазине; магазин с минимальным товарным ассортиментом.
2	Исходный файл содержит список преподавателей в формате «ФИО, должность, кафедра, заработная плата» (произвольное количество строк): <pre> 1 Иванов С.Е. доцент ИСТ 56.00 2 Петров А.А. профессор ИБАС 100.23 3 Новиков П.С. ст. преподаватель ИСТ 35.00 4 Ганотченко Е.П. профессор ИБАС 200.45 5 Петров А.А. профессор ИБАС 105.67 </pre> Необходимо вычислить: количество преподавателей; сумму зарплат преподавателей по каждой кафедре.
3	Исходный файл содержит список преподавателей в формате «Модель, производитель, стоимость, пробег» (произвольное количество строк): <pre> 1 ВАЗ-2107; Автоваз; 140000.57; 16000 2 Гога-1; Хуань Вей Трейд; 50.00; 1000 3 ВАЗ-5000; Автоваз; 1400000.00; 1000 4 X35; Hyundai; 750000; 1 </pre> Необходимо вычислить: суммарную стоимость всех автомобилей; автомобиль с наименьшим пробегом.
4	Исходный файл содержит список преподавателей в формате «Транспорт; пункт отправки; пункт назначения; стоимость билета» (произвольное количество строк): <pre> 1 Самолет*Москва*Пекин*30000.00 2 Автобус*Ставрополь*Москва*3000.00 3 Автобус*Невинномысск*Ставрополь*10.00 4 Автобус*Москва*Владивосток*50000 5 Самолет*СПб*Москва*1500 </pre> Необходимо вычислить: количество автобусных рейсов; суммарную стоимость билетов рейсов самолетов; самый дорогой билет.

5	Исходный файл содержит список преподавателей в формате «ФИО, группа, курс, количество задолженностей» (произвольное количество строк): <pre>1 Иванов Е.А.*ИСТ-б-о-121*3*2 2 Петров А.А.*ИСТ-б-о-111*4*1 3 Кононов А.Е.*БАС-б-о-111*4*0 4 Катченко Е.А.*БАС-б-о-121*3*1</pre> Необходимо вычислить: суммарное количество задолженностей по каждому курсу; количество студентов без задолженностей.
6	Исходный файл содержит список преподавателей в формате «Товар, товарная группа, цена закупки, цена продажи» (произвольное количество строк): <pre>1 Молоко, 0.5 л % Продукты % 10.00 % 23.00 2 Хлеб белый % Продукты % 15.00 % 22.00 3 Молоко, 1.0 л % Продукты % 18.00 % 30.00 4 Туалетный утенок % Химия % 11.00 % 17.58 5 феири, 0.5 л % Химия % 17.00 % 88.00</pre> Необходимо вычислить: суммарную стоимость товаров по каждой группе; самый дорогой товар в каждой товарной группе.
7	Исходный файл содержит список преподавателей в формате «Товар, товарная группа, цена, склад» (произвольное количество строк): <pre>1 Кабель сетевой % Сетевое оборудование % 10.00 % Пулково 1 2 Коммутатор ASUS X100 % Сетевое оборудование % 15.00 % Склад №7 3 ПО Windows 8.1, 1.0 л % ПО % 18.00 % Склад №19 4 ПО Office 365 % ПО % 11.00 % Склад №19 5 Коннектор 234511 % Сетевое оборудование % 17.00 % Склад №7</pre> Необходимо вычислить: среднюю стоимость товаров по каждой группе; склад, содержащий максимальное количество продукции.
8	Исходный файл содержит список преподавателей в формате «Тип строения, количество комнат, метраж, стоимость» (произвольное количество строк): <pre>1 Дом*6*150*35000000 2 Квартира*1*49*790000 3 Квартира*2*59*1500000 4 Квартира*3*70*3000000 5 Комната*1*12*500000</pre> Необходимо вычислить: для каждого строения рассчитать стоимость квадратного метра; суммарную стоимость объектов по каждому типу.
9	Исходный файл содержит список преподавателей в формате «Название книги, жанр, цена» (произвольное количество строк): <pre>1 Привет с того света Детектив 150.00 2 Кто подставил кролика Детектив 200.10 3 Почемучка Детская литература 60.22 4 Капитошка Детская литература 230.14 5 Звезда 112 Фантастика 12.90</pre> Необходимо вычислить: самую дешевую книгу в каждом жанре; среднюю стоимость книг в каждом жанре.
