

Лабораторная работа №5. Импорт и экспорт данных

Часть I. Простые текстовые файлы

Задание 1: Дополните окно «Управление заказчиками» возможностью сохранения ФИО и адресов электронной почты заказчиков в текстовый файл.

1. В классе формы «Управление заказчиками» подключите пространство имен **System.IO** для работы с функциями ввода-вывода.
2. При нажатии на кнопку «Вывод в файл» выполните следующие действия:

```
try
{
    string fileName = "customers.txt"; //имя файла
    //файловый поток создается для записи
    FileStream fsOutput = new FileStream(fileName, FileMode.Create,
    FileAccess.Write);
    //объект для записи в файл
    StreamWriter swOutput = new StreamWriter(fsOutput);

    //вывод строки в файл
    swOutput.WriteLine("Список заказчиков");

    //выполняем обход таблицы
    for (int i = 0; i < dataGridView1.Rows.Count; i++)
    {
        //заносим значение ФИО из 1го столбца таблицы
        swOutput.Write(dataGridView1.Rows[i].Cells[1].Value.ToString());
        swOutput.Write(" ");
        //заносим значение email из 3го столбца таблицы
        swOutput.Write(dataGridView1.Rows[i].Cells[3].Value.ToString());
        //добавляем перенос строки
        swOutput.WriteLine();
    }
    swOutput.WriteLine("Итого записей: " + (dataGridView1.Rows.Count - 1).ToString());
    swOutput.Close(); //закрытие потока
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Произошла ошибка!");
}
```

3. Запустите программу и проверьте правильность ее работы.

Часть II. Стандарт CSV

Задание 2: Дополните окно «Управление складом» возможностью выгрузки текущего состояния склада в файл формата CSV.

1. Для формы «Управление складом» подключите пространство имен для работы с функциями ввода-вывода.
2. При нажатии на кнопку «Вывод в файл» выполните следующие действия:

```
try
{
    string fileName = "store.csv"; //имя файла
    //файловый поток создается для записи
    FileStream fsOutput = new FileStream(fileName, FileMode.Create, FileAccess.Write);
    //поток для записи в файл
    StreamWriter swOutput = new StreamWriter(fsOutput);

    //вывод шапки таблицы
    swOutput.WriteLine("Name;Count");

    //выполняем обход таблицы
    for (int i = 0; i < dataGridView1.Rows.Count; i++)
    {
        //заносим Название товара
        swOutput.Write(dataGridView1.Rows[i].Cells[1].Value.ToString());
        //добавляем разделитель
        swOutput.Write(";");
        //заносим значение Количества
        swOutput.Write(dataGridView1.Rows[i].Cells[2].Value.ToString());
        //добавляем перенос строки
        swOutput.WriteLine();
    }
    swOutput.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Произошла ошибка!");
}
```

4. Запустите программу и проверьте правильность ее работы.
5. Дополните решение возможностью выбора каталога сохранения файла.

Задание 3: Дополните окно «Управление складом» возможностью загрузки информации о поставках на склад из файла формата CSV (AddToStore.csv). В данном файле данные представлены следующим набором полей: название товара, количество, дата поставки.

1. В классе для работы с базой данных создайте метод получения кода товара по его названию:

```
static public string GetProductId(string productName)
{
    msCommand.CommandText = @"SELECT Product FROM Assortiment WHERE Name='" +
                             productName + "'";
    object result = msCommand.ExecuteScalar();

    //если результат запроса не пустой
    if (result != null)
    {
        return result.ToString(); //возвращаем полученное значение
    }
}
```

```

else
    return null; //иначе возвращаем пустое значение
}

