|  |
| --- |
| 09.02.07 ПР-419  **ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**  **ПМ.11 Разработка, администрирование и защита баз данных** |
| **УП.11.02 Разработка приложения** |

|  |  |  |  |
| --- | --- | --- | --- |
| Практикант |  |  | Гудожникова Д.М.  Высоцкая А.Н. |
| Руководитель практики |  |  | **Пьяненкова А.В.**  **Бабикова .Т.М.** |
|  |  |  |  |

Оглавление

[УП.11.02 Разработка приложения 3](#_Toc118540826)

[**База практики 3**](#_Toc118540827)

[**Период прохождения практики 3**](#_Toc118540828)

[**Основная задача практики 3**](#_Toc118540829)

[**Виды работ 3**](#_Toc118540830)

[**Задание на разработку 4**](#_Toc118540831)

[Описание организации данных 5](#_Toc118540832)

[**Схема данных 6**](#_Toc118540833)

[**Описание таблиц 6**](#_Toc118540834)

[**Описание интерфейса 8**](#_Toc118540835)

[**Заключение 12**](#_Toc118540836)

[Используемые источники информации 13](#_Toc118540837)

**УП.11.02 Разработка приложения**

**База практики**

Практика проводилась на базе полигона вычислительной техники БПОУ ОО «Омский авиационный колледж имени Н.Е. Жуковского»

**Период прохождения практики**

Практика Разработка приложения по разработке программных продуктов проходила с **31.10.2022 г.** по **5.11.2022 г.** на полигоне вычислительной техники Омавиат.

**Основная задача практики**

Основная задача практики - формирование практических навыков по разработке кода программного продукта на основе готовой спецификации на уровне модуля, проведения тестирования программного модуля по определенному сценарию, использования основных методологий процессов разработки программного обеспечения, работы с объектами базы данных в конкретной СУБД.

**Виды работ**

* создание базы данных;
* разграничение данных пользователей;
* разработка приложения;
* документирование и презентация решений/

**Задание на разработку**

**Описание**

Требуется разработать программное решение для учета платежей физических лиц. Используя полученный программный продукт, физические лица могут вести учет своих платежей. Кроме того, появится возможность производить несложный анализ затрат в разрезе периодов или категорий. Также можно получать результаты анализа в печатной форме.

**Требования**

**Разграничение данных пользователей**

1. Программное решение должно быть многопользовательским;
2. Пароли должны обязательно храниться в зашифрованном виде;
3. Логин можно выбирать из выпадающего списка;

**Окно с основными данными**

1. В окне основных данных выводятся только данные текущего пользователя;
2. Данные должны помещаться на один экран по ширине;
3. Данные можно отображать за выбранный период (от даты до даты);
4. Данные можно отображать по одной из категорий;
5. Данные можно добавлять;
6. Данные можно удалять;
7. По отображаемым данным можно получить отчет;

**Добавление платежей**

1. При добавлении данных можно указывать категорию платежа;
2. При добавлении данных обязательно указывается назначение платежа;
3. Назначение платежа указывается на русском языке;
4. Назначение платежа состоит как минимум из 3 букв;
5. Едино разово можно совершить несколько одинаковых платежей;
6. Количество обязательно должно быть целым положительным числом;
7. Для платежа обязательно указывается цена в рублях;
8. Цена не может быть отрицательной;
9. Стоимость платежа рассчитывается как количество \* цена;
10. Дата платежа указывается как текущее время на компьютере;

**Удаление платежей**

1. При удалении данных система должна запрашивать подтверждение;
2. В подтверждении нужно выводить назначение платежа и другую значимую информацию;
3. Удаление должно сопровождаться звуковым или визуальным оформлением;

**Отчет**

1. Отчет может выводиться встроенными средствами, либо в распространенных программных продуктах (Microsoft Office Word, Microsoft Office Excel, Internet-браузер);
2. У отчета обязательно должен быть заголовок на первой странице;
3. В заголовке может быть указан период и категория платежей;
4. На каждой странице указывается ФИО пользователя и номер страницы;
5. Платежи группируются по категориям;
6. Внутри каждой категории платежи сортируются по дате;
7. В конце отчета выводится суммарная стоимость всех показанных платежей.

