

Введение в технологии Web и .NET.

Mark Sapossnek

CS 594
Computer Science Department
Metropolitan College
Boston University

Перевод: Сидякин И.М. МГТУ Кафедра ИУ-3 sidiakin@iu3.bmstu.ru 2003

Предварительные требования

- ♦ Навыки работы с компьютером
- ♦ Опыт работы с World Wide Web
- ♦ Опыт разработки объектно-ориентированных программ

2

Задачи курса

- ♦ Обзор Web и Internet технологий
- ♦ Обзор современных технологий программирования для Web
- ♦ Введение в проектирование для .NET

3

План лекций

- ♦ **Internet технологии**
- ♦ Языки программирования и принципы программирования
- ♦ Программирование для Web
- ♦ Обзор .NET

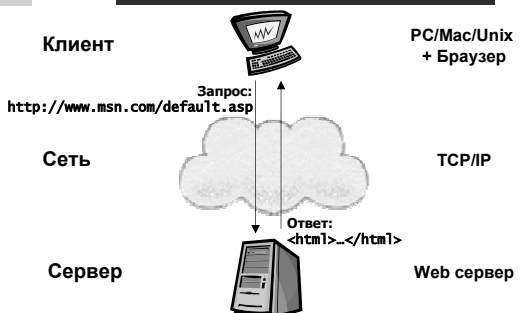
4

Internet технологии World Wide Web

- ♦ Способ распространения и получения информации
 - Технические статьи, реклама, рецепты, ...
- ♦ Большое число компьютеров объединённых в сеть: Internet
- ♦ Не только текст, но и графика
- ♦ Информация связана с другой информацией
- ♦ Платформа для разработки приложений
 - Магазин на диване
 - Поддержка потребителей и партнёров в режиме самообслуживания
 - ...

5

Internet технологии Архитектура www



6

Internet технологии Архитектура www

- ♦ Клиент/Сервер, принцип Запрос/Ответ
 - Вы запрашиваете Web страницу
 - например <http://www.msn.com/default.asp>
 - HTTP запрос
 - Web сервер отвечает возвращая данные в виде Web страницы
 - HTTP ответ
 - Web страница описывается на языке HTML
 - Для идентификации страниц используется универсальный локатор ресурсов (URL)
 - Протокол: http
 - Web сервер: www.msn.com
 - Web страница: [default.asp](http://www.msn.com/default.asp)
 - Можно так же передать параметры: [?name=Leon](http://www.msn.com/?name=Leon)

7

Internet технологии Web стандарты

- ♦ Internet Engineering Task Force Проблемная группа проектирования Internet (IETF)
 - <http://www.ietf.org/>
 - Основана 1986
 - Request For Comments (RFC) по адресу <http://www.ietf.org/rfc.html>
- ♦ World Wide Web Consortium (W3C)
 - <http://www.w3.org>
 - Основан 1994 Tim Berners-Lee
 - Публикует технические отчёты и рекомендации

8

Internet технологии Принципы проектирования для Web

- ♦ *Функциональная совместимость*: Языки и протоколы Web должны быть совместимы друг с другом и независимы от аппаратной и программной платформы.
- ♦ *Способность к эволюции*: Web должен иметь способность вмещать новые, будущие технологии. Поощряется простота, модульность и расширяемость.
- ♦ *Децентрализация*: Улучшает масштабируемость и надёжность.

9

Internet технологии Язык гипертекстовой разметки (HTML)

- ♦ Язык разметки используется для описания процедуры отображения Web страниц
 - Разработан для отображения данных, а не для хранения/передачи
- ♦ Разбирается и отображается Web браузером
- ♦ Может содержать **ссылки** на изображения, документы и другие страницы
- ♦ Не расширяем
- ♦ Развился из стандартного обобщённого языка разметки (SGML)
- ♦ HTML 3.2, 4.01, XHTML 1.0

10

Internet технологии HTML формы

- ♦ Позволяют создавать интерактивные элементы пользовательского интерфейса
 - Кнопки
 - Поля ввода
 - Выпадающие списки
 - Поля выбора
- ♦ Пользователи заполняют форму и дают подтверждение
- ♦ При подтверждении данные на форме посылаются Web серверу по протоколу HTTP

