

Литература

.NET/C#

Лекция 1

Clean Code

Обзор .NET/C#, типы данных, классы vs структуры

Лекция 2

Циклы, коллекции, Generic типы, Generic коллекции

Лекция 3

Основы ООП, применение ООП

Лекция 4

Сериализация, рефлексия, Input/Output

Лекция 5

Делегаты, события, многопоточность

Задание

Лекция 6

Linq, лямбда выражения

Лекция 7

Advanced Topics, ADO.Net - part 1

Итоговое задание по C#

ASP.NET

Лекция 1

Обзор ASP.NET, IIS, Запросы, Page Life Cycle

Лекция 2

Создание веб сайтов (IIS). Global.asax/Session/Application/Web.config

Лекция 3

Core ASP.NET server controls, ASP.NET page validation, Control events. View State of Controls. Master Pages

Лекция 4

HTTPModule, HTTPHandler

Лекция 5

Сложные контролы для отображения данных, ADO.Net - part 2

Лекция 6

Создание своих контролов, User Control, Custom Control

Лекция 7

JavaScript

Лекция 8

AJAX/UpdatePanel/WebServices

Лекция 9

Аутентификация и Авторизация

Литература

[The Robert C. Martin Clean Code Collection](#)

[Code Complete: A Practical Handbook of Software Construction, Second Edition](#)

[CLR via C# \(4th Edition, Jeffrey Richter\)](#)

[C# 5.0 in a Nutshell \(Joseph Albahari\)](#)

[C# 3.0 Cookbook](#)

[Pro C# 5.0 and the .NET 4.5 Framework](#)

[C# in Depth \(3rd Edition, Jon Skeet\)](#)

[Pro ASP.NET 4.5 in C#](#)

[Git \(Bitbucket\) manual](#)

.NET/C#

Лекция 1

Clean Code

<http://habrahabr.ru/post/159973/>

<http://habrahabr.ru/post/206868/>

В том же духе можно много почерпнуть на хабре по тегу **Совершенный Код***

Обратить внимание: культура написания кода. Краткость и лаконичность, короткие методы, осознанное именование, повторное использование. Для теста можно попробовать почитать свой же код через 1-3 дня и понять, что этот код выполняет и как.

Для более глубокого и полного изучения (изучать позже):

[The Robert C. Martin Clean Code Collection](#)

[Code Complete: A Practical Handbook of Software Construction, Second Edition](#)

.Net Platform, C#

Обзор .NET/C#, типы данных, классы vs структуры

CLR via C# (4th Edition, Jeffrey Richter):

CHAPTER 1 The CLR's Execution Model

CHAPTER 4 Type Fundamentals

CHAPTER 5 Primitive, Reference, and Value Types

CHAPTER 21 The Managed Heap and Garbage Collection (читать до Forcing Garbage Collections)

<http://habrahabr.ru/sandbox/68552/>

C# 5.0 in a Nutshell (Joseph Albahari):

CHAPTER 1: Introducing C# and the .NET Framework

CHAPTER 2: C# Language Basics

CHAPTER 3: Creating Types in C#

Обратить внимание: Модель выполнения .NET платформы, управление памятью, базовая иерархия наследования типов, value types/reference types (их суть и разница).

Для более глубокого и полного изучения (изучать позже):

.NET Platform: [CLR via C# \(4th Edition, Jeffrey Richter\)](#)

Лекция 2

Циклы, коллекции, Generic типы, Generic коллекции

1. Основные моменты:

- a. Циклы:
 - i. Виды циклов
 - ii. Ключевые слова break, continue, yield
- b. Коллекции:
 - i. В каких ситуациях следует использовать основные типы коллекций: ArrayList, SortedList, Queue, Stack, Hashtable, BitArray, StringCollection, StringDictionary, ListDictionary, HybridDictionary, NameValueCollection
 - ii. Интерфейс IEnumerable
- c. Generic типы: принцип работы, преимущества, ограничения ("constraints")
- d. Generic коллекции: List<>, Queue<>, Stack<>, Dictionary<>, SortedList<>, SortedDictionary<>, LinkedList<>

2. Источники:

- a. [C# 3.0 Cookbook](#) : chapter 4 “Generics”, chapter 5 “Collections”, chapter 6 “Iterators, Partial Types, and Partial Methods”
- b. [Pro C# 5.0 and the .NET 4.5 Framework](#) - chapter 3, chapter 9

3. Практические задания:

- a. Вывести на консоль таблицу умножения (отформатированный квадрат 10x10) циклами for, do..while, while, foreach. Диагональ с квадратами чисел выделить цветом.
- b. Реализовать generic коллекцию, которая внутри себя хранит элементы в List<>, если количество элементов <=5, и переключается на SortedList<> (элементы копируются в другую внутреннюю коллекцию), как только количество элементов доходит до 6. Оценить скорость работы этой коллекции в сравнении с List<> и SortedList<>
- c. Оценить скорость а) добавления элементов б) получения элементов в) сортировки элементов в коллекциях типа List<> и ArrayList. Использовать элементы типа System.Int32 и System.String.
- d. Реализовать Generic класс: а) ограниченный "non-nullable value type" параметрами; б) содержит в себе public property указанного при

**создании типа; с) коллекция экземпляров созданного класса
сортируется по таким же правилам, как и тип параметризованного
public property.**

Лекция 3

Основы ООП, применение ООП

1. Основные моменты:
 - a. Объектно-ориентированный подход:
 - b. Базовые понятия ООП:
 - i. Инкапсуляция
 - ii. Наследование
 - iii. Полиморфизм
 - iv. Абстракция
 - c. Применение ООП:
 - i. Интерфейсы
 - ii. Абстрактные классы
 - iii. Explicit Interface Implementations
2. Источники:
 - a. [Object-Oriented Analysis and Design with Applications, 3rd Edition](#)
Chapter 1-4
 - b. [ООП Википедия](#)
 - c. [Интерфейсы VS Класы](#)
3. Практические задания:

Создать иерархию классов и выделить основные свойства

Живые Существа			
Животные		Рыбы	
Лошадь	Собака	Карась	Плотва

- a. Класс «Живые Существа» должен быть абстрактным и содержать только свойство «Идентификатор Существа».