Описание организации данных

**Логическая модель данных**

Таблица 1. Описание основных объектов разрабатываемой системы

|  |  |
| --- | --- |
| Объект | Описание |
| Пользователь | Содержит информацию о пользователях, их имени, фамилии, отчестве, логине и пароле. |
| Продукт | Содержит информацию о продукте, его наименовании, цене и идентификатор категории |
| Категория | Содержит наименование категорий |
| Платежи | Содержит сведения платежей пользователя |

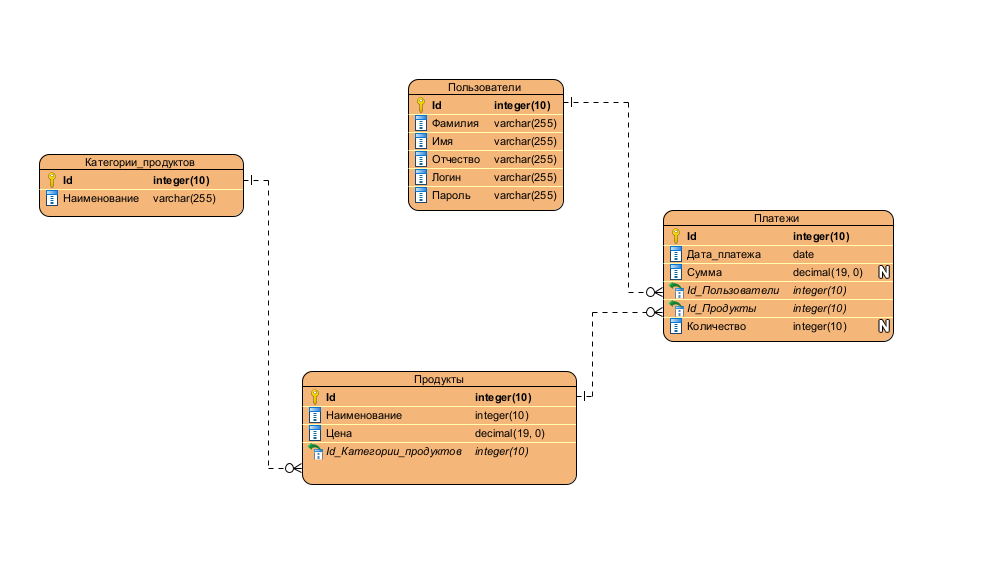


Рисунок 1. ER-диаграмма

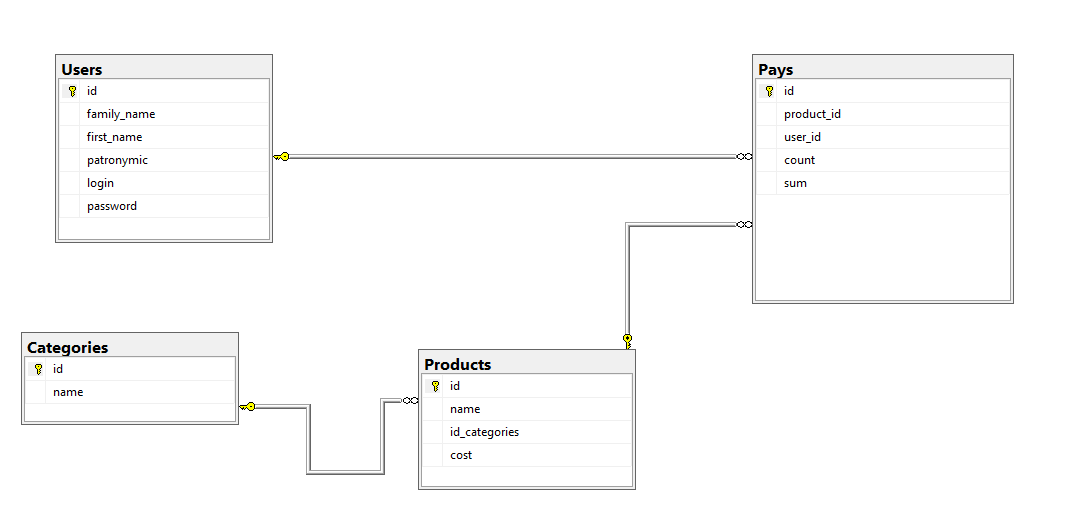
**Схема данных**

Рисунок 2. Схема данных, разработанная в СУБД

**Описание таблиц**

Таблица 2. Таблица Users

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Описание** | **Тип значения/поля** | **Кол-во знаков** |
| ID | Идентификатор пользователя | int |  |
| family\_name | Фамилия пользователя | nvarchar | 100 |
| first\_name | Имя пользователя | nvarchar | 100 |
| patronymic | Отчество пользователя | nvarchar | 100 |
| login | Логин пользователя | nvarchar | 100 |
| password | Пароль пользователя | nvarchar | 100 |

