Семестровая работа по курсу «Документоведение».

Василов Артур,

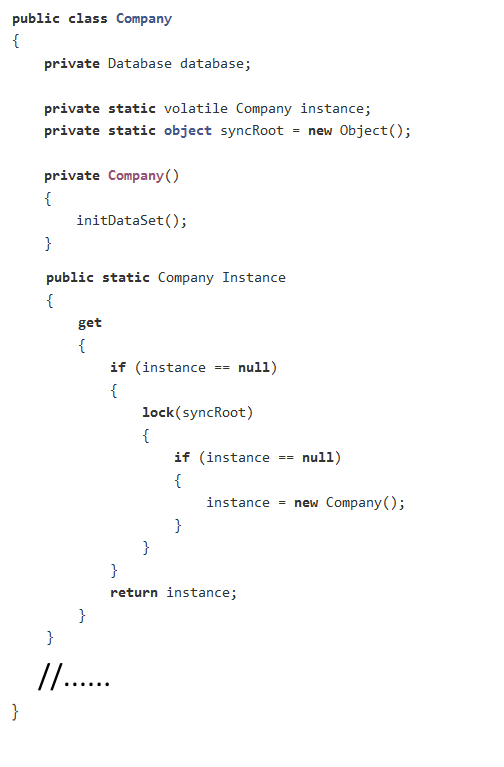
Группа 09-311,

2014г.

Разработанное приложение имитирует систему документооборота аутсорсинговой компании. Для примера используются 4 документа – договор о найме сотрудника, проектный договор, отчет сотрудника и финансовая декларация (в простой форме – доход-расход-прибыль).

Рассмотрим, как была реализована работа с системой хранения документов. Примечание: код будет приводиться фрагментами, полная версия доступна в гит-репозиторие. [Ссылка.](https://github.com/ArturVasilov/PracticeITcs/tree/master/DocumentsSecurity/DocumentsSecurity)

Во-первых, для удобной работы нам надо создать класс, для обработки всех запросов, операций поиска, удаления и добавления документов. Этот класс должен иметь только один экземпляр и содержать все необходимые методы. Таким образом, этот класс должен реализовывать паттерн Singleton, а также частично являться DAO (data access object). Я использую ленивую инициализацию и механизм двойной блокировки. Также я добавил несколько методов, которые можно выполнять из других классов (я опускаю комментарии в коде, так как в комментировании нуждается только плохой код):