```

2. Для кнопки «Загрузка поставок из файла» выполните следующие действия:

```

try
{
    string fileName = "AddToStore.csv"; //имя файла
    FileStream fsInput = new FileStream(fileName, FileMode.Open, FileAccess.Read);
    //поток для чтения из файла
    StreamReader srInput = new StreamReader(fsInput);

    int count = 0; //счетчик количества записей
    string line = srInput.ReadLine(); //считываем заголовок
    //пока из файла не считана пустая строка
    while ((line = srInput.ReadLine()) != null)
    {
        string[] record = line.Split(';'); //разделяем строку

        //получаем код товара по названию
        record[0] = DBConnection.GetProductId(record[0]);

        if (record[0] != null) //если код товара получен верно
        {
            //добавляем запись на склад
            DBConnection.AddToStore(record[0], record[1], record[2]);
            count++; //увеличиваю счетчик
        }
    }
    srInput.Close();
    DBConnection.GetStoreList(); //обновляем набор данных
    MessageBox.Show("Добавлено записей: " + count.ToString(), "Сообщение");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Произошла ошибка!");
}
}

```

3. Запустите программу и проверьте правильность ее работы.

Часть III. Создание отчетов

Задание 4: Для заказчиков добавьте функцию вывода на печать ассортимента продукции компании.

1. В среде **Workbench** создайте представление, объединяющее таблицы **Ассортимент** и **Категории** (поля **Наименование**, **Срок годности**, **Стоимость** и **Категория**).
2. В **Visual Studio** добавьте источник данных:
 - 1) выберите пункт **Проект – Добавить новый источник данных**;
 - 2) в **Мастере** настройки источника данных выберите **тип – База данных**;
 - 3) укажите **тип модели – Набор данных**;
 - 4) создайте подключение к БД со следующими параметрами:
Источник данных – MySQL Database;
Server name – localhost;
User name – root;
Password – qwerty;
Database name – MurmanRyb.
 - 5) далее выберите в качестве объекта базы созданное представление.
3. Создайте макет отчета:
 - 1) выполните команды **Проект – Добавить компонент – Мастер отчетов**;
 - 2) выберите **Источник данных**, созданный в предыдущем пункте;
 - 3) далее в область **Группы строк** перенесите поле **Категория**;
 - 4) в область **Значения** перенесите поле **Наименование**;
 - 5) далее снимите все флаги в параметрах макета;
 - 6) далее выберите цветовое оформление.
6. В открывшемся проекте отчета добавьте столбцы в таблицу:
 - 1) в контекстном меню таблицы выберите пункт **Вставить столбец – Справа**;
 - 2) щелкните в появившейся ячейке мышью и с помощью пиктограммы в ее верхнем правом углу выберите имя отображаемого поля – **Срок годности**;
 - 3) аналогично добавьте столбец с полем **Стоимость**;
 - 4) замените названия столбцов на русскоязычные.
7. Добавьте вывод количества записей в таблице:
 - 1) в контекстном меню поля **Категория** выберите пункт **Добавить итог – Снизу**;
 - 2) в качестве итога выведите значение **[Count()]**.
8. С **Панели компонентов** вынесите в отчет **Текстовое поле** для отображения заголовка отчета: «Ассортимент товаров компании МурманРыб».
9. Отобразите дату формирования отчета:
 - 1) перейдите на вкладку **Данные – Встроенные поля**;

- 2) перенесите в проект отчета компонент **Время выполнения**.
4. Создайте форму для отображения отчета.
5. Вынесите на форму компонент типа **Report Viewer**.
6. С помощью пиктограммы в верхнем правом углу компонента выполните привязку к родительскому контейнеру.
7. Там же в поле Выбрать отчет укажите созданный вами макет.
8. В окне «Просмотр ассортимента» создайте кнопку «Печать», при нажатии на которую откройте форму с отчетом.
9. Запустите программу и проверьте правильность ее работы.

Часть VI. Задания для самостоятельной работы

1. Добавьте на форму «Управление пользователями» возможность сохранения списка пользователей в формате CSV.
2. Дополните программу функцией формирования отчета о продажах за указанный месяц.