10	Исходный файл содержит список преподавателей в формате «Товар, группа, стоимость, процент скидки» (произвольное количество строк): <pre>1 Молоко \ Продукты \ 40.00 \ 1 2 Тряпка \ Хозтовары \ 10.00 \ 5 3 Порошок \ Хозтовары \ 750.00 \ 10 4 Мясо \ Продукты \ 350 \ 5 5 Швабра \ Хозтовары \ 900.00 \ 5</pre> Необходимо вычислить: цену каждого товара за вычетом скидки; суммарную стоимость товаров по каждой группе.

11	Исходный файл содержит список преподавателей в формате «Товар, цена, количество, магазин» (произвольное количество строк): <pre> 1 Рис 1 кг. / 120.11 / 54 / Магнит 2 Ручка шариковая / 23.00 / 15 / ТЦ Карандаш 3 Тетрадь 12 л. / 15.00 / 200 / ГеоКнига 4 Тетрадь 60 л. / 90.70 / 300 / ГеоКнига 5 Торт 1 кг. / 450 / 4 / Магнит </pre> Необходимо вычислить: среднюю цену товара по каждому магазину; суммарную стоимость для каждого товара.
12	Исходный файл содержит список преподавателей в формате «ФИО, должность, кафедра, заработная плата» (произвольное количество строк): <pre> 1 Иванов С.Е. доцент ИСТ 56.00 2 Петров А.А. профессор ИБАС 100.23 3 Новиков П.С. ст. преподаватель ИСТ 35.00 4 Ганотченко Е.П. профессор ИБАС 200.45 5 Петров А.А. профессор ИБАС 105.67 </pre> Необходимо вычислить: суммарный фонд заработной платы; среднюю зарплату по каждой кафедре.
13	Исходный файл содержит список преподавателей в формате «Модель, производитель, стоимость, пробег» (произвольное количество строк): <pre> 1 ВАЗ-2107; Автоваз; 140000.57; 16000 2 Гога-1; Хуань Вей Трейд; 50.00; 1000 3 ВАЗ-5000; Автоваз; 1400000.00; 1000 4 X35; Hyundai; 750000; 1 </pre> Необходимо вычислить: средний пробег по каждому производителю; среднюю стоимость автомобилей по каждому производителю.
14	Исходный файл содержит список преподавателей в формате «Транспорт; пункт отправки; пункт назначения; стоимость билета» (произвольное количество строк): <pre> 1 Самолет*Москва*Пекин*30000.00 2 Автобус*Ставрополь*Москва*3000.00 3 Автобус*Невинномысск*Ставрополь*10.00 4 Автобус*Москва*Владивосток*50000 5 Самолет*СПб*Москва*1500 </pre> Необходимо вычислить: среднюю стоимость билета по каждому виду транспорта; количество рейсов из Москвы.
15	Исходный файл содержит список преподавателей в формате «ФИО, группа, курс, количество задолженностей» (произвольное количество строк): <pre> 1 Иванов Е.А.*ИСТ-б-о-121*3*2 2 Петров А.А.*ИСТ-б-о-111*4*1 3 Кононов А.Е.*БАС-б-о-111*4*0 4 Катченко Е.А.*БАС-б-о-121*3*1 </pre> Необходимо вычислить: суммарное количество задолженностей по каждой группе; среднее количество задолженностей по каждому курсу.
16	Исходный файл содержит список преподавателей в формате «Товар, товарная группа, цена закупки, цена продажи» (произвольное количество строк): <pre> 1 Молоко, 0.5 л % Продукты % 10.00 % 23.00 2 Хлеб белый % Продукты % 15.00 % 22.00 3 Молоко, 1.0 л % Продукты % 18.00 % 30.00 4 Туалетный утенок % Химия % 11.00 % 17.58 5 феири, 0.5 л % Химия % 17.00 % 88.00 </pre> Необходимо вычислить: суммарную прибыль от продажи товаров; среднюю цену закупки по каждой группе.

17	Исходный файл содержит список преподавателей в формате «Товар, товарная группа, цена, склад» (произвольное количество строк): <pre> 1 Кабель сетевой & Сетевое оборудование & 10.00 & Пулково 1 2 Коммутатор ASUS X100 & Сетевое оборудование & 15.00 & Склад №7 3 ПО Windows 8.1, 1.0 л & ПО & 18.00 & Склад №19 4 ПО Office 365 & ПО & 11.00 & Склад №19 5 Коннектор 234511 & Сетевое оборудование & 17.00 & Склад №7 </pre> Необходимо вычислить: среднюю стоимость товаров по каждому складу; суммарную стоимость по каждой группе.