Таблица 3. Таблица Categories

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Описание** | **Тип значения/поля** | **Кол-во знаков** |
| ID | Идентификатор категории | int |  |
| name | Наименование категории | varchar | 100 |

Таблица 4. Таблица Products

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Описание** | **Тип значения/поля** | **Кол-во знаков** |
| ID | Идентификатор продукта | int |  |
| name | Наименование продукта | varchar | 100 |
| id\_categories | Идентификатор категории | int |  |
| coast | Цена продукта | money |  |

Таблица 5. Таблица Pays

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Описание** | **Тип значения/поля** | **Кол-во знаков** |
| ID | Идентификатор платежа | int |  |
| product\_id | Идентификатор продукта | int |  |
| user\_id | Идентификатор продукта | int |  |
| count | Цена продукта | int |  |
| sum | Сумма покупки | money |  |

**Описание интерфейса**

Форма авторизации (рис.3-4) – форма, предназначенная для авторизации пользователя. При вводе логина и пароля, приложение проверяет наличие такого пользователя в базе данных. При успешной авторизации пользователь переходит на главную страницу с данными, где отображены все его платежи. В ином случае система выводит ошибку авторизации.

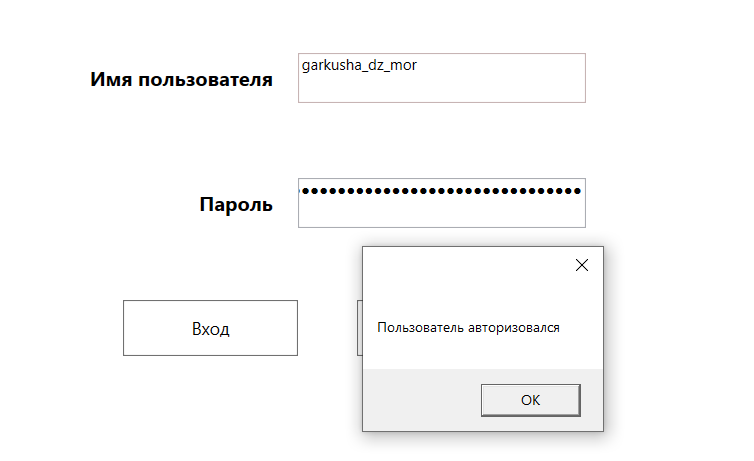
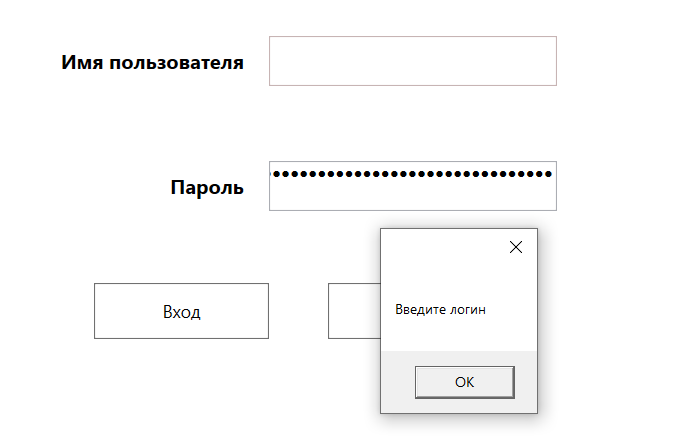


Рисунок 4. Форма авторизации. Пользователь неуспешно прошел авторизацию

Рисунок 3. Форма авторизации. Пользователь успешно прошел авторизацию

Листинг 1. Описание логики авторизации

private void btn\_input\_Click(object sender, RoutedEventArgs e)