11

Internet технологии Протокол передачи гипертекстовых файлов (HTTP)

- ♦ Протокол верхнего уровня использующийся для запроса и передачи данных
 - Таких как HTML страницы, картинки GIF, JPEG, документы Microsoft Word, документы Adobe PDF, и т.п.
- ♦ Протокол типа Запрос/Ответ
- ♦ Методы: GET, POST, HEAD, ...
- ♦ HTTP 1.0: простой
- ♦ HTTP 1.1: более сложный

12

Internet технологии HTTP запрос

Метод Файл Версия HTTP Заголовки

```
GET /default.asp HTTP/1.0
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Connection: Keep-Alive
If-Modified-Since: Sunday, 17-Apr-96 04:32:58 GMT
```

Пустая строка

Данные – отсутствуют для GET

13

Internet технологии HTTP ответ

Версия HTTP Код статуса Описание кода статуса Заголовки

```
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Content-Length: 2543
```

Данные

```
<HTML> Какие то данные... blah, blah, blah </HTML>
```

14

Internet технологии HTTP коды статуса сервера

Код	Описание
200	ОК
201	Создано
301	Перемещено
302	Временно перемещено
400	Непонятный запрос
401	Нет прав доступа
403	Запрещено нет прав доступа
404	Не найдено
500	Внутренняя ошибка сервера

15

Internet технологии HTTP

- ♦ HTTP не имеет средств хранения состояния (stateless protocol)
- ♦ Каждый запрос HTTP независим от предыдущего и последующего
- ♦ HTTP 1.1 использует сообщения об активности (keep-alive) для повышения эффективности
- ♦ Отсутствие средств хранения состояния сильно влияет на принципы создания приложений

16

Internet технологии Cookies

- ♦ Механизм для хранения небольших объёмов информации (до 4КБайт) на клиенте
- ♦ Cookie ассоциируется с определённым web узлом
- ♦ Cookie посылается в заголовке HTTP
- ♦ Cookie посылается с каждым запросом HTTP
- ♦ Может существовать в течение одной сессии (пока браузер не будет закрыт) или сохраняться
- ♦ Может в некоторый момент устареть

17

Internet технологии HTTPS

- ♦ Защищённая версия HTTP
- ♦ Позволяет клиенту и серверу обмениваться данными конфиденциально, так что данные не могут быть изменены или перехвачены
- ♦ Использует протокол защищённых сокетов (SSL)/защиту транспортного уровня (TLS)

18

Internet технологии URI, URL и URN

- ♦ Универсальный идентификатор ресурсов (URI = URL или URN)
 - Базовый термин для всех текстовых имён/адресов
- ♦ Универсальный локатор ресурсов (URL)
 - Набор записей URI который содержит прямые инструкции о том как получить доступ к ресурсу в Internet, например по http, ftp, gopher
- ♦ Универсальное имя ресурса (URN)
 - 1) URI не содержащий информации о расположении ресурса.
 - 2) Определённая схема идентифицирующая ресурсы например urn:schemas:httpmail:subject

19

Internet технологии

Многоцелевые расширения электронной почты в Internet (MIME)

- ♦ Определяет типы данных/документов
 - текст/чистый
 - текст/html
 - рисунок/gif
 - рисунок/jpeg
 - аудиозапись/x-pn-realaudio
 - аудиозапись/x-ms-wma
 - видеозапись/x-ms-asf
 - приложение/octet-stream

20

Internet технологии MIME

- ♦ Задаёт набор символов, например ASCII
- ♦ Поддерживает сообщения состоящие из нескольких частей
- ♦ Первоначально разработан для электронной почты, но так же используется в других технологиях, например в HTTP

21

Internet технологии Браузеры

- ♦ Клиентские приложения
- ♦ Запрашивают HTML с Web сервера и разбирают его
- ♦ Популярные браузеры:
 - Netscape
 - Internet Explorer
 - Opera
- ♦ Также называются Пользовательскими агентами

22

Internet технологии Клиенты и серверы

- ♦ Клиентский и серверный компьютер оба имеют:
 - ЦПУ
 - Память
 - Устройства ввода/вывода
 - Диски
 - Сеть
 - Шину
 - Многозадачную операционную систему
 - Приложения