- b. Реализовать метод «Дышать под водой» у класса «Рыбы» и метод «Есть сено» у класса «Лошадь».**
- c. Сделать так чтобы собака была обязана реализовать метод «Дышать под водой», а карась «Есть сено»**
- d. Создать коллекцию из объектов типа «Живые существа».**
- e. Реализовать метод который будет считать количество ног у всех элементов этой коллекции.**
- f. Реализовать метод который найдет и выведет идентификаторы существ которые могут дышать под водой**

Лекция 4

Сериализация, рефлексия, Input/Output

Pro C# 5.0 and the .NET 4.5 Framework:

CHAPTER 18

CHAPTER 20

C# 5.0 in a Nutshell (Joseph Albahari):

CHAPTER 15

CHAPTER 17

CHAPTER 19

Обратить внимание:

1. Сериализация
 - a. Концепция сериализации
 - b. Форматеры. Использование форматеров
 - c. Сериализация коллекций
2. Рефлексия
 - a. Концепция рефлексии
 - b. Рефлексия типов и методов
 - c. Динамически загружаемые сборки
 - d. Позднее связывание
3. Input/Output
 - a. System.IO, работа с файлами и папками
 - b. Класс Stream, классы Writers и Readers

Задание:

Реализовать класс Employee, который будет иметь открытые свойства: Last Name, First Name, Age, Department, Address и закрытое поле EmployeeID. Класс должен сериализоваться в XML формат. Рутовая нода при сериализации/десериализации Employees должна называться Employees (а не ArrayOfEmployee например)

Написать Console Application, которое должно выполнить следующую последовательность действий:

1. **Десериализовать из заранее подготовленного файла коллекцию объектов Employee, при этом поле EmployeeID должно содержать в себе значение Last Name + First Name.**
2. **Вывести на экран информацию о каждом Employee в таком виде: № п/п, Last Name: value, First Name: value, и т.д. Значения EmployeeID и Address взять из закрытых полей используя рефлексия.**
3. **Отобразить Employee в возрасте от 25 до 35 лет и упорядочить их по EmployeeID. Полученную коллекцию сериализовать в новый файл.**

Лекция 5

Делегаты, события, многопоточность

1. Делегаты:

- Понятие и определение типа данных Делегат в .NET [1, стр. 357 - 360, 362 - 367]
- System.Delegate vs. System.MulticastDelegate [1, 360 - 361]
- Цепочки делегатов [1, 367 - 370]
- Обобщенные делегаты [1, 370 - 371]
- Action<>, Func<> [1, 371 - 373]

2. События:

- Паттерн Издатель/Подписчик [2]
- Понятие событий в C# [1, 373 - 379]
- event vs. delegate [3]
- Custom Event Args [1, 379 - 380]
- EventHandler<T> [1, 381]
- Анонимные методы в C# [1, 381 - 384]

3. Многопоточность:

- Процесс, домен приложения, контекст, поток [1, 652 - 655]
- Асинхронная природа делегатов в .NET [1, 655 - 657]
- IAsyncResult [1, 657 - 664]
- Пространство имен System.Threading [1, 664 - 675]
- Синхронизация потоков (lock, Monitor, Interlocked, SynchronizationAttribute) [1, 675 - 680]
- Timer [1, 680 - 681]
- Пул потоков CLR [1, 681 - 683]
- Доступ к элементам UI во вторичных потоках [1, 686 - 687]
- Advanced Threading [5, chapter 22]
- Parallel Programming [5, chapter 23]

Список литературы:

1. Andrew Troelsen - Pro C# 5.0 and the .NET 4.5 Framework - 2013 русское издание
2. [Издатель-подписчик \(шаблон проектирования\) Вики](#)
3. [C# events vs. delegates](#)

4. Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов - 2011
5. [C# 5.0 in a Nutshell \(Joseph Albahari\)](#)

Задание

Необходимо разработать многопоточное приложение для ОС Windows (WinForms), которое будет отлавливать ошибки времени выполнения, возникающие в дочерних потоках, с выводом информации об ошибках в текстовый блок.

Требования:

1. Приложение должно состоять из трех потоков, которые выполняются параллельно:
 - "основной поток" (`Application.Run()`);
 - два дочерних "потока, генерирующих исключения". Это должны быть два независимых потока, генерирующих ошибки времени выполнения различных типов (например `FileNotFoundException` и `OutOfMemoryException`). Каждый такой поток должен записывать сообщение о своей ошибке в общую очередь (`Queue`) и уведомлять основной поток о возникновении ошибки.
2. Дочерние потоки, генерирующие исключения, должны создаваться и запускаться при старте приложения, работать на протяжении всего времени жизни приложения и завершать работу при завершении работы приложения.
3. Процесс генерации и регистрации исключений должен запускаться и останавливаться по нажатию на соответствующие кнопки в окне приложения. При остановке генерации и отлова дочерние потоки должны продолжать выполняться!!!
4. Каждая запись об ошибке времени выполнения из дочернего потока должна содержать имя потока, из которого пришла и тип сгенерированного исключения.
5. Для уведомления основного потока о возникновении исключений необходимо использовать события.
6. Выводить сообщения об ошибках времени выполнения из очереди необходимо в элемент управления `TextBox` (мультилайн), который необходимо разместить на главном окне приложения.
7. Синхронизация между потоками должна осуществляться с помощью базовых механизмов, предоставляемых платформой .NET, таких как `Monitor`, типов, производных от `WaitHandle`, и т.д. Не рекомендуется использовать

