

EPAM Systems, RD Dep.

Homework Task

11 – ADO.NET

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
1.0	Initial version	Alexander_Kuznetsov1	02.03.2012		
2.0		Nikolay Piskarev	29.05.2017		

Формулировка задания

Нашей задачей является разработка части DAL (Data Access Layer) к базе Northwind. При этом необходимо соблюдать ряд требований:

- DAL предоставляет объектный интерфейс. Т.е. данные которые он принимает и выдает на вход являются обычными [POCO \(plain old CLR object\)](#) объектами
- DAL должен быть спроектирован таким образом, чтобы его можно было легко заменить на fakes или mocks при модульном тестировании бизнес-слоя
- DAL должен поддерживать подключение к произвольной базе (строки подключения нигде не хардкодятся).
- DAL должен быть спроектирован максимально в provider independent стиле. При этом должна быть оставлена возможность заменить некоторые методы на специфические для конкретной СУБД

Кроме того, при работе с базой Northwind мы будем исходить из следующего:

- Заказ (таблица Orders) может находиться в следующих состояниях:
 - Новый (не отправленный), если его поле OrderDate == NULL
 - В работе (не выполненный), если поле ShippedDate == NULL
 - Выполненный, если ShippedDate != NULL

О самом задании

Наше гипотетическое приложение – это рабочее место оператора, однако никакого UI мы создавать не будем, а для выполнения работы требуется:

- Создать сам DAL
- Создать набор тестов, демонстрирующих его (DAL) работу. Тесты работают непосредственно с базой, но отката базы на исходное состояние не требуется.

Задание 1

Реализуйте возможность через DAL управлять заказами:

1. Показывать список введенных заказов (таблица [Orders]). Помимо основных полей должны возвращаться:
 - a. Статус заказа в виде Enum поля.
2. Показывать подробные сведения о конкретном заказе, включая номенклатуру заказа (таблицы [Orders], [Order Details], и [Products], требуется извлекать как Id, так и название продукта)
3. Создавать новые заказы.
4. Менять статус заказа. Для реализации этого пункта предлагается сделать специальные методы (именование выбирайте сами):
 - a. Передать в работу: устанавливает дату заказа
 - b. Пометить как выполненный: устанавливает ShippedDate
5. Получать статистику по заказам, используя готовые хранимые процедуры:
 - a. CustOrderHist
 - b. CustOrdersDetail.

Требования к оформлению

1. Для каждой домашней работы необходимо создать собственную папку в корне репозитория.
2. Каждое задание должно представлять собой отдельный проект (project) в рамках общего решения (solution).
3. Правила именования:

Все названия и имена должны быть переведены на английский язык. Именование транслитерацией не допускается.

 - a. Маска именования репозитория: EPAM_Ext_Lab_<Квартал старта группы>_<Год>_<Фамилия_Имя>.
Например, EPAM_Ext_Lab_Q2_2017_Ivanov_Ivan
 - b. Маска именования решения: HMT_<Порядковый номер>.
Например, HMT_01, HWT_11.
 - c. Маска именования проектов: <Задача><Номер задачи>.
Например, Task01, Task12.
 - d. Каждое задание должно быть оформлено следующим образом:
 1. Формулировка задания в виде комментария (просто копировать-вставить из этого файла).
 2. Код, решающий задачу.
4. Написать письмо преподавателю, информирующее о том, что домашняя работа выполнена. Тема письма должна выглядеть так: [STYYYY_N]Фамилия_Имя_TaskNN. Здесь ST – сокращённое наименование программы (Students Training), YYYY – номер года, N – номер группы, NN – порядковый номер задания в виде двух цифр, напр. 01, 05, 12 и т.д.
Например, ST2017_1 Иванов Иван Task01.

Рекомендации к оформлению кода

1. При добавлении нового файла желательно вверху файла добавить блок с комментарием, для чего предназначен этот файл.

2. При добавлении метода в блоке `<summary></summary>` описать предназначение метода, его входных/выходных параметров и возвращаемых значений.

3. Использовать утилиту StyleCop для проверки правил оформления кода. Перед отправкой задания на проверку запускать StyleCop и исправлять все предупреждения, возникающие в секции Warning Visual Studio. Предупреждения об отсутствии документации можно игнорировать (либо настроить StyleCop таким образом, чтобы он не проверял документирование кода).