

## SQLite. Семинар 2 (продолжение)

Вы наверняка, справившись с написанием SQL-запросов, обратили внимание, что C#-проект не свободен от недостатков. Этот семинар будет посвящен работе над улучшением этого проекта.

### Задача 1. Работа над соединением с БД.

Вы, вероятно, заметили, что открытие соединения с БД происходит в методе Main() в то время, как вся остальная работа с БД осуществляется внутри класса `SQLUtilities`. Однако, соединение необходимо открывать на максимально короткий срок! Поэтому Вам необходимо модифицировать код так, чтобы соединение открывалось внутри `SQLUtilities` (в качестве параметра в методы этого класса можно передавать строку соединения с БД `"Data Source=olympics.db"`). Не забудьте обернуть в using объект типа `SQLiteCommand`.

### Задача 2. Работа над ~~ошибками~~ исключениями.

Если произойдет ошибка при работе с БД – возникнет необработанное исключение. Обработайте исключения, которые могут возникнуть в вашей программе (при работе с БД). Помните, что обработка исключений – задача кода, использующего библиотеку.

### Задача 3. Работа над switch.

Метод `ExecuteTask` явно можно написать с использованием коллекций вместо switch. Сделайте задуманное.

### Задача 4. Переходим к ООП, наконец.

Если бы не метод `FillDB`, который объявлен нестатическим (хотя он и не использует никаких данных объекта), класс `SQLUtilities` вполне можно было бы объявить статическим. Однако в модуле, посвященном работе с ООП это требует срочной переработки! Выполните рефакторинг кода так, чтобы статический класс `SQLUtilities` превратился в класс `SQLiteOlympicsStorage`, объект которого умел бы все тоже, что и старый `SQLUtilities`. Не забудьте перенести в код класса все SQL-операторы для решения задач (1-6).

### Задача 5. А как же записи?!

Воодушевившись сделанным, вы вспоминаете, что недавно на лекции проходили материал посвященный record – ссылочным типам, которые часто используются для представления неизменяемых объектов, полученных из БД. Вы решаете создать типы-записи для всех пяти (по количеству таблиц в БД) типов объектов, чтобы затем можно было в классе `SQLiteOlympicsStorage` написать методы, осуществляющие чтение “объектов” (как списка, так и по одному) из БД. Реализуйте эту задачу – разработайте типы-записи и набор методов для чтения данных. При

### Задача 6. Добавление данных.

Перед вами стоит задача добавления в БД информации о “новых” олимпийских играх, прошедших когда-то давно в 2008 году в Пекине. Это ж в целых пять таблиц надо добавлять сведения, осознаете вы! Более того, ситуация усугубляется тем, что данных у вас нет, ибо приятель, обещавший их раздобыть, конечно же, был занят доделками последнего пира и не выполнил обещания (и придется выкручиваться и сгенерировать данные любым способом, который вы сочтете подходящим, авось, никто второпях и не заметит подлога (например, можно взять за основу ОИ-2004 и “подправить” результаты и список участников). Конечно же, все это надо делать из кода на C#.

## Дополнительные задачи.

### Задача доп1. Вспоминаем про FillDB().

Протестируйте метод `FillDB()`, использующийся для первоначального создания БД. Обратите внимание на:

- 1) необходимость обработки исключений (файлы + БД);
- 2) многократное создание `SqlCommand()` в цикле;
- 3) использование жестко прописанной в коде строки подключения к БД;

Все это надо исправить!

### Задача доп2. Вспоминаем про типы, допускающие значение null.

Выполняя аудит написанного кода, вы осознаете, что, решая задачу 5 с записями, была упущена важная деталь – не все значения атрибутов у записей в БД могут быть определены! Возможны значения NULL. Вам необходимо внести соответствующие изменения в код, что решения этой проблемы (учесть, что из БД можно прочесть NULL).