**высокоуровневные механизмы, предоставляемые т.н. TPL такие, как
BlockingCollection, различные Concurrent* коллекции.**

- 8. Алгоритм генерации исключения и выбор типов генерируемых исключений
остается на усмотрение исполнителя.**

Лекция 6

Linq, лямбда выражения

[C# 5.0 in a Nutshell \(Joseph Albahari\):](#)

CHAPTER 4. Advanced C#->Lambda Expressions

CHAPTER 8. LINQ Queries (319)

CHAPTER 9. LINQ Operators (375)

CHAPTER 10. LINQ to XML (423-450)

Custom LINQ Providers

LINQ Expression Trees

[C# in Depth \(3rd Edition, Jon Skeet\):](#)

Appendix A, LINQ standard query operators

<https://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b>

<http://stackoverflow.com/questions/16322/learning-about-linq>

Для более глубокого и полного изучения(изучать позже):

[C# in Depth \(3rd Edition, Jon Skeet\)](#)

Задание:

Создать приложение "Менеджер Контактов". В качестве хранилища данных необходимо использовать XML файл. Как технологию доступа к данным использовать LINQ to XML.

1. Приложение должно содержать два проекта.
2. Первый проект(проект UI): Windows Forms проект, который обеспечивает отображение менеджера контактов. Задача этого проекта - только отобразить возможные функции, следовательно требования минимальны и НЕ ожидается создание красивого и "правильного" UI.
3. Второй проект(проект Core): DLL (Assembly). В данном проекте описана основная логика работы "Менеджера Контактов".
4. Проект UI только вызывает необходимые классы/методы из проекта Core.
5. Менеджер контактов позволяет добавлять, удалять, редактировать список контактов.
6. Контакт обладает следующими свойствами:
 - Имя
 - Фамилия
 - Группа
 - Домашний телефон
 - Мобильный телефон
 - Фото
7. Над списком контактов необходимо отобразить:
 - Переключатель для группировки списка контактов по полю "Группа". Если включен, то необходимо отображать контакты в группах (в виде дерева)
 - Поле для фильтрации списка контактов одновременно по полям Имя и Фамилия.

8. Примерный вид программы:

Имя

Фамилия

Фото

Добавить

Удалить

Сохранить

Лекция 7

Advanced Topics, ADO.Net - part 1

1. Advanced Topics:

1. Индексаторы, многомерные индексаторы, индексаторы в интерфейсах [1, 391 - 396]
2. Перегрузка операций [1, 396 - 402]
3. Специальные преобразования типов, explicit и implicit преобразования [1, 402 - 408]
4. Extension methods [1, 408 - 412]
5. Анонимные типы в C# [1, 412 - 417]

2. ADO.Net - part 1:

- a. Введение в ADO.NET [1, 745 - 747]
- b. Поставщики данных в ADO.NET [1, 747 - 750]

- c. Пространство имен System.Data [1, 750 - 755]
- d. Абстрагирование поставщиков данных с помощью интерфейсов [1, 755 - 756]
- e. Конфигурационные файлы приложения в контексте ADO.NET [1, 757 - 758]

Список литературы:

1. Andrew Troelsen - Pro C# 5.0 and the .NET 4.5 Framework - 2013 русское издание
2. [Издатель-подписчик \(шаблон проектирования\) Вики](#)
3. [C# events vs. delegates](#)
4. Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов - 2011

Итоговое задание по C#

Необходимо разработать приложение для ОС Windows, которое позволяет для указанной директории произвести рекурсивный перебор всех вложенных поддиректорий и файлов с занесением результатов в файл XML формата и визуализацией в виде дерева.

Требования:

1. По каждой поддиректории и файлу, в XML файле результатов должно быть указано имя, дата создания и размер (только для файлов; реализация размера для директорий будет засчитана как плюс).
2. Помимо размещения результатов в файл они также должны быть занесены в визуальный элемент управления - дерево, которое необходимо разместить на основном окне приложения.
3. Приложение должно быть организовано в виде четырех потоков, которые должны выполняться параллельно:
 - "основной" (Application.Run());
 - "поток сбора информации". Собирает информацию об объектах файловой системы и сохраняет ее в некую общедоступную коллекцию (shared state). Выполняет сканирование указанной директории и создаются только на время сканирования.
 - "поток занесения результатов в XML файл" (получает от "поток сбора информации" сведения об очередной поддиректории или файле из вышеуказанной коллекции и заносит эти сведения в XML файл асинхронно). Иерархия нод в XML файле должна отражать иерархию в файловой системе. Поток должен работать на протяжении всего времени жизни приложения.
 - "поток занесения результатов в дерево" (получает от "поток сбора информации" сведения об очередной поддиректории или файле из вышеуказанной коллекции и заносит эти сведения в элемент управления - дерево асинхронно). Иерархия нод в дереве должна отражать иерархию в файловой системе. Поток должен работать на протяжении всего времени жизни приложения.
4. Синхронизация между потоками должна осуществляться с помощью базовых механизмов, предоставляемых платформой .NET, таких как Monitor, типов, производных от WaitHandle, и т.д. Не рекомендуется использовать

высокоуровневные механизмы, предоставляемые т.н. TPL такие, как BlockingCollection, различные Concurrent* коллекции.