18	Исходный файл содержит список преподавателей в формате «Тип строения, количество комнат, метраж, стоимость» (произвольное количество строк): <pre> 1 Дом*6*150*3500000 2 Квартира*1*49*790000 3 Квартира*2*59*1500000 4 Квартира*3*70*3000000 5 Комната*1*12*500000 </pre> Необходимо вычислить: вывести объект с максимальной стоимостью квадратного метра; среднюю стоимость квадратного метра по объектам типа «Квартира».
19	Исходный файл содержит список преподавателей в формате «Название книги, жанр, цена» (произвольное количество строк): <pre> 1 Привет с того света Детектив 150.00 2 Кто подставил кролика Детектив 200.10 3 Почему-то Детская литература 60.22 4 Капитошка Детская литература 230.14 5 Звезда 112 фантастика 12.90 </pre> Необходимо вычислить: суммарную стоимость книг по каждому жанру; среднюю стоимость книг по жанру «Фантастика».
20	Исходный файл содержит список преподавателей в формате «Товар, группа, стоимость, процент скидки» (произвольное количество строк): <pre> 1 Молоко \ Продукты \ 40.00 \ 1 2 Тряпка \ Хозтовары \ 10.00 \ 5 3 Порошок \ Хозтовары \ 750.00 \ 10 4 Мясо \ Продукты \ 350 \ 5 5 Швабра \ Хозтовары \ 900.00 \ 5 </pre> Необходимо вычислить: среднюю цену товара (с учетом скидки) по каждой товарной группе; товар с минимальной скидкой.
21	Исходный файл содержит список преподавателей в формате «Товар, цена, количество, магазин» (произвольное количество строк): <pre> 1 Рис 1 кг. / 120.11 / 54 / Магнит 2 Ручка шариковая / 23.00 / 15 / ТЦ Карандаш 3 Тетрадь 12 л. / 15.00 / 200 / ГеоКнига 4 Тетрадь 60 л. / 90.70 / 300 / ГеоКнига 5 Торт 1 кг. / 450 / 4 / Магнит </pre> Необходимо вычислить: суммарную стоимость товаров в каждом магазине; магазин с минимальным товарным ассортиментом.
22	Исходный файл содержит список преподавателей в формате «ФИО, должность, кафедра, заработная плата» (произвольное количество строк): <pre> 1 Иванов С.Е. доцент ИСТ 56.00 2 Петров А.А. профессор ИБАС 100.23 3 Новиков П.С. ст. преподаватель ИСТ 35.00 4 Ганотченко Е.П. профессор ИБАС 200.45 5 Петров А.А. профессор ИБАС 105.67 </pre> Необходимо вычислить: количество преподавателей; сумму зарплат преподавателей по каждой кафедре.

23	Исходный файл содержит список преподавателей в формате «Модель, производитель, стоимость, пробег» (произвольное количество строк): <pre> 1 ВАЗ-2107; Автоваз; 140000.57; 16000 2 Гога-1; Хуань Вей Трейд; 50.00; 1000 3 ВАЗ-5000; Автоваз; 1400000.00; 1000 4 X35; Hyundai; 750000; 1 </pre> Необходимо вычислить: суммарную стоимость всех автомобилей; автомобиль с наименьшим пробегом.
24	Исходный файл содержит список преподавателей в формате «Транспорт; пункт отправки; пункт назначения; стоимость билета» (произвольное количество строк): <pre> 1 Самолет*Москва*Пекин*30000.00 2 Автобус*Ставрополь*Москва*3000.00 3 Автобус*Невинномысск*Ставрополь*10.00 4 Автобус*Москва*Владивосток*50000 5 Самолет*СПб*Москва*1500 </pre> Необходимо вычислить: количество автобусных рейсов; суммарную стоимость билетов рейсов самолетов; самый дорогой билет.
25	Исходный файл содержит список преподавателей в формате «ФИО, группа, курс, количество задолженностей» (произвольное количество строк): <pre> 1 Иванов Е.А.*ИСТ-б-о-121*3*2 2 Петров А.А.*ИСТ-б-о-111*4*1 3 Кононов А.Е.*БАС-б-о-111*4*0 4 Катченко Е.А.*БАС-б-о-121*3*1 </pre> Необходимо вычислить: суммарное количество задолженностей по каждому курсу; количество студентов без задолженностей.
