ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | Н.В. Богословская |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5 |
| Автономная модель доступа к данным |
| по курсу: ТЕОРИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3641 |  |  |  |  |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2019

**Цель работы:** Рассмотреть приемы анализа и представления данных, а также способы обработки таблиц базы данных и обеспечить их взаимодействие с сервером баз данных, также научиться использовать класс DataAdapter, компонент DataSet, а также методы Fill() и Update().

**Результаты, полученные в ходе выполнения работы:**

***WinForm:***

1. ***Загрузка данных в объект DataSet:***

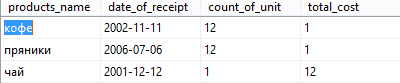


Рис. 1- Вид таблицы с данными на сервере MS SQL Server.

Первым шагом построим интерфейс с помощью которого будем осуществлять выбор нужных нам данных: с помощью DataGridView выведем таблицу с данными . На рисунке 2 показан результат, действительно таблицы идентичны.

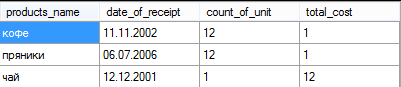


Рис. 2- Отображение DataSet в DataGridView windows-формы

С помощью Метода ***Fill()*** заполним объект *DataSet* данными, полученными от источника данных, а с помощью метод ***Update()*** обновим источник данных в соответствии с изменениями данных в таблицах объекта *DataSet*.

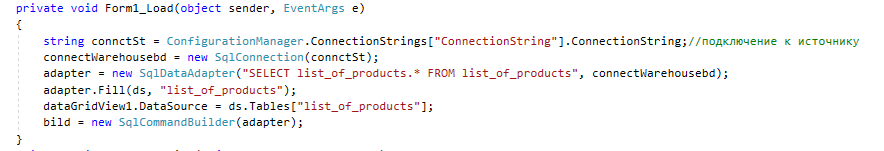


Рис. 3- Код для загрузки данных в объект DataSet в windows-форме

На рисунке 3 продемонстрирован программный код, где реализованы все выше приведенныее методы, и произведен вывод данных в форму. А информация содержащаяся в конфигурационном файле определяет подключение к базе данных.

1. ***Возможности объекта DataTable:***
2. *Добавлние строк методом NewRow()*

Для того, чтобы добавить новые значения в таблицу- в форме, создадим дополнительные элементы: *button1*, и *TextBox1, TextBox2,* и т.д. Используя все названия столбцов которые будем заполнять.

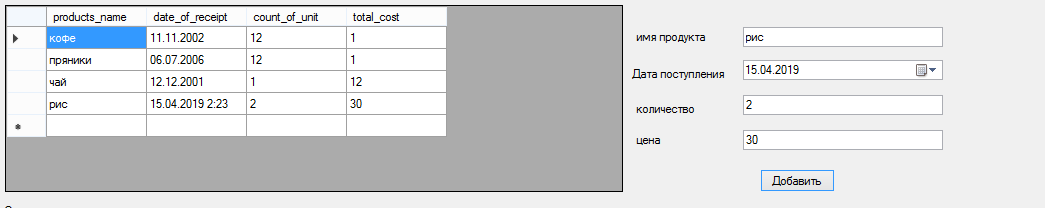


Рис.4- Вид таблицы формы с элементами

Далее на рисунке 5 опишем каждый элемент кодом, который и будет заполнять строку.

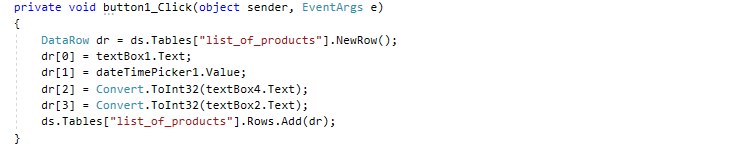


Рис.5- Код для добавления строки в таблицу

На рисунке 6 показано что, при заполнении пустых полей данными: новая строка появляется в таблице.

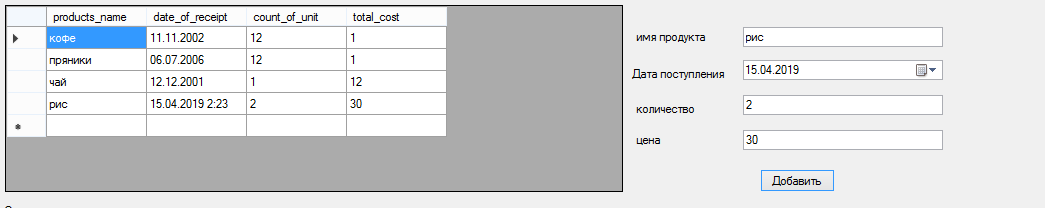


Рис.6- Результат добавления строки в таблицу

1. *Удаление строк методом Delete():*

Итак, далее будем удалять/восстанавливать строки из таблицы. Для этого добавим необходимые элементы. Интерфейс и результат показан на рисунке 7