5. Выбор директории и указание размещения файла результата должно осуществляться посредством графического интерфейса пользователя.
6. Особое внимание необходимо уделить обработке возможных ошибок времени выполнения и оповещению о них пользователя. Для оповещения об ошибках времени выполнения необходимо использовать события.
7. Код для вывода сообщения об ошибках необходимо оформить в виде Extension-метода, доступного для всех типов, реализующих интерфейс ISynchronizeInvoke.

ASP.NET

Лекция 1

Обзор ASP.NET, IIS, Запросы, Page Life Cycle

1. Обзор ASP.NET
 - a. Введение в ASP.NET [1, Chapter1]
 - b. Разработка сайтов в Visual Studio [1, Chapter2]
2. Основы архитектуры IIS
 - a. Основы IIS [2]
 - b. ASP.NET and IIS [3, Chapter 2, Chapter 3]
 - c. IIS Pipeline [2],[3, page28 - page39]
3. Запросы
 - a. Типы http запросов [4, 5]
4. Page Life Cycle
 - a. Модель событий ASP.NET [1, page103 - page110] [\[Жизненный цикл страницы в ASP.NET 4.0 WebForms\]](#) [3, Chapter 5]

Список литературы:

1. [Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов](#)
2. [Основы архитектуры IIS, или запросопровод для ASP.NET](#)
3. [Dino Esposito - Programming with ASP.NET 4.0](#)
4. [Генерация http запросов](#)
5. [Как работает браузер: HTTP запрос](#)

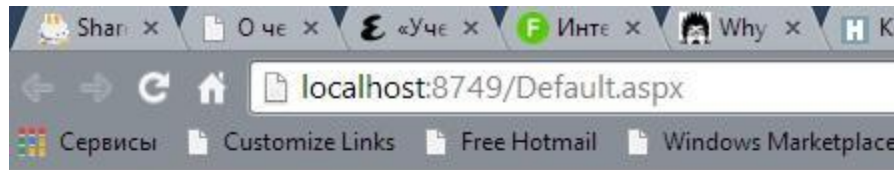
Задание:

Развернуть IIS на своей рабочей машине. Настроить его для работы с ASP.NET. Создать веб-сайт (описание сайта смотри ниже). Приготовить инсталляционный пакет для развертывания вашего сайта на сервере.[3, Chapter 2 (Deploying ASP.NET Applications)]

Описание сайта: Добавить одну кнопку на страницу Default.aspx. Кнопка должна отправлять постбек на сервер. При помощи свойства страницы Response (Page.Response...) вывести лог всех методов и событий страницы Default.aspx. [Смотри: [Жизненный цикл страницы в ASP.NET 4.0 WebForms](#)]. В комментариях к коду приложить подробное пояснение почему вызов некоторых методов невозможно отложить на страницу методом Page.Response.Write()

Также вывести на страницу удобным для чтения форматом (желательно в табличку) значения всех элементов объекта Page.Request.Form (хранит в себе NameValueCollection). Объяснить в комментариях к коду что это за коллекция. И почему у нее после нажатия на кнопку хранятся уже другие элементы.

В результате выполнения задания вы должны получить страницу как указано на рисунке. Тут указано лишь несколько событий для примера.



```
Page_PreInit 07.01.2015 16:10:09
Page_Init 07.01.2015 16:10:09
Page_PreLoad 07.01.2015 16:10:09
Page_Load 07.01.2015 16:10:09
Page_LoadComplete 07.01.2015 16:10:09
Page_PreRenderComplete 07.01.2015 16:10:09
```

Button For PostBack

Проверка задания: вам нужно будет передать инсталяционный пакет нам, чтобы мы без ручного копирования файлов смогли развернуть сайт у себя на сервере.

Лекция 2

Создание веб сайтов (IIS). Global.asax/Session/Application/Web.config

1. Создание веб сайтов (IIS) [1, Chapter 3]
2. Global.asax/Session/Application/Web.config[1, Chapter 5,6]

Задание:

Создать страницу для отображения статистики web-приложения. Она должна содержать в себе следующую информацию:

1. Количество запросов к web-приложению (за все время).
2. Количество запросов к web-приложению (за день).
3. Количество запросов к конкретной странице (за все время).
4. Количество уникальных посетителей (за все время).
5. Количество посетителей (за день).

Нельзя использовать IIS Counters. Реализация должна быть основана на основании материала из текущего модуля (Session, Application и так далее)

Список страниц, для которых необходимо отображать статистику, должен храниться в текстовом файле. Путь к файлу хранить в файле web.config. Для тестирования создать не менее 3 страниц.

Исходный код задания должен содержать пакет для развертывания сайта на IIS.

Список литературы:

1. [Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов - 2011](#)

Лекция 3

Core ASP.NET server controls, ASP.NET page validation, Control events. View State of Controls. Master Pages

1. Core ASP.NET server controls
 - a. ASP.NET controls in general [1, Chapter 4], [2, Chapter 6]
 - b. HTML Controls [1, Chapter 4, page 151(если русское издание)], [2, page 235 - 252]
 - c. Web Controls [1, Chapter 4, page 162(если русское издание)], [2, page 253 - 268]
2. ASP.NET page validation
 - a. Page validation [2, Chapter 9 page 379]
 - b. Page validation controls [1, page 180 - 195(если русское издание)], [2, Chapter 9 page 379 - 396]
3. Control events
 - a. Свойства, События и методы класса Control [2, Chapter 6, page 218 - 235], [[Жизненный цикл страницы в ASP.NET 4.0 WebForms](#)]
4. View State of Controls
 - a. View State of Controls [2, Chapter 6, page 227 - 228]
5. Master Page
 - a. Работа с мастер страницой [2, Chapter 8, page 320 - 336]

Список литературы:

1. [Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов](#)
2. [Dino Esposito - Programming with ASP.NET 4.0](#)

Задание:

Необходимо разработать форму регистрации пользователей для некоего портала. Форма должна позволять ввести данные о пользователе и провалидировать введенные пользователем данные. Форма должна быть реализована в виде страницы ASP.NET Web Forms. ВНИМАНИЕ!!! Все контролы на форме необходимо создавать динамически, так же

привязка данных к контролам должна осуществляться динамически. Использование Visual Studio Page Designer для создания контролов запрещено.