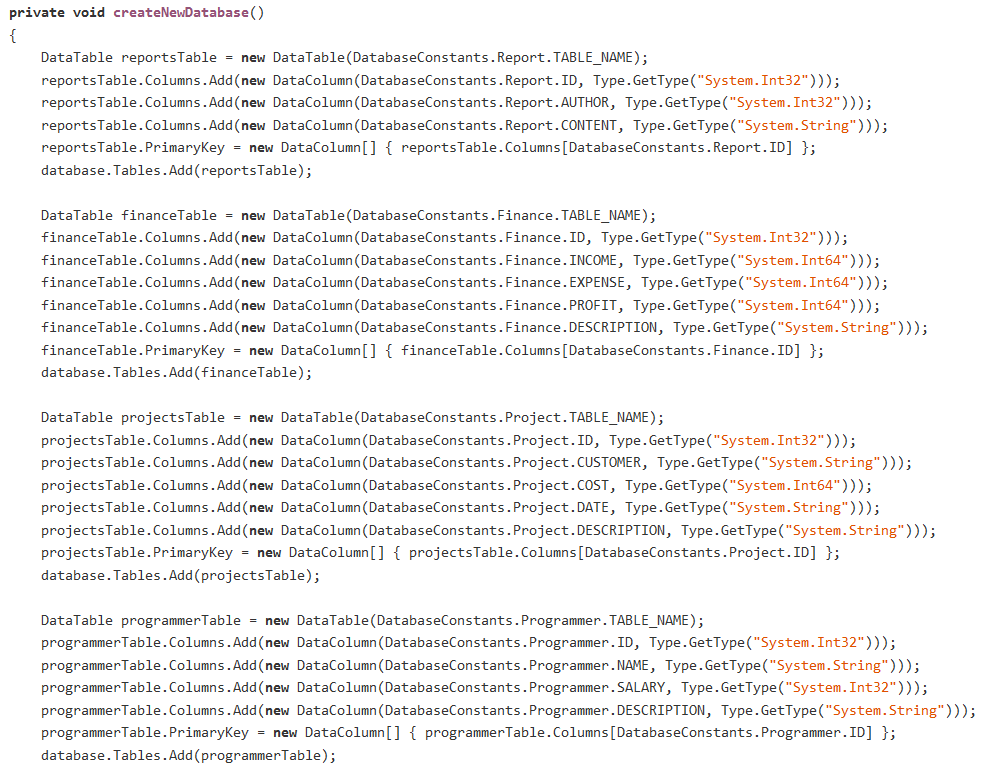


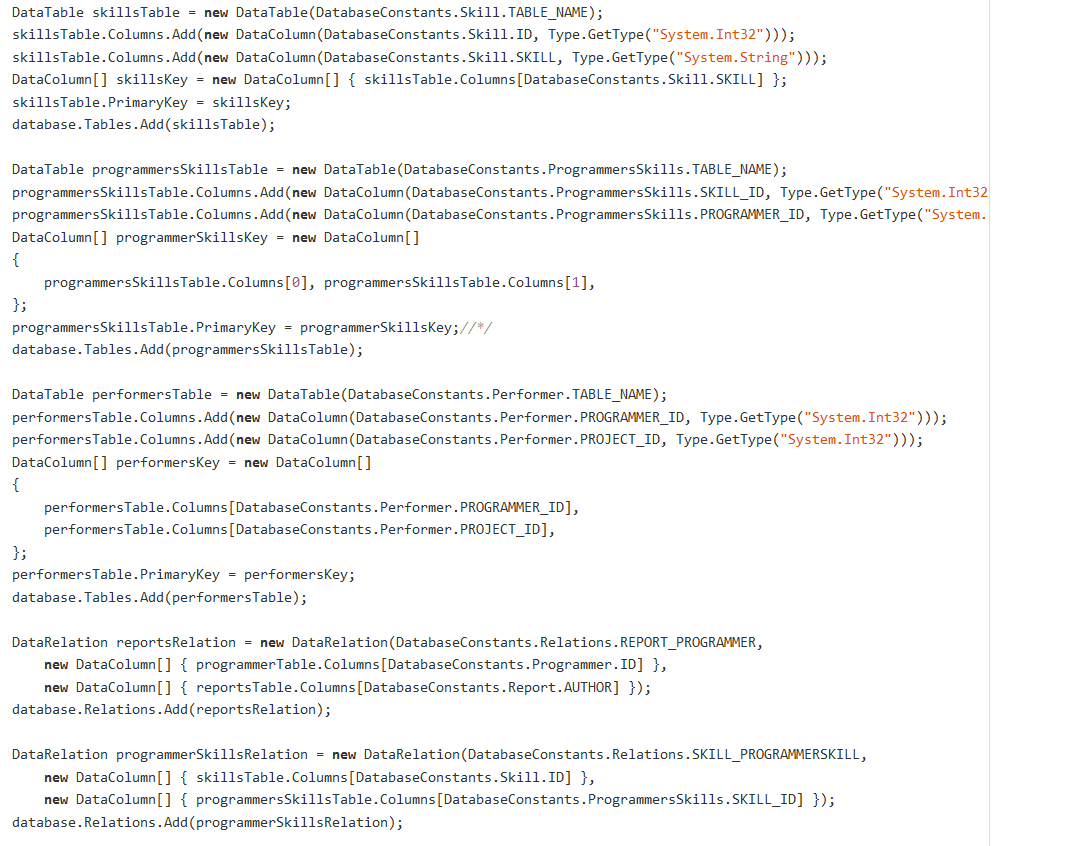
Как видно, все методы вызывают соответствующие методы в классе Database. Этот класс уже непосредственно выполняет работу с классрм DataSet.

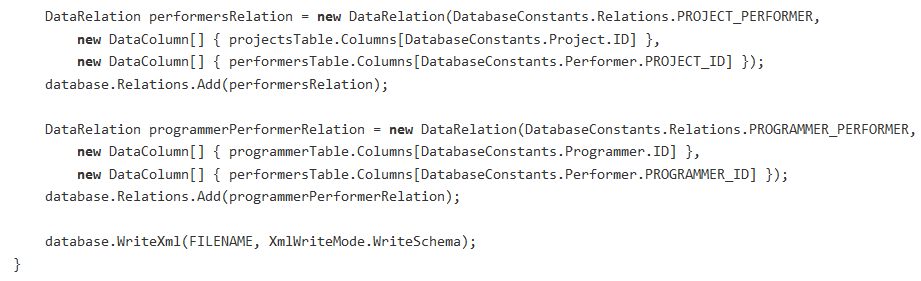
Класс DataSet (основные фрагменты кода):



Метод создания базы, создает все таблицы и связи:

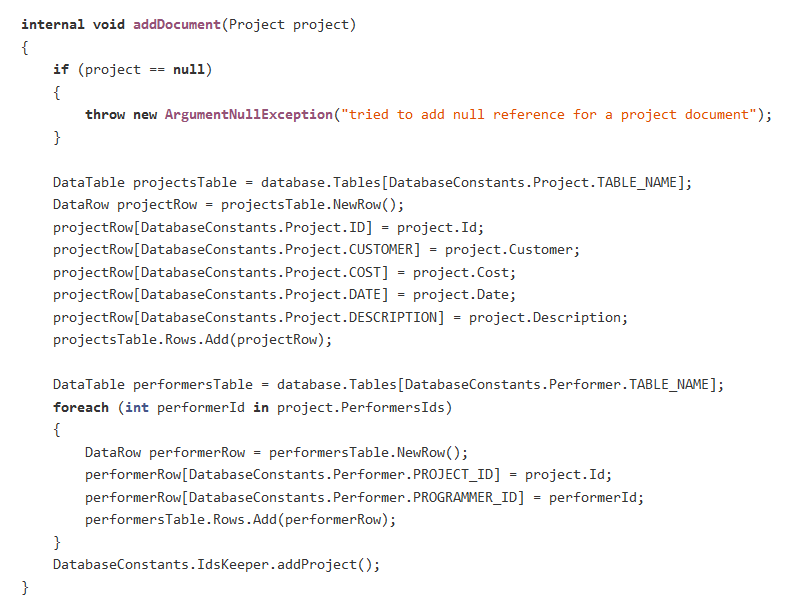




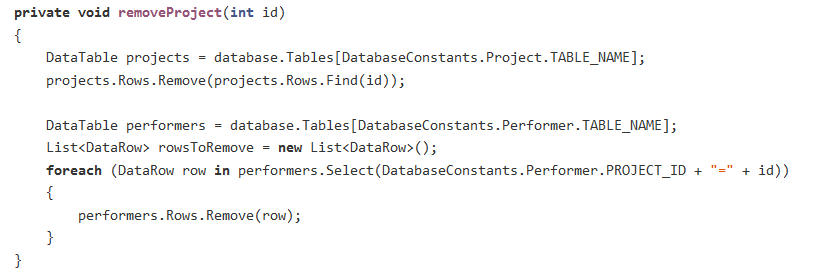


Рассмотрим методы добавления / удаления, редактирования, получения на примере проектного договора.

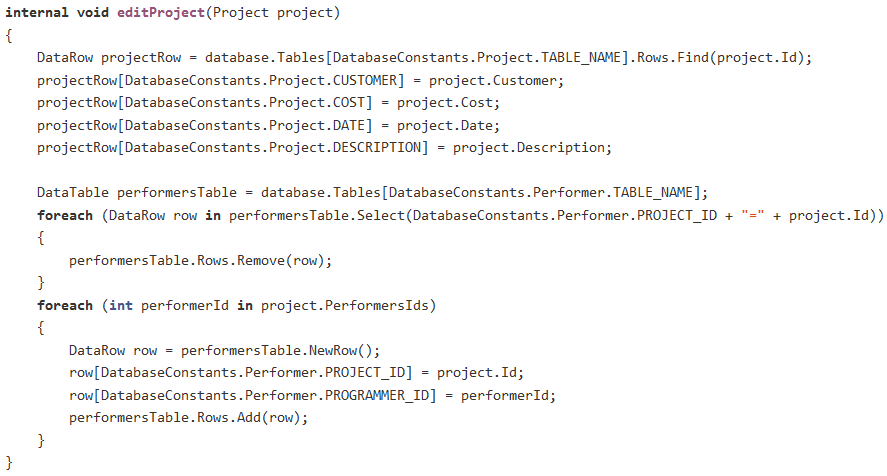
Добавление:



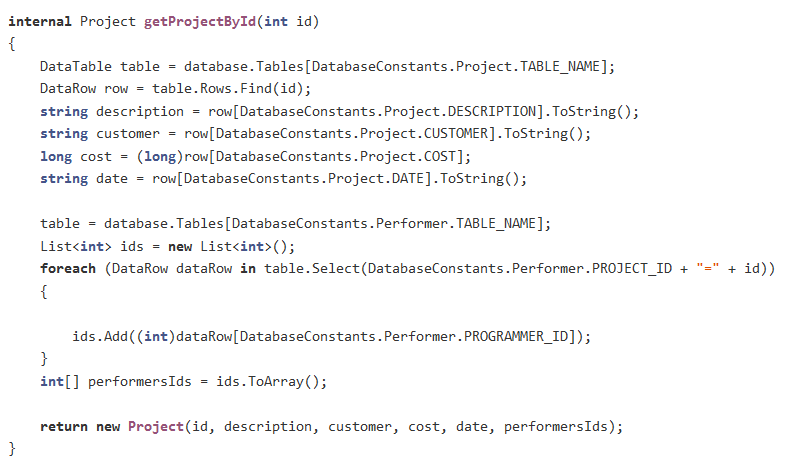
Удаление:



Редактирование:

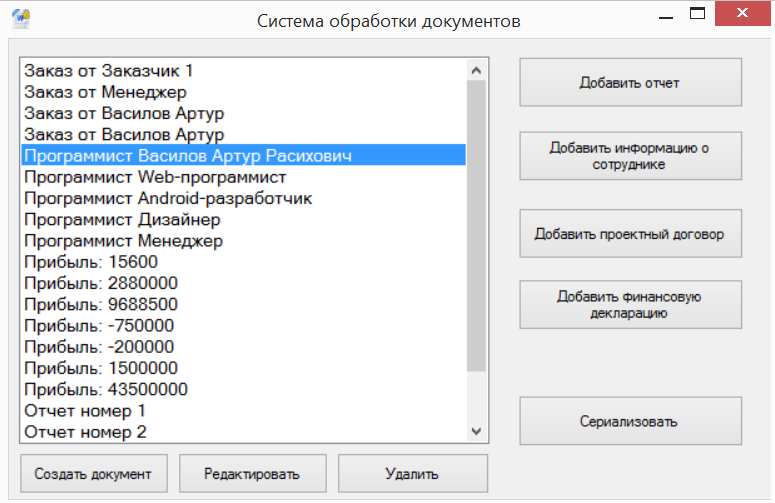


Получение:

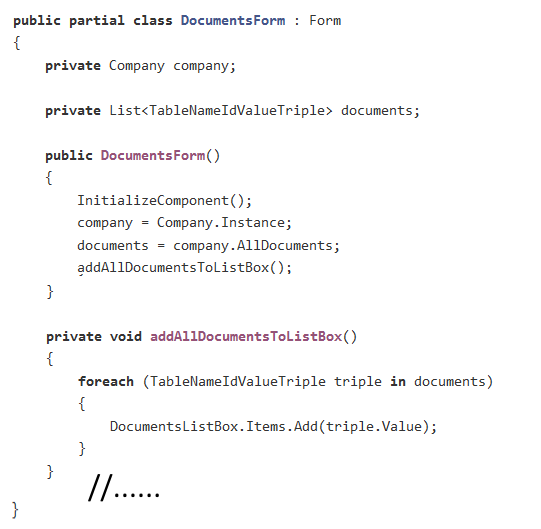
1

Для каждого документа создается отдельный класс, который наследуется от базового класса Document, который содержит общие поля и описание документа (т.е. все те поля, который должны содержаться во всех документах). Эти классы достаточно примитивны и просто содержат все нужные поля, рассматривать их не имеет особого смысла.

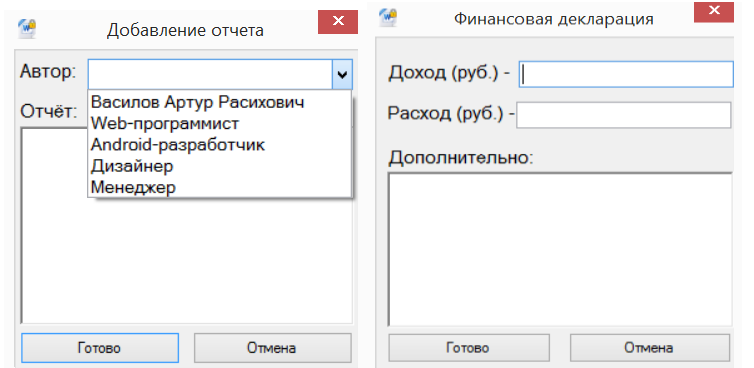
Приложение должно быть удобно в использовании. Для этого средствами Windows Forms был создан следующий интерфейс пользователя:



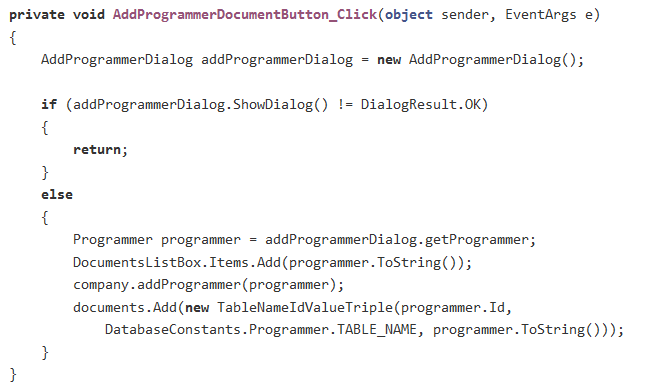
При запуске приложения все документы считываются из базы и добавляются в компонент ListBox на форме. Для этого можно выполнить примерный код:



Создание документов выполняется с помощью соответствующих кнопок. По нажатию на них вызываются соответствующие диалоговые окна:



Код обработки нажатия на кнопку:



Сам класс AddProgrammerDialog считывает значение нужны полей по завершению ввода и генерирует соответствующий документ. Для редактирования используется этот же диалог с уже заполненными полями:

