

accuracy (**точність**) — дотримання норм, пов'язаних з точним виконанням інструкцій для експлуатації систем і обробці інформації, чесним і соціально-відповідальним відношенням до своїх обов'язків;

property (**приватна власність**) — недоторканність приватної власності — основа майнового порядку в економіці. Проходження цьому принципу означає дотримання права власності на інформацію й норм авторського права;

accessibility (**доступність**) — право громадян на інформацію, її доступність у будь-який час і в будь-якому місці.

Дослідження показали, що практично всі існуючі кодекси щодо етичних норм інженерів ПЗ базуються на загальнолюдських цінностях. Вони не регламентують виконання окремих дій, виконуваних в тій чи іншій ситуації, а створюють основу для прийняття індивідуальних моральних рішень.

Список літератури: 1. Наша культура // [Електрон.ресурс].-Спосіб доступу: <http://www.google.com/intl/uk/about/company/facts/culture/>. 2. Федченко А. Жизнь предоставляет всем людям шансы. Кто-то воспользуется ими, кто-то – нет/ А. Федченко // [Електрон.ресурс].-Спосіб доступу: http://incom.ua/index.php?option=com_content&task=view&id=375301&Itemid=32. 3. Абдуліна І. Роблячи ставки на розробку програмного забезпечення [Текст] // І. Абдуліна, В. Березанська // Інтелектуальна власність.–2012.–№9.–С8-12. 4. Сидорова Н. М. Навчання інженерії програмного забезпечення – систематичний огляд літератури/ Н. М. Сидорова // Матеріали міжнародної науково-практичної конференції аспірантів і студентів «Інженерія програмного забезпечення 2011». [Електрон.ресурс].-Спосіб доступу: [http://www.nbuu.gov.ua/portal/natural/Ipz/2011_2/Sidorova.pdf]. 5. Морозова Т. Ю. Про необхідність вивчення комп'ютерної етики майбутніми ІТ-фахівцями / Т. Ю. Морозова // [Електрон.ресурс].-Спосіб доступу: <http://www.nbuu.gov.ua/portal/natural/vkpi/FPP/2006-2/05Morozova.pdf>.

Надійшла до редколегії 20.03.2013

УДК 378.214.46:

Проблеми формування етичного кодексу інженерів програмного забезпечення/ В. О. Ліщина // Вісник НТУ «ХПІ». Серія: Нові рішення в сучасних технологіях. – Х: НТУ «ХПІ», – 2013. - № 1 (977). – С. 90-95. – Бібліогр.: 5 назв.

В статтю затронута проблема необхідності формування кодексу комп'ютерної етики, його составних елементів. Исследовано процесс формирования кодексов профессиональной деятельности стиспециалистов програмного обеспечения, включа зарубежный опыт. Проанализированы основне принципы формирования этических норм деятельности в сфере информационных технологий.

Ключевые слова: специалист ИТ-отдела, программное обеспечение, кодекс этики, ценности, корпоративная культура, этические нормы, профессиональная деятельность.

The paper initiated the problem need to develop a computer code of ethics and its components. The process of formation of professional codes of professional software such as foreign experience. The basic principles of forming ethical standards activities in the field of information technology.

Keywords: expert IT department, software, codeo fethics, values, corporate culture, ethics, professional activity.

УДК 004.051

И. Н. ЛОЗОВСКИЙ, студент, ХНУРЭ, Харьков

СРАВНЕНИЕ ПОДХОДОВ CODE FIRST И DATABASE FIRST В ПОСТРОЕНИИ ХРАНИЛИЩА ДАННЫХ ВЕБ-ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ ASP.NET.

Данная работа посвящена современным тенденциям в разработке хранилищ данных для приложений на технологии ASP.Net с использованием языка программирования С#. В этой статье я представляю основные преимущества и недостатки использования наиболее гибких, и популярных подходов Code First и Database First к построению хранилищ данных веб-приложений с использованием ORM технологии от компании Microsoft - ADO.Net Entity Framework.

© И. Н. ЛОЗОВСКИЙ, 2013

Ключевые слова: ASP.Net Code First; ASP.Net Database First; ADO.Net Entity Framework; технология ASP.Net; ORM технология.

Введение. В рамках данной статьи будут рассмотрены два наиболее популярных и гибких подхода: Code First и Database First, их особенности, и проблемы, которые возникают при разработке приложений, использующих данные подходы. Представленный в статье материал представляет собой сравнительную характеристику и результаты практического применения двух данных подходов, и как результат, представляет собой итог о целесообразности и эффективности использования данных подходов в разработке реальных веб-приложений.

Целью работы. Целью данной работы является сравнительный анализ и оценка достоинств и недостатков применения подходов Code First и Database First при построении моделей данных хранилищ приложений, разрабатываемых на технологии ASP.Net.

Анализ предметной области. Современная разработка веб-приложений не может обойтись без использования хранилищ данных, наиболее популярным вариантом которого является база данных. Разработанное ORM (object-relational mapping) решение ADO.Net Entity Framework от компании Microsoft предоставляет два наиболее гибких подхода к созданию хранилища данных приложения: Code First и Database First. Данные подходы создают “виртуальную объектную базу данных”, которая представлена в виде классов с набором полей, отражающих структуру базы данных, что отлично вписывается в рамки парадигмы объектно-ориентированного программирования [1].