На форме должны присутствовать следующие поля (все поля на форме обязательны для заполнения):

1. Псевдоним пользователя.
2. Имя. Значение поля должно отличаться от "Псевдоним пользователя".
3. Фамилия. Значение поля должно отличаться от "Имя".
4. Дата рождения. Значение данного поля должно находиться в диапазоне дат с 1го декабря 1960го года по сегодняшний день (DateTime.Now). Поле необходимо реализовать с помощью TextBox контрола, в который попадают значения из Calendar контрола (для примера см. Рис. 1).
5. E-mail. Для валидации введенного значения необходимо использовать регулярное выражение.
6. Страна. Представляет собой выпадающий список без пустых значений.
7. Город. Представляет собой выпадающий список без пустых значений. Доступные для выбора города меняются в зависимости от выбранной страны (значения в выпадающем списке зависят от выбранного значения в поле Страна).
8. Итоги валидации. Выводит списком результаты валидации формы. Для реализации поля необходимо использовать ValidationSummary контрол.

Кроме того на форме должны быть расположены две кнопки: Submit и Reset. Первая делаетPostBack с валидацией, вторая очищает поля формы от введенных значений.

Исходный код задания должен содержать пакет для поднятия сайта со страницей под IIS.

Registration form:

Nick Name:	<input type="text" value="123"/>																																																																									
First Name:	<input type="text" value="123"/>	Should not be equal to Nick Name:																																																																								
Last Name:	<input type="text"/>	This field is required.																																																																								
Date of Birth:	<div><div>2/11/2015</div><table><tr><td>≤</td><td colspan="7">February 2015</td><td>≥</td></tr><tr><td>Sun</td><td>Mon</td><td>Tue</td><td>Wed</td><td>Thu</td><td>Fri</td><td>Sat</td><td></td><td></td></tr><tr><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td></td><td></td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td></td><td></td></tr><tr><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td></td><td></td></tr><tr><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td></td><td></td></tr><tr><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td></td><td></td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td></td><td></td></tr></table></div>	≤	February 2015							≥	Sun	Mon	Tue	Wed	Thu	Fri	Sat			25	26	27	28	29	30	31			1	2	3	4	5	6	7			8	9	10	11	12	13	14			15	16	17	18	19	20	21			22	23	24	25	26	27	28			1	2	3	4	5	6	7			Date of Birth should be between 12/1/1960 and 1/11/2015
≤	February 2015							≥																																																																		
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																																																				
25	26	27	28	29	30	31																																																																				
1	2	3	4	5	6	7																																																																				
8	9	10	11	12	13	14																																																																				
15	16	17	18	19	20	21																																																																				
22	23	24	25	26	27	28																																																																				
1	2	3	4	5	6	7																																																																				
E-mail:	<input type="text" value="1111"/>	E-mail provided is invalid																																																																								
Location Country:	<input type="text" value="USA"/>																																																																									
Location City:	<input type="text" value="Washington"/>																																																																									
Validation Summary:	<ul style="list-style-type: none">• Should not be equal to Nick Name:• This field is required.• Date of Birth should be between 12/1/1960 and 1/11/2015• E-mail provided is invalid																																																																									
<div><input type="button" value="Submit"/> <input type="button" value="Reset"/></div>																																																																										

Рисунок 1 - Один из возможных вариантов внешнего вида формы

Лекция 4

HTTPModule, HTTPHandler

Основные моменты:

1. Предназначение HTTP Module, HTTP Handler
2. Синхронные и асинхронные хэндлеры
3. Регистрация хэндлеров для обработки различных типов файлов

Список литературы:

1. Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов - 2011 (глава 5)
2. Dino Esposito - Programming with ASP.NET 4.0 (chapter 4)
3. [https://msdn.microsoft.com/en-us/library/bb398986\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/bb398986(v=vs.140).aspx)

Задание:

Разработать сайт, добавляющий copyright (watermark) к картинкам. Требуемая функциональность:

- Страница "Home": содержит ссылку на страницу Upload, дропдаун со списком картинок и кнопку. При нажатии на кнопку в новом окне открывается изображение. (Задание со звёздочкой: При нажатии на кнопку начинается загрузка затребованного изображения, страница остаётся рабочей)
- Страница "Upload": даёт возможность заливать картинки на сервер. После загрузки изображения на сервер показывает пользователю назначенный уникальный идентификатор.
- Изображения на сервере должны храниться в исходном виде, watermark добавляется только во время отправки картинки пользователю. Текст, добавляемый на картинку, хранится в web.config.
- Известно, что в будущем будет разработано iPad приложение, одной из функций которого будет "User inputs the known ID of image, clicks the button and image with watermark is shown in separate tab of iPad application".

Все страницы сайта, перечисленные выше, и страницы, которые будут добавлены в будущем разработчиками из другой команды, должны выводить текст "You are not authorized" и не выводить своего содержимого, если в запросе не передан параметр UserKey, хранящий MD5 представление строки "I_Am_Real_Admin".