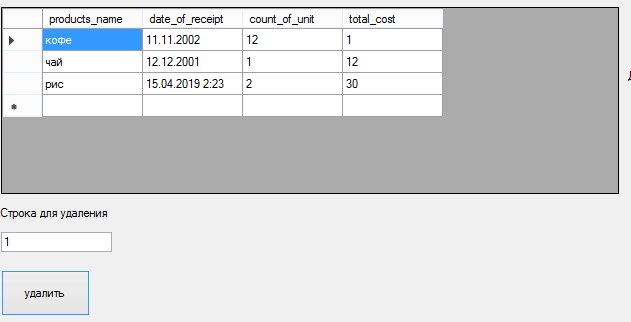


Рис.7-Результат удаления «1» строки из таблицы

После восстановим эту же строку. Результат показан на рисунке 8.

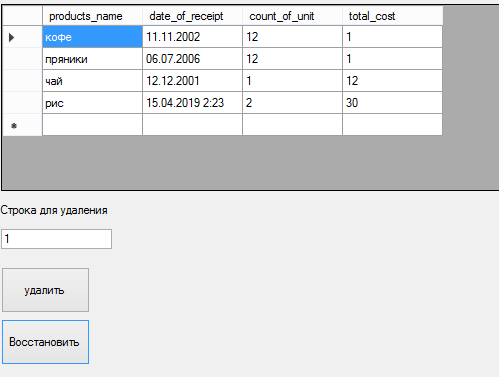


Рис.8- Восстановление строки в таблице.

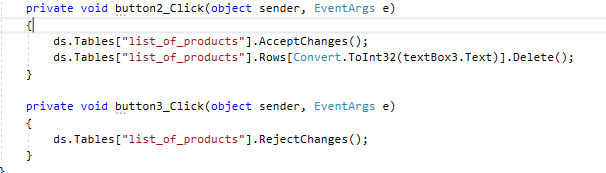


Рис.9- Код для удаления/восстановления строки в таблицу

***WebForm:***

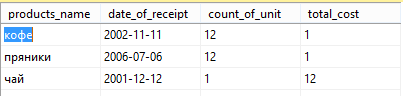


Рис.10- Вид таблицы с данными на сервере MS SQL Server.

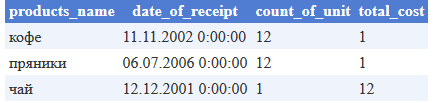


Рис.11- Результат выполнения программы в веб-форме

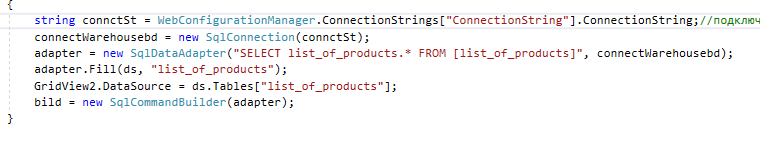


Рис.12- Код программы.

Отличия *DataGridView* от *GridView:*

Элемент *GridView* является преемником элемента управления *DataGrid*. Как и элемент управления *DataGrid*, элемент управления *GridView* предназначен для отображения данных в таблице HTML. При привязке к источнику данных элементы управления *DataGrid* и *GridView* каждый из них отображает строку из *DataSource* в виде строки в выходной таблице.

Оба элемента управления *DataGrid* и *GridView* выводятся из класса. Хотя он имеет сходную объектную модель с моделью *DataGrid*, элемент управления GridView также имеет ряд новых функций и преимуществ над элементом управления *DataGrid*, который включает в себя:

* Более богатые возможности разработки.
* Улучшены возможности связывания источника данных.
* Автоматическая обработка сортировки, подкачки, обновления и удаления.
* Дополнительные типы столбцов и операции столбца "дизайн-время".
* Пользовательский пользовательский интерфейс пользовательского интерфейса (UI) с свойством PagerTemplate.

Различия между элементом управления *GridView* и элементом управления *DataGrid* включают:

* Поддержка пользовательских пейджинга.
* Различные модели событий.

Сортировка, пейджинг и редактирование данных на месте требуют дополнительного кодирования при использовании элемента управления *DataGrid*. Элемент управления *GridView* позволяет добавлять функции сортировки, подкачки и редактирования без написания кода. Вместо этого вы можете автоматизировать эти задачи вместе с другими распространенными задачами, такими как привязка данных к источнику данных, путем установки свойств элемента управления.

**Выводы:** выполняя данную работу, я научиться использовать класс DataAdapter, компонент DataSet, а также методы Fill() и Update(). Открыл для себя возможности обработки таблиц с данными с помощью методов которые включают данные методы и класс, научился взаимодействовать с Microsoft SQL Server, также были освоены возможности технологии доступа к данным в информационных системах-ADO.NET.Проекты были написаны в win-форме и web-форме Программы были успешно реализованы, собраны и отлажены.