26	Исходный файл содержит список преподавателей в формате «Товар, товарная группа, цена закупки, цена продажи» (произвольное количество строк): <pre> 1 Молоко, 0.5 л % Продукты % 10.00 % 23.00 2 Хлеб белый % Продукты % 15.00 % 22.00 3 Молоко, 1.0 л % Продукты % 18.00 % 30.00 4 Туалетный утенок % Химия % 11.00 % 17.58 5 феири, 0.5 л % Химия % 17.00 % 88.00 </pre> Необходимо вычислить: суммарную стоимость товаров по каждой группе; самый дорогой товар в каждой товарной группе.
27	Исходный файл содержит список преподавателей в формате «Товар, товарная группа, цена, склад» (произвольное количество строк): <pre> 1 Кабель сетевой % Сетевое оборудование % 10.00 % Пулково 1 2 Коммутатор ASUS X100 % Сетевое оборудование % 15.00 % Склад №7 3 ПО Windows 8.1, 1.0 л % ПО % 18.00 % Склад №19 4 ПО Office 365 % ПО % 11.00 % Склад №19 5 Коннектор 234511 % Сетевое оборудование % 17.00 % Склад №7 </pre> Необходимо вычислить: среднюю стоимость товаров по каждой группе; склад, содержащий максимальное количество продукции.
28	Исходный файл содержит список преподавателей в формате «Тип строения, количество комнат, метраж, стоимость» (произвольное количество строк): <pre> 1 Дом*6*150*3500000 2 Квартира*1*49*790000 3 Квартира*2*59*1500000 4 Квартира*3*70*3000000 5 Комната*1*12*500000 </pre> Необходимо вычислить: для каждого строения рассчитать стоимость квадратного метра; суммарную стоимость объектов по каждому типу.

29	Исходный файл содержит список преподавателей в формате «Название книги, жанр, цена» (произвольное количество строк): <pre> 1 Привет с того света Детектив 150.00 2 Кто подставил кролика Детектив 200.10 3 Почемучка Детская литература 60.22 4 Капитошка Детская литература 230.14 5 Звезда 112 Фантастика 12.90 </pre> Необходимо вычислить: самую дешевую книгу в каждом жанре; среднюю стоимость книг в каждом жанре.
30	Исходный файл содержит список преподавателей в формате «Товар, группа, стоимость, процент скидки» (произвольное количество строк): <pre> 1 Молоко \ Продукты \ 40.00 \ 1 2 Тряпка \ Хозтовары \ 10.00 \ 5 3 Порошок \ Хозтовары \ 750.00 \ 10 4 Мясо \ Продукты \ 350 \ 5 5 Швабра \ Хозтовары \ 900.00 \ 5 </pre> Необходимо вычислить: цену каждого товара за вычетом скидки; суммарную стоимость товаров по каждой группе.
31	Исходный файл содержит список преподавателей в формате «Товар, цена, количество, магазин» (произвольное количество строк): <pre> 1 Рис 1 кг. / 120.11 / 54 / Магнит 2 Ручка шариковая / 23.00 / 15 / ТЦ Карандаш 3 Тетрадь 12 л. / 15.00 / 200 / ГеоКнига 4 Тетрадь 60 л. / 90.70 / 300 / ГеоКнига 5 Торт 1 кг. / 450 / 4 / Магнит </pre> Необходимо вычислить: среднюю цену товара по каждому магазину; суммарную стоимость для каждого товара.
32	Исходный файл содержит список преподавателей в формате «ФИО, должность, кафедра, заработная плата» (произвольное количество строк): <pre> 1 Иванов С.Е. доцент ИСТ 56.00 2 Петров А.А. профессор ИБАС 100.23 3 Новиков П.С. ст. преподаватель ИСТ 35.00 4 Ганотченко Е.П. профессор ИБАС 200.45 5 Петров А.А. профессор ИБАС 105.67 </pre> Необходимо вычислить: суммарный фонд заработной платы; среднюю зарплату по каждой кафедре.
33	Исходный файл содержит список преподавателей в формате «Модель, производитель, стоимость, пробег» (произвольное количество строк): <pre> 1 ВАЗ-2107; Автоваз; 140000.57; 16000 2 Гога-1; Хуань Вей Трейд; 50.00; 1000 3 ВАЗ-5000; Автоваз; 1400000.00; 1000 4 X35; Hyundai; 750000; 1 </pre> Необходимо вычислить: средний пробег по каждому производителю; среднюю стоимость автомобилей по каждому производителю.