Code First. Code First – подход к построению хранилища данных приложения, поддержка которого осуществляется с помощью технологии Entity Framework (далее EF) начиная с версии 4.0 [2-3]. Основной идеей данного подхода является написание кода (модели данных или сущностей, которые представляют собой объекты-классы, логически взаимосвязанные между собой) и генерация базы данных на основе этой модели с помощью технологии EF. В данном случае программист описывает класс, который будет представлять таблицу базы данных и поля, которые будут представлять собой поля в таблице базы данных.

Модель данных очень легко изменить, вследствие чего изменится и сама база данных. Эти изменения EF отслеживает с помощью своей служебной таблицы – «EdmMetaData», которая создается во время генерации базы данных из модели [3]. Она хранит хеш-код модели данных приложения. Сравнивая этот хеш-код, EF знает, произошли ли изменения в модели или нет, и если да, то её надо обновить. Стивен Сандерсон в своей книге – «ASP.NET MVC 3 Framework с примерами на C# для профессионалов» многократно использует в своих примерах данный подход и небезосновательно.

Применение данного подхода весьма привлекательно для программиста в связи с тем, что для того чтобы обновить структуру базы данных, необходимо обновить модель путем добавления или удаления полей, названия которых, в последствии, станут полями таблиц базы данных [2-4]. Применение данного подхода влечет и ряд проблем таких, как потеря данных после обновления модели, что в бизнес системах попросту не допустимо!

Существует подход, который частично решает данную проблему – «Code First Migrations», рассмотрение которого выходит за рамки данной статьи, однако и он не лишен своих не приятных нюансов, заставляя программиста писать большое количество кода, чтобы внести изменения в модель. Данная проблема, например, решается путем создания резервной копии базы данных, из которой впоследствии можно извлечь уже существующую информацию, однако это представляет собой определенные проблемы, так как для синхронизации баз данных придется выделять время на написание

отдельных модулей, которые будут этим заниматься, что не многие заказчики смогут одобрить. Результат применения данного подхода показан на рис.

Как можно увидеть на рис. – справа – код модели, а слева – сама таблица базы данных, отражающая данную модель. Каждому полю модели соответствует поле в таблице базы данных.

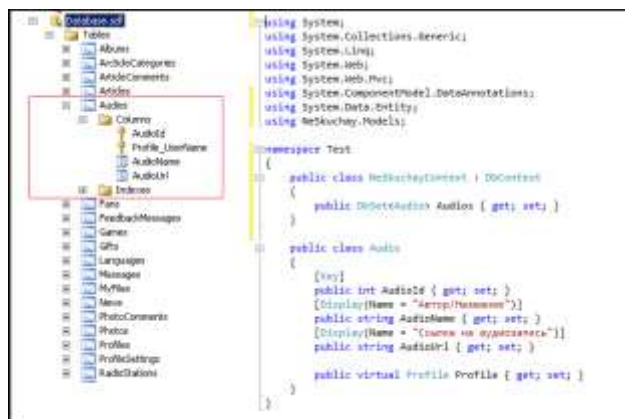


Рис. - Результат применения подхода Code First в приложении ASP.Net

Database First. Database First – подход, основной идеей которого является в первую очередь создание базы данных, а затем, на её основе, сгенерировать модель данных приложения с помощью EF. Для получения контекста базы данных нужно добавить в приложение файл – «ADO.Net Entity Data Model». Контекст базы данных это своеобразный слой доступа к данным, обладающий CRUD (Create, Read, Update, Delete) возможностями [4]. Данный файл представляет собой набор взаимосвязанных прокси-классов, которые отражают модель базы данных. Данный подход возлагает проблему создания модели на EF, а также позволяет программно вызывать хранимые процедуры из сгенерированного кода, что существенно экономит время. Данная возможность отсутствует в подходе Code First, и заставляет разработчиков вручную добавлять дополнительный код, который будет работать с хранимыми процедурами [4-5]. Данный подход является противовесом Code First, однако модификация структуры модели данных при таком подходе существенно затруднена из-за необходимости обновления всей структуры данных приложения целиком. Также, код, сгенерированный EF, имеет много сторонних объектов, которые ухудшают скорость доступа к данным.

Выводы. Были рассмотрены два наиболее гибких подхода, поддерживаемых Entity Framework, по созданию хранилища данных приложений, основанных на веб-платформе ASP.Net. Исходя из достоинств и недостатков данных подходов, можно выбрать наиболее подходящую модель для разрабатываемого приложения.