Лекция 5

Сложные контролы для отображения данных, ADO.Net - part 2

1. ADO.Net - part 2:
 1. Автономный уровень ADO.Net [1, 795 - 796]
 2. DataSet [1, 797 - 799]
 3. DataColumn [1, 799 - 802]
 4. DataRow [1, 802 - 805]
 5. DataTable [1, 805 - 811]
2. Привязка данных:
 1. Привязка данных (Data Binding) основные понятия [2, 361 - 375]
 2. Жизненный цикл страницы с привязкой данных [2, 375 - 377]
 3. SqlDataSource [2, 377 - 392]
 4. ObjectDataSource [2, 392 - 402]
 5. Ограничения элементов управления источниками данных [2, 402 - 406]
3. Сложные контролы для отображения данных:
 - a. GridView основы [2, 407 - 412]
 - b. Форматирование в GridView [2, 412 - 419]
 - c. Выбор строк в GridView [2, 419 - 423]
 - d. Сортировка в GridView [2, 423 - 427]
 - e. Разбиение на страницы (paging) в GridView [2, 428 - 434]
 - f. Шаблоны GridView [2, 434 - 446]
 - g. ListView [2, 447 - 452]
 - h. DetailsView, FormView [2, 452 - 457]
 - i. Расширенные табличные элементы управления [2, 457 - 473]

Список литературы:

1. [Andrew Troelsen - Pro C# 5.0 and the .NET 4.5 Framework - 2013 русское издание](#)
2. [Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов - 2011](#)

Задание

Необходимо разработать ASP.NET Web Forms страницу, которая предоставляет функциональность работы с содержимым Файлов Ресурсов ([RESX](#)), позволяет просматривать, редактировать, добавлять и удалять содержимое таких файлов.

Общие требования:

1. Страница должна уметь отображать имена всех ресурсных файлов из указанной директории в виде выпадающего списка (DropDownList). Путь к рабочей директории явно указывается в web.config.
2. При выборе имени ресурсного файла в выпадающем списке, его содержимое должно отобразиться в виде таблицы, с разбиением на "страницы" (Paging). На каждой странице необходимо отобразить 20 записей (см. рис. 1). Для отображения содержимого необходимо использовать контрол GridView.
3. При первой загрузке страницы должно отобразиться содержимое первого файла в выпадающем списке. В случае отсутствия ресурсных файлов в указанной директории предусмотреть соответствующее сообщение.
4. Для привязки данных к GridView необходимо использовать ObjectDataSource в паре с DataRepository, который бы позволял производить нужные CRUD операции с записями выбранного ресурсного файла.
5. При нажатии на ссылку Edit в соответствующей строке таблицы, выбранная строка должна принять вид, позволяющий редактировать содержимое всех полей ресурсной записи (см. Рис.2). При нажатии на кнопку Update внесенные изменения сохраняются, при нажатии на кнопку Cancel - отменяются. После нажатия на любую из этих кнопок, таблица должна вернуться к состоянию отображения данных с учетом внесенных изменений, если они были сохранены (см. Рис.1)
6. При нажатии на кнопку Delete соответствующая запись должна удалиться как из файла, так и из отображения.
7. При нажатии на кнопку Add New Record в файл должна добавиться новая запись. При этом ключ (Key) является обязательным для заполнения полем. Если ключ пустой, необходимо вывести соответствующее сообщение (см. Рис.3)
8. Ключ записи является уникальным идентификатором записи.

Please select resource file to edit:
RADSLClient.resx

Currently selected file is:
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\Resources\RADSLClient.resx

Key	Value	Comment	
SUI.Sec.MyHighLevelNeedsDrafts	My Interaction Plans (Drafts)	SystemUI	Edit Delete
SUI.Sec.MyHighLevelNeedsWithApprovalStatus	My Interaction Plans (with Approval Status)	SystemUI	Edit Delete
CMN.CostCenterDescription	Cost Center Description	Common	Edit Delete
CPM.DispCostCenters.Header.DisplCostCentersTitle	View Cost Center	Content Pages	Edit Delete
CPM.DispDeptBU.Header.DisplayDeptBUTitle	View Cost Center Description	Content Pages	Edit Delete
CPM.EditCostCenters.Header.EditCostCentersTitle	Edit Cost Center	Content Pages	Edit Delete
CPM.EditDeptBU.Header.EditDeptBUTitle	Edit Cost Center Description	Content Pages	Edit Delete
CPM.NewCostCenters.Header.NewCostCentersTitle	New Cost Center	Content Pages	Edit Delete
CPM.NewDeptBU.Header.NewDeptBUTitle	New Cost Center Description	Content Pages	Edit Delete
CPM.SearchDeptBU.Header.SearchDeptBUTitle	Search for a Cost Center Descriptions	Content Pages	Edit Delete
Form.Dept.CostCenterDescrSection	Cost Center Description	Forms	Edit Delete
L.Cont.FF.Cont_Baxter_Agreement	Baxter contracting party to the agreement (Note: Must be associated with the cost center description above)	Field Names	Edit Delete
L.Cont.FF.Cont_DeptBU_DL	Cost Center Description	Field Names	Edit Delete
L.Cont.FF.Cont_Select_DeptBU	Select a Cost Center Description	Field Names	Edit Delete
L.CostCenters.FF.CostCenters_Description	Cost Center Description	Field Names	Edit Delete
L.CostCenters.FF.Title	Cost Center Code	Field Names	Edit Delete
L.Deliver.FF_DeptBU_DeptBU_DL	Cost Center	Field Names	Edit Delete
L.Dept.FF_DeptBU_DeptBU_ID	Cost Center Description	Field Names	Edit Delete
L.Dept.FF_DeptBU_DeptBU_ID	Cost Center Number	Field Names	Edit Delete
L.Employee.FF_Emp_Manager_DeptBU_DL	Manager Cost Center	Field Names	Edit Delete
L.Employee.FF_Emp_Primary_DeptBU_DL	Primary Cost Center Description	Field Names	Edit Delete
L.Employee.FF_Emp_Sec_DeptBU_DL	Secondary Cost Center Description(s)	Field Names	Edit Delete
L.Employee.FF_Emp_SelectManager_DeptBU	Select a Manager Cost Center ([All available] - if empty)	Field Names	Edit Delete
L.Employee.FF_Emp_Select_Primary_DeptBU	Select a Primary Cost Center Description	Field Names	Edit Delete
L.Employee.FF_Emp_Select_Sec_DeptBU	Select a Secondary Cost Center Description(s)	Field Names	Edit Delete
L.Engage.FF_Eng_CostCenter	Cost Center	Field Names	Edit Delete
L.Engage.FF_Eng_DeptBU	Cost Center Description	Field Names	Edit Delete
L.Engage.FF_Eng_Select_DeptBU	Select a Cost Center Description	Field Names	Edit Delete
L.GroupExp.FF_GE_DeptBU	Cost Center Description	Field Names	Edit Delete
L.GroupExp.FF_GE_DeptBU_DL	Cost Center Description	Field Names	Edit Delete