34	Исходный файл содержит список преподавателей в формате «Транспорт; пункт отправки; пункт назначения; стоимость билета» (произвольное количество строк): <pre> 1 Самолет*Москва*Пекин*30000.00 2 Автобус*Ставрополь*Москва*3000.00 3 Автобус*Невинномысск*Ставрополь*10.00 4 Автобус*Москва*Владивосток*50000 5 Самолет*СПб*Москва*1500 </pre> Необходимо вычислить: среднюю стоимость билета по каждому виду транспорта; количество рейсов из Москвы.

35	Исходный файл содержит список преподавателей в формате «ФИО, группа, курс, количество задолженностей» (произвольное количество строк): <pre>1 Иванов Е.А.*ИСТ-б-о-121*3*2 2 Петров А.А.*ИСТ-б-о-111*4*1 3 Кононов А.Е.*БАС-б-о-111*4*0 4 Катченко Е.А.*БАС-б-о-121*3*1</pre> Необходимо вычислить: суммарное количество задолженностей по каждой группе; среднее количество задолженностей по каждому курсу.
36	Исходный файл содержит список преподавателей в формате «Товар, товарная группа, цена закупки, цена продажи» (произвольное количество строк): <pre>1 Молоко, 0.5 л % Продукты % 10.00 % 23.00 2 Хлеб белый % Продукты % 15.00 % 22.00 3 Молоко, 1.0 л % Продукты % 18.00 % 30.00 4 Туалетный утенок % Химия % 11.00 % 17.58 5 Фейри, 0.5 л % Химия % 17.00 % 88.00</pre> Необходимо вычислить: суммарную прибыль от продажи товаров; среднюю цену закупки по каждой группе.
37	Исходный файл содержит список преподавателей в формате «Товар, товарная группа, цена, склад» (произвольное количество строк): <pre>1 Кабель сетевой % Сетевое оборудование % 10.00 % Пулковое 1 2 Коммутатор ASUS X100 % Сетевое оборудование % 15.00 % Склад №7 3 ПО Windows 8.1, 1.0 л % ПО % 18.00 % Склад №19 4 ПО Office 365 % ПО % 11.00 % Склад №19 5 Коннектор 234511 % Сетевое оборудование % 17.00 % Склад №7</pre> Необходимо вычислить: среднюю стоимость товаров по каждому складу; суммарную стоимость по каждой группе.
38	Исходный файл содержит список преподавателей в формате «Тип строения, количество комнат, метраж, стоимость» (произвольное количество строк): <pre>1 Дом*6*150*35000000 2 Квартира*1*49*790000 3 Квартира*2*59*1500000 4 Квартира*3*70*3000000 5 Комната*1*12*500000</pre> Необходимо вычислить: вывести объект с максимальной стоимостью квадратного метра; среднюю стоимость квадратного метра по объектам типа «Квартира».
39	Исходный файл содержит список преподавателей в формате «Название книги, жанр, цена» (произвольное количество строк): <pre>1 Привет с того света Детектив 150.00 2 Кто подставил кролика Детектив 200.10 3 Почемучка Детская литература 60.22 4 Капитошка Детская литература 230.14 5 Звезда 112 Фантастика 12.90</pre> Необходимо вычислить: суммарную стоимость книг по каждому жанру; среднюю стоимость книг по жанру «Фантастика».
40	Исходный файл содержит список преподавателей в формате «Товар, группа, стоимость, процент скидки» (произвольное количество строк): <pre>1 Молоко \ Продукты \ 40.00 \ 1 2 Тряпка \ Хозтовары \ 10.00 \ 5 3 Порошок \ Хозтовары \ 750.00 \ 10 4 Мясо \ Продукты \ 350 \ 5 5 Швабра \ Хозтовары \ 900.00 \ 5</pre> Необходимо вычислить: среднюю цену товара (с учетом скидки) по каждой товарной группе; товар с минимальной скидкой.

21.4. Вопросы для защиты работы

1. Чем отличается привязка к элементам управления от привязки к произвольным объектам?
2. Поясните синтаксис привязки к данным. Какие ключевые слова и для каких целей используются при оформлении привязки с помощью XAML?
3. Какие режимы привязки существуют? Поясните назначение всех режимов привязки.
4. Поясните схему привязки элементов управления. Назовите основные элементы данной схемы.
5. Что означают понятия «целевой элемент», «элемент-источник привязки», «целевое свойство», «свойство-источник»?
6. Какое ограничение на источник привязки существует в WPF?
7. Какое ограничение существует на целевое свойство привязки?