{

if (txt\_name.Text.Length > 0) // проверяем введён ли логин

{

if (password.Password.Length > 0) // проверяем введён ли пароль

{ // ищем в базе данных пользователя с такими данными

DataTable dt\_user = mainWindow.Select("SELECT \* FROM [dbo].[Users] WHERE [login] = '" + txt\_name.Text + "' AND [password] = '" + password.Password + "'");

DataTable id = mainWindow.Select("Select \* From [dbo].[Users]");

if (dt\_user.Rows.Count > 0) // если такая запись существует

{

MessageBox.Show("Пользователь авторизовался");

mainWindow.MainFrame.Navigate(new MainTable());

}

else MessageBox.Show("Пользователь не найден"); // выводим ошибку

}

else MessageBox.Show("Введите пароль"); // выводим ошибку

}

else MessageBox.Show("Введите логин"); // выводим ошибку

}

Форма с основными данными (рис.5) – форма, которая содержит информацию о платежах пользователя. С помощью данной формы пользователь может осуществлять фильтрацию своих данных. А также может перейти на формы добавления и удаления платежей.

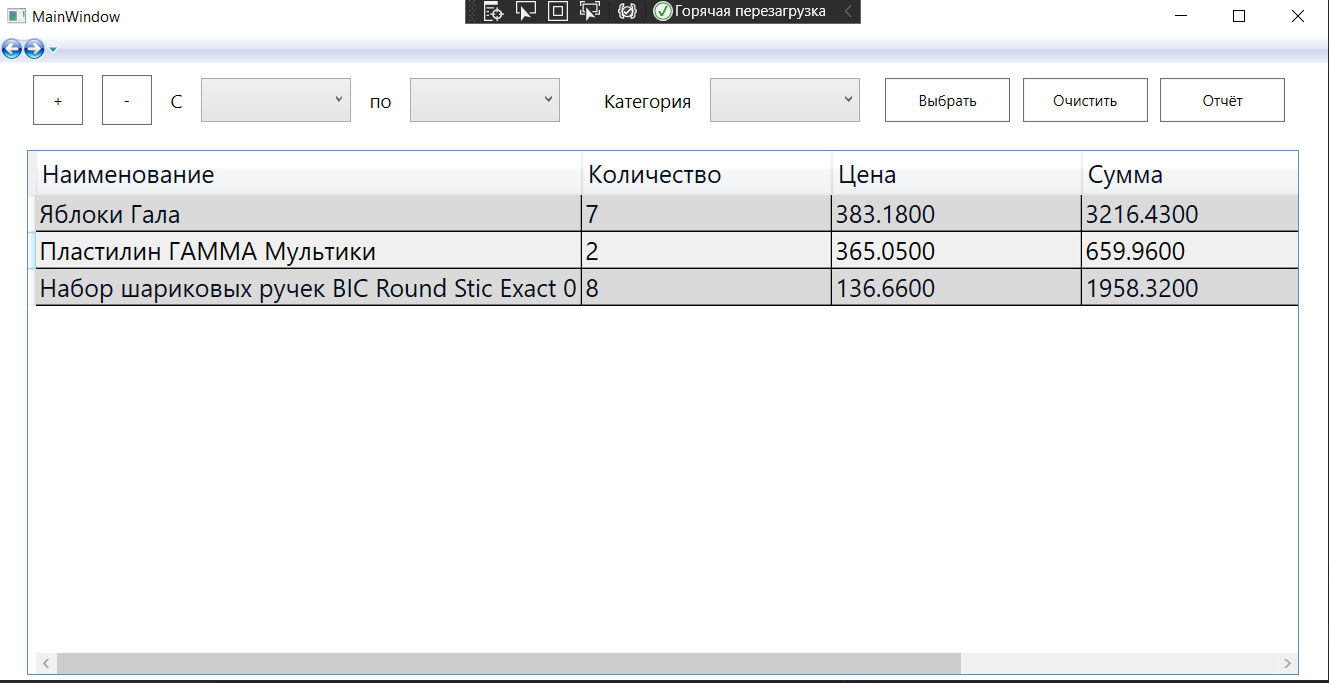


Рисунок 5. Форма с основными данными

Листинг 2. Xml-код для добавления данных в DataGrid

<DataGrid x:Name="grid\_categ" Grid.Row="1" Margin="10" IsReadOnly="True" AutoGenerateColumns="False" Foreground="#FF020C1D" Grid.RowSpan="3" FontSize="20" MinColumnWidth="200" MaxColumnWidth="500" CanUserReorderColumns="False" RowBackground="#FFDADADA" AlternatingRowBackground="White" Background="White" >

<DataGrid.Columns>

<DataGridTextColumn Header="Наименование" Binding="{Binding Наименование}"></DataGridTextColumn>

<DataGridTextColumn Header="Количество" Binding="{Binding Количество}"></DataGridTextColumn>

<DataGridTextColumn Header="Цена" Binding="{Binding Цена}"></DataGridTextColumn>

<DataGridTextColumn Header="Сумма" Binding="{Binding Сумма}"></DataGridTextColumn>

<DataGridTextColumn Header="Категория" Binding="{Binding Category.name}"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

Листинг 3. Связь с таблицей базы данных для добавления в DataGrid

public MainTable()

{

InitializeComponent();

grid\_categ.ItemsSource = Расчет\_платежейEntities.GetContext().products\_user.ToList();

}

Форма добавления платежей (рис.6) – отображает список категорий, назначение платежа, количество и цену продукта.

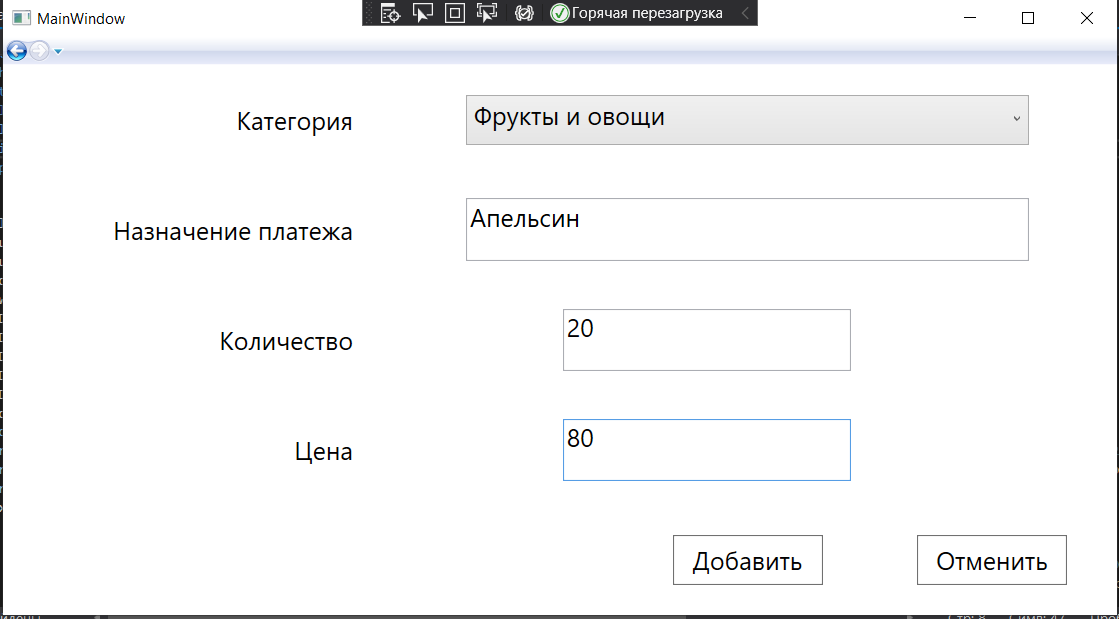


Рисунок 6. Форма добавления платежей

Листинг 4. Описание логики добавления

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(txt\_product.Text))

error.AppendLine("Укажите продукт");

if (string.IsNullOrWhiteSpace(txt\_cost.Text))

error.AppendLine("Укажите количество");

if (string.IsNullOrWhiteSpace(txt\_count.Text))

error.AppendLine("Укажите цену");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

var category = cmb.SelectedItem as Category;

pays.Категория = category.id;

pays.Наименование = txt\_product.Text;

pays.Количество = int.Parse(txt\_count.Text);

pays.Цена = int.Parse(txt\_cost.Text);

pays.Сумма = pays.Цена \* pays.Количество;

Instance.db.products\_user.Add(pays);

Instance.db.SaveChanges();

NavigationService.Navigate(new Uri("/MainTable.xaml", UriKind.Relative));

Форма удаления платежей (рис.7) – предоставляет пользователю возможность удалить по желанию строку таблицы платежей.