1
2
3
4
Add New Record

Рисунок 1 - Вариант внешнего вида страницы в режиме отображения контента

L.Cont.FF_Cont_Baxter_Agreement	Baxter contracting party to the agreement (Note: Must be associated with the cost center description above)	Field Names	Edit Delete
L.Cont.FF_Cont_DeptBU_DL	Cost Center Description	Field Names	Edit Delete
L.Cont.FF_Cont_Select_DeptBU	Select a Cost Center Description	Field Names	Edit Delete
L.CostCenters.F.F_CostI	Cost Center Description	Field Names	Update Cancel
L.CostCenters.F.Title	Cost Center Code	Field Names	Edit Delete
L.Deliver.FF_Del_CostCenter_DL	Cost Center	Field Names	Edit Delete
L.Dept.FF_OptBU_DebtBU	Cost Center Description	Field Names	Edit Delete

Рисунок 2 - Вариант внешнего вида страницы в режиме редактирования записи

L.Engage.FF_Eng_Select_DeptBU	Select a Cost Center Description	Field Names	Edit Delete
L.GroupExp.FF_GE_DeptBU	Cost Center Description	Field Names	Edit Delete
L.GroupExp.FF_GE_DeptBU_DL	Cost Center Description	Field Names	Edit Delete
<div>*Record key should not be empty</div>		<div>Add New Record</div>	
1	2	3	4

Рисунок 3 - Пример валидации при создании новой ресурсной записи

Лекция 6

Создание своих контролов, User Control, Custom Control

1. Способы создания собственных контролов в ASP.NET
2. User Control [1, Chapter 15]
3. Custom Control [1 Chapter 27], [2, Chapter 12]

Список литературы:

1. [Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов](#)
2. [Dino Esposito - Programming with ASP.NET 4.0](#)

Задание:

Создать свой Custom Control (dll) который будет отображать вопросы с возможными вариантами ответа на них. Ваш контрол должен принимать в виде настроек (public property): вопросы с вариантами ответа на эти вопросы. Правильный вариант ответа также должен быть обозначен. Правильный вариант ответа возможен только один.

Также добавить два свойства контрола это размер шрифта вопросов и размер шрифта ответов (в пикселях).

Контрол отображает все вопросы с возможными вариантами ответа на них в виде нумерованного списка. Контрол должен позволять пользователю дать ответы на каждый вопрос. В низу контрола отображается суммарная информация - общее количество вопросов и количество отвеченных вопросов.

Контрол содержит кнопку "Завершить тест".

После завершения теста необходимо скрыть список вопросов и показать результаты ответов на вопросы, указав, какие ответы правильные, а какие нет.

Также контрол имеет два пользовательских события. Одно наступает когда пользователь дает ответ на вопрос. Второе наступает когда тест окончен (Нажали кнопку завершить тест).

Вариант ответа который выбрал пользователь должен также выделяться по стилю отображения.

Контрол предоставляет удобный интерфейс доступа к вопросам (нужно иметь возможность обращаться из кода к вопросам и просматривать был ли уже дан ответ на

этот вопрос. И какой вариант ответа был выбран. Помимо этого также хранится правильный вариант ответа).

Примерный вид контрола смотри Рисунок 1

1). Первый вопрос?
<input type="radio"/> Ответ 1
<input checked="" type="radio"/> Ответ 2
<input type="radio"/> Ответ 3
2). Второй вопрос?
<input type="radio"/> Ответ 1
<input type="radio"/> Ответ 2
<input type="radio"/> Ответ 3
3). Третий вопрос?
<input type="radio"/> Ответ 1
<input type="radio"/> Ответ 2
<input type="radio"/> Ответ 3

Всего вопросов: 3 Из них отвечено 1

Завершить тест

Рисунок 1 - Custom Control

Лекция 7

JavaScript

Основные моменты:

1. Основные концепции JavaScript: функции, объекты и операции
2. Дополнительно: ООП в JavaScript

Список литературы:

1. [W3C Lectures JavaScript](#)
2. [Учебник по JavaScript](#)

Задание:

Создать страницу для регистрации пользователей. Страница должна быть разделена на 5 секций, между которыми можно перемещаться кнопками назад и вперед. Перемещение между секциями должно быть возможно только если вся валидация текущей секции прошла успешно. Задание должно быть написано на чистом JavaScript и HTML без использования дополнительных библиотек. Программа должна состоять из HTML страницы и js файла с кодом.

