

УДК 004.04

¹Ника Субботина, ¹Сергей Голиков, ²Сергей Чёрный

¹Херсонский национальный технический университет, Украина

²Керченский государственный морской технологический университет, Украина

АКТУАЛЬНЫЕ ПРОБЛЕМЫ ОПТИМИЗАЦИИ И НОРМАЛИЗАЦИИ ПРОГРАММНОГО КОДА В ПРИЛОЖЕНИЯХ

Nicka Subbotina, Sergei Golikov, Sergei Chornuy

THE OPTIMIZATION AND NORMALIZATION OF THE CODE IN THE APPLICATIONS

Неудержимый процесс создания и оптимизации программного обеспечения в период глобализации информационного общества является обыденным явлением жизненного цикла. Создание программ или комплексов (модулей, групп и т.д.) зачастую процесс очень сложный и требующий значительного концептуального анализа, но временные рамки не всегда этому способствуют. Это не свидетельствует тому что конечный продукт является «некачественным» или не работающим правильно, он может быть просто не достаточно оптимизирован и легок в языковом (программном) коде. Все это приводит и к оптимизации проекта (продукта), к быстройдействию, к наглядности и прочее. Оптимизировать программный код – не значит его просто уменьшить, сделать множество процедур или фрагментарных замен, а оптимизация не значит что необходимо подключить множество библиотек и думать, что функциональность улучшилась (ускорение алгоритмов вычислений, сокращение расходуемой памяти). «Совершенству нет придела» известное высказывание, так и в данном процессе, оптимизируя и нормализуя программный код необходимо помнить о работоспособности программы и методах своевременной сдачи проекта. Все это ведет к своевременному рефакторингу кода. Цель рефакторинга прежде всего структуризация и упрощение кода, приведение его в надлежащий вид, что бы другим специалистам, которые будут с ним работать после разработчика, не возникла необходимость полного изменения кода. Авторы книг М.Фаулер, Д.Кериевски, С.Макконнелла и др. процессы рефактоинга применять необходимо там где они действительно нужны, а не для простой переработки программного кода. В рамках данного аспекта можно рассмотреть MSSql Server CE и применение *.dll. что является достаточно актуальным процессом при оптимизации многих приложений. На рис.1. представлена структуризация. Необходимо структурировать знания, в какой *.dll что находится (XX в конце имени dll-файла обозначают версию SQL Server CE.



Рисунок 1 – Структура SQL Server CE

Например, для SQL Server CE 3.5 XX надо заменить на 35, для SQL Server CE 4.0 соответственно на 40): Storage Engine - sqlceseXX.dll. Query Processor - sqlceqpXX.dll. OLEDB Provider - sqlceoledbXX.dll. DB Utilities - sqlcecompactXX.dll. Managed Extensions - sqlcemeXX.dll. Merge Replication, Remote Data Access - sqlcecaXX.dll. Sync Services: Client Sync Provider - Microsoft.Synchronization.Data.SqlServerCe.dll. Sync Common - Microsoft.Synchronization.Data.dll. Server Sync Provider - Microsoft.Synchronization.Data.Server.dll.

ADO.NET Provider (ADO.NET v2 Provider) - System.Data.SqlServerCe.dll. LINQ/Entities (ADO.NET v3 Provider), Entity Framework - System.Data.SqlServerCe.Entity.dll. LINQ/SQL не требуется отдельной dll для подключения Error Messages - sqlceer35en.dll (для локализации необходимо изменить EN в названии файла).