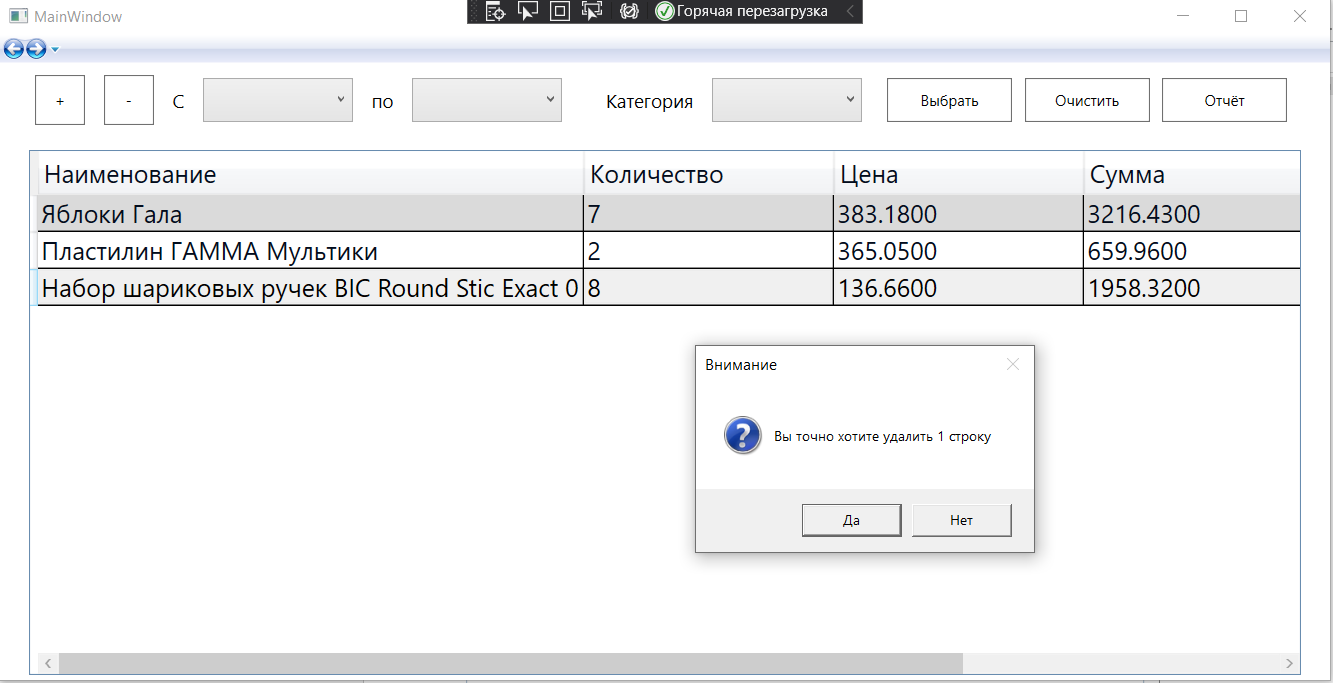


Рисунок 7. Форма удаления платежей

**Заключение**

В результате учебной практики был разработан программный продукт для учёта платежей физических лиц.

Разработка проходила в несколько этапов, были проведены тестирования программного модуля по определенному сценарию, использование основных методологий процессов разработки программного обеспечения, работы с объектами базы данных в конкретной СУБД.

# Используемые источники информации

1. Шарп Джон Microsoft Visual C#. Подробное руководство. 8-е изд. — СПб.: Питер, 2017. — 848 с.: ил.
2. Новиков Б.А., Домбровская Г.Р. Настройка приложений баз данных. – БХВ – Петербург, 2012. – 240 с.
3. Голощапов А.Л. Microsoft® Visual Studio 2010 – СПб.: БХВ – Петербург, 2011. – 544 с.+ CD-ROM – (В подлиннике)
4. Постолит А. В. Visual Studio .NET: разработка приложений баз данных. – БХВ – Петербург, 2003. – 538 с.
5. Культин Н. Б. Microsoft Visual C# в задачах и примерах. – БХВ – Петербург, 2009. – 320 с
6. Р. Стюард. Практическое руководство по доступу к данным: Пер. с англ. – СПб.: БХВ - Петербург, 2013. – 304 с.: ил.