Первая секция «Данные Аутентификации»:

Контролы:

- i. Логин пользователя (текстовое поле)
- ii. Пароль (текстовое поле)
- iii. Подтверждения пароля (текстовое поле)

Валидация:

- i. Логин пользователя должен содержать минимум одну букву в верхнем регистре, должен заканчиваться на цифру, длина должна быть не меньше 5 символов, состоять из английских букв и цифр.
- ii. Подтверждения пароля и пароль должны совпадать
- iii. В пароле должны присутствовать символы трех категорий из числа следующих четырех:
 - прописные буквы английского алфавита от A до Z;
 - строчные буквы английского алфавита от a до z;
 - десятичные цифры (от 0 до 9);
 - неалфавитные символы (например, !, \$, #, %)

Вторая секция «Личная информация»:

Контролы:

- i. ФИО (текстовое поле)
- ii. Дата рождения (текстовое поле)
- iii. Возраст (лейбл)

Валидация:

- i. Поле ФИО может содержать только символы и состоять из 3 слов, разделенных пробелом
- ii. Дата рождения должна быть валидной датой в формате dd/mm/yyyy.
- iii. Если дата рождения валидна, посчитать текущий возраст пользователя и записать в лейбл.
- iv. Если возраст пользователя меньше 12 лет, вывести сообщение «Регистрация только с 12»

Третья секция «Адрес»:

Контролы:

- i. Адрес (текстовое поле)
- ii. Город (текстовое поле, значение по умолчанию Харьков)
- iii. Область (дроп даун, значение по умолчанию Харьковская)
- iv. Почтовый индекс (текстовое поле)

Валидация

- i. Почтовый индекс должен содержать только 5 цифр

Четвертая секция «Подписка на уведомления»:

Контролы:

- i. Email (текстовое поле)
- ii. Хотите ли вы получать дополнительные новости (чекбокс)
- iii. Список подписок (чекбокс лист, значения: Политика, Спорт, Музыка, по умолчанию скрыто)

Валидация:

- i. Email должен содержать валидный адрес электронной почты
- ii. Если пользователь чекает «Хотите ли вы получать дополнительные новости», должно появиться поле «Список подписок». Пользователь обязан выбрать хотя бы одну.

Пятая секция «Проверка данных»:

Контролы:

- i. Таблица, в которой показаны все данные пользователя
- ii. Кнопка "Завершить"

Если пользователь попал на эту секцию, должна появиться кнопка «Завершить», которая по нажатию создает алерт с текстом «Регистрация прошла успешно». Если пользователь нажимает кнопку назад, кнопка «Завершить» не должна исчезать. Но алерт должен показываться только если все данные на секции валидны.

JQuery

Список литературы:

1. [jQuery](#)
2. [jQuery UI](#)

Задание:

Изменить предыдущее задание с использованием JQuery/JQuery UI.

1. Все ошибки теперь должны отображаться значком , а текст ошибки появляться в tooltip по наведению на значек.
2. Дата рождения должна использовать datePicker контрол
3. Поле возраст должно появляться только после выбора даты рождения
4. Итоговая таблица должна быть отображена с помощью «Accordion». Разбить его на 4 секции и отображать инпут только из текущей секции
5. По нажатию на кнопку завершения пользователь должен увидеть рорир окно с вопрос «_Значение из поле ФИО_ вы уверены, что хотите закончить регистрацию?» и опциями «да» и «нет». Если пользователь выберет «Да» то надо отчистить все значения и открыть первую секцию, если – «Нет» то просто вернуть на страницу результатов откуда бы пользователь не нажал кнопку «завершить»

Лекция 8

AJAX/UpdatePanel/WebServices

Основные моменты:

1. UpdatePanel: для чего нужна, как используется, ограничения
2. Web Services: виды
3. XMLHttpRequest: кроссбраузерность

Список литературы:

1. [Мак-Дональд М., Фримен А., Шпушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов \(Chapter 29, Chapter 30\)](#)
2. [Dino Esposito - Programming with ASP.NET 4.0 \(Chapter 20\)](#)

Задание:

Необходимо реализовать страницу "Список Товаров". Требования: изначально отображается 3 товара (картинка, краткое описание, цена); кнопка "далее", по нажатию на которую асинхронно подгружается ещё 3 товара. кнопка активна, пока товары не закончились. Приблизительный внешний вид:



[Acer Aspire Switch 10 SW5-012-1209 \(NT.L6UEU.004\)](#)

9 527 грн.



[Asus Transformer Book T100TA 64GB+500GB \(T100TA-DK007H\)](#)

12 162 грн.



[Asus Transformer Book T100TA 64GB \(T100TA-DK003H\)](#)

6 999 грн.

Также необходимо реализовать страницу "Управление товарами", которая предоставляет возможности добавления, редактирования и удаления товаров из базы (картинка, описание, цена)

Лекция 9

Аутентификация и Авторизация

Основные моменты:

1. Аутентификация\Авторизация: в чем отличия.
2. Виды Аутентификации в .Net
3. Membership, Simple Membership

Литература:

1. [Pro ASP.NET 2.0 C# 2005 Part 4 Security](#)
2. [Хорошая статья](#)

Задание:

Создать сайт, использующий Membership для аутентификации пользователей. На этом сайте разместить 2 контрола для менеджмента пользователей.

1 контрол должен отображать список всех пользователей в Membership с возможностью выбора/удаления пользователя.

После выбора 2 контрол должен отобразить информацию для редактирования.

Имя пользователя (не изменяемо), Email, Пароль, Подтверждение Пароля