Список литературы: 1. Сандерсон, С. ASP.NET MVC 3 Framework с примерами на C# для профессионалов, 3-е изд. [Текст] : пер. с англ. - М.: ООО “И.Д. Вильямс”, 2012.- С. 221-240. 2. Макдональд, М. Microsoft ASP.Net 4.0 с примерами на C# 2010 для профессионалов, 4-е изд. [Текст]: пер. с англ. – М.: ООО “И.Д. Вильямс”, 2011.- С. 840-910. 3. Code-First Development with Entity Framework 4 [Электронный ресурс] / Official ASP.Net web-blogs.— Режим доступа: \www/ URL: <http://weblogs.asp.net/scottgu/archive/2010/07/16/code-first-development-with-entity-framework-4.aspx> — 18.03.2013 г. — Загл. с экрана. 4. ASP.Net MVC Application Using Entity Framework Code First [Электронный ресурс] / MSDN.— Режим доступа: \www/ URL: <http://code.msdn.microsoft.com/ASPNET-MVC-Application-b01a9fe8> — 18.03.2013 г. — Загл. с экрана. 5. ASP.Net Web Forms Application Using Entity Framework 4.0 Database First [Электронный ресурс] / MSDN.— Режим доступа: \www/ URL: <http://code.msdn.microsoft.com/ASPNET-Web-Forms-97f8ee9a> — 18.03.2013 г. — Загл. с экрана.

Надійшла до редколегії 20.03.2013

УДК 004.051

Сравнение подходов Code First и Database First в построении хранилища данных веб-приложений на платформе ASP.Net/ Лозовский И. Н. // Вісник НТУ «ХП». Серія: Нові рішення в сучасних технологіях. – Х: НТУ «ХП», – 2013. - № 1 (977). – С. 95-98. – Бібліогр.:5 азв.

Дана робота присвячена сучасним тенденціям в розробці сховищ даних для додатків на технології ASP.Net з використанням мови програмування C#. У цій статті я представляю основні переваги та недоліки використання найбільш гнучких, і популярних підходів Code First і Database First

до побудови сховищ даних веб-додатків з використанням ORM технології від компанії Microsoft - ADO.Net Entity Framework.

Ключові слова: ASP.Net Code First; ASP.Net Database First; ADO.Net Entity Framework; технологія ASP.Net; ORM технологія.

This work is devoted to the modern trends in the development of storage technologies for applications on ASP.Net using C# programming language. In this article I present the main advantages and disadvantages of the most versatile and popular approaches Code First and Database First to data warehousing in Web-applications using ORM technology provided by Microsoft - ADO.Net Entity Framework.

Keywords: ASP.Net Code First; ASP.Net Database First; ADO.Net Entity Framework; ASP.Net technology; ORM technology.

УДК 518.9+681.51.011

М. Б. МУНИБ, соискатель, Таврический национальный университет им.В.И.Вернадского, Симферополь

МНОГОКРИТЕРИАЛЬНАЯ ОПТИМИЗАЦИЯ В ЗАДАЧАХ УПРАВЛЕНИЯ

Задачи управления решаются на основе многокритериальной оптимизации. Разработана последовательность действий формализующих решение задачи управления. Предлагается три основных метода применения многокритериальной оптимизации основанных на искусственном слиянии нескольких выбранных показателей в один. Это метод минимизации и максимизации, «метод последовательных уступок» и метод предложенный И. Никовським.

Ключевые слова: многокритериальная оптимизация, задачи управления, метод последовательных уступок, слияние показателей, система, критерии оптимизации, математические модели.

Введение. Обращаясь к классической постановке задачи математического программирования, необходимо, в первую очередь, отметить, что такая задача предполагает только одну целевую функцию, которая количественно определена. Рассматривая же реальные системы важно понимать, что на роль критерия оптимальности претендуют несколько десятков показателей. Так же, желательным является применение нескольких критериев синхронно, причем они могут быть вообще несовместимы, в частности, требование достичь максимальной эффективности при минимальных затратах с точки зрения математической постановки задачи является некорректной. Минимальные затраты – это нулевые затраты, имеющие место при полном отсутствии каких-либо процессов. Подобно максимальной эффективности может быть достигнута только в случае использования определенных объемов ресурсов. Поэтому корректными является постановки задач такого типа: достичь максимальной эффективности при заданных затратах или достичь заданного эффекта при минимальных затратах.

Так как не существует единого универсального критерия эффективности, то довольно часто прибегают к рассмотрению многокритериальной оптимизации. Хотя задача управления предусматривает одну целевую функцию, разработаны математические методы, позволяющие строить компромиссные планы, то есть осуществлять многокритериальную оптимизацию. Однако, на сегодняшний день, в научной литературе мало внимания уделено изучению многокритериальной оптимизации в задачах управления, поэтому данная работа будет посвящена этому вопросу.

Эффективное решения задач управления стало допустимым благодаря развитию средств вычислительной техники, информационных технологий, методов системного анализа, математического моделирования и теории оптимизации. Решением этих проблем, в свое время занимались такие ученые как: В. Михайлович, В. Волкович [1], И. В. Сергиенко [2], А. Волошин, В. Заславский, И. Ушаков [3], В. Танаев [4], В. Горелик [5], С. И. Наконечный, С.С. Савина [6], Ю. Н. Кузнецов, В. И. Кузубов, А. Б. Волощенко [7],