

МАСШТАБИРОВАНИЕ ЭКРАННОГО ИЗОБРАЖЕНИЯ В ОС WINDOWS.

Нестеров А.И., 3 курс,

Севостьян Д.М., м.т.н., ассистент,

УО «Белорусский государственный технологический университет»

Приложение NuGenPresenter разработано на C# 2.0 для платформы Microsoft .NET Framework 2.0 с использованием WinAPI. Основное назначение приложения: масштабирование рабочего стола с возможностью последующего комментирования при помощи маркера. В качестве дополнительной возможности предусмотрена функция экспорта текущего изображения в графические файлы различных форматов. Пользователю также доступна функция установки выходного разрешения и преобразования изображения из цветного в черно-белое либо монохромное.

Приложение реализовано в Microsoft Visual Studio 2005 с использованием таких вспомогательных инструментов, как TortoiseSVN (Windows-клиент системы контроля версий Subversion), NUnit и DotNetMock (для блочного тестирования), а также собственной разработки автора - плагина (add-in в терминах Microsoft) для VS IDE – NuGenStudioTask.

За основу принята методология Кента Бека TDD (Test-Driven Development), суть которой заключается в том, что на первом этапе разрабатывается тестирующий код, т.е. код, который предполагает наличие определенной функциональности. И лишь после этого разрабатываются собственно классы. DotNetMock является компонентом пакета Framework, который предназначен для создания «заглушек», когда часть кода еще не написана, но наличие которой необходимо для тестирования. На основе этого компонента автором была разработана технология тестирования событийной составляющей классов. Суть технологии состоит в следующем: создается mock-объект (наследник класса MockObject). В конструктор передается ссылка на класс, чьи события необходимо протестировать. Также в mock-объекте создаются специальные свойства, которые могут отвечать за ожидаемое количество вызовов события, правильность переданных параметров, а также порядок вызовов событий.

Для решения проблемы тестирования визуальных элементов управления в рамках методологии TDD автором также используется технология собственной разработки. Суть этой технологии в том, что движок контроля представляет собой state-machine (конечный автомат), который имеет несколько методов для изменения своего состояния (MouseDown, MouseEnter, MouseLeave, MouseUp и пр.) Посредством вызовов этих методов имитируются действия пользователя, а текущее состояние автомата проверяется на предмет соответствия ожидаемому состоянию. Чтобы оптимизировать процесс тестирования предусмотрен плагин авторской разработки NuGenStudioTask для ведения всех записей. По сути плагин представляет собой расширенную версию TaskList из VS IDE. Из дополнительных возможностей: объединение задач в каталоги, пять уровней для задач, а также сохранение всех записей в XML-файле в одной директории с *.sln файлом проектов.

В процессе разработки был использован Inversion of Control (или Dependency Injection). Обычно некоторый класс представляет собой законченный функциональный блок, работу которого можно незначительно корректировать при помощи свойств и/или методов. В то же время Inversion of Control позволяет создать открытую архитектуру и использовать один и тот же код в различных целях. Так, например, один и тот же класс логгера (сохранение статистики) может вести лог как в текстовый файл, так и в консоль или базу данных. Необходимо только «инжектировать» необходимые сервисы в логгер.

Различают три вида инжектирования. Наиболее простые – это constructor-injection и setter-injection (т.е. инжектирование происходит через установку свойства). Service locator более сложен и обычно комбинируется с первым или вторым. Имеется контекст, который представляет сервисы на основе типа интерфейса, который этот сервис реализует. Таким образом, поведение класса может динамически изменяться в зависимости от контекста (при этом может использоваться даже один и тот же экземпляр класса). Автором был применен service locator вместе с constructor-injection.

Выводы: В процессе разработки приложения NuGenPresenter были использованы самые современные подходы к объектно-ориентированному проектированию и разработке программного обеспечения. Данный проект позволил убедиться, что идеи, родившиеся еще во времена Smalltalk (разделение данных представления), жизнеспособны по сей день и, кроме того, получают дополнительное развитие. Так было замечено, что частичный отказ от наследования позволяет получить более гибкую архитектуру приложения. Именно это качество наиболее ценно в условиях постоянных изменений требований к проекту.

Проект имеет практическую направленность и используется в ходе учебных занятий. Кроме того, данное приложение может использоваться в ходе конференций и семинаров для акцентирования внимания слушателей на наиболее важных моментах презентации докладчика.