23

Internet технологии Клиенты и серверы

- ♦ Клиенты
 - Обычно работают с одним пользователем
 - Оптимизированы для быстрого взаимодействия с пользователем
 - Пользовательский интерфейс, графика
- ♦ Серверы
 - Работают с несколькими пользователями
 - Оптимизируется производительность
 - Больше: процессоров (SMP), памяти, дисков (SAN), Устройств ввода/вывода
 - Предлагают службы (такие как Web, файлы, печать, базы данных, почта, факс, транзакции, telnet, каталог)

24

Internet технологии

Прокси серверы и брандмауэры

- ♦ Прокси сервер
 - Сервер располагающийся между клиентом (с запущенным браузером) и Internet
 - Улучшает производительность кэшируя часто используемые Web страницы
 - Может фильтровать запросы запрещая пользователям доступ к некоторым Web узлам
- ♦ Брандмауэр
 - Сервер расположенный между сетью и Internet для предотвращения неавторизованного доступа к сети из Internet

25

Internet технологии

Сети

- ♦ Сеть = набор независимых компьютеров связанных друг с другом
- ♦ Зачем нужна сеть?
 - Совместный доступ к ресурсам
 - Надёжность
 - Снижение затрат
 - Связь
- ♦ Web технологии добавляют:
 - Новую бизнес модель: электронную коммерцию, рекламу
 - Развлечения
 - Приложения без установки на клиентском компьютере

26

Internet технологии

Сети

- ♦ Типы сетей
 - internet: набор связанных сетей
 - Internet: особая мировая сеть созданная на протоколе TCP/IP, используется для связи компаний, университетов, правительств, организаций и частных лиц. Развилась из ARPANET, созданной Департаментом Обороны США.
 - intranet: внутренняя сеть компании или организации основанная на технологиях Internet
 - extranet: сеть основанная на технологиях Internet связывающая одну компанию с другой

27

Internet технологии

Сети

- ♦ Сетевые технологии определяются масштабом сетей:
 - Локальные сети (ЛВС/LAN): покрывают площадь в пределах нескольких километров. Используются топологии шина и звезда
 - Глобальные сети (ГВС/WAN): Могут покрывать страну или континент. В ГВС для соединения линий связи, как промежуточные устройства, используются маршрутизаторы

28

Internet технологии

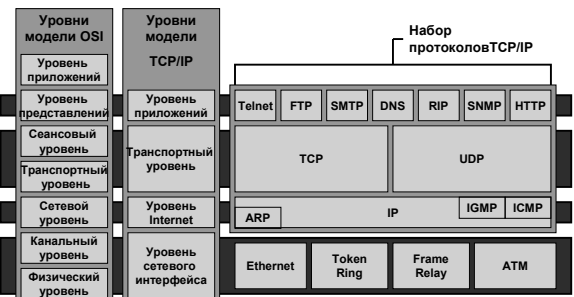
Сети

- ♦ Сетевые технологии
 - Широковещательная передача
 - Пакеты данных посылаются одной станцией и принимаются всеми станциями в сети
 - Групповая передача: пакеты принимаются группой станций в сети
 - Точка-точка
 - Пакеты передаются от одной станции к другой; может быть несколько путей передачи пакетов
 - Как правило, географически локализованные сети используют широковещательную передачу, а удалённые сети передачу точка-точка

29

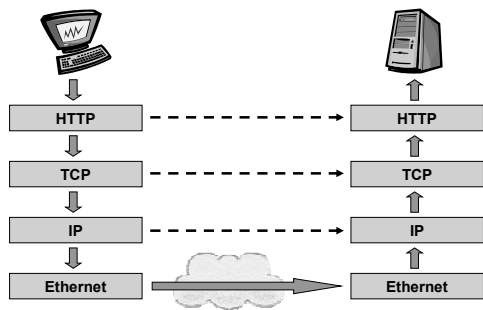
Internet технологии

Сети



30

Internet технологии Стек сетевых протоколов



31

Internet технологии Сети – Уровень Internet

- ♦ Internet протокол (IP)
- ♦ Отвечает за доставку пакетов от источника к приёмнику чрез несколько узлов
- ♦ Ненадёжный
- ♦ IP адрес: 32 значение обычно записываемое десятичными разделёнными точками 8-битовыми значениями (от 0 до 255); например 130.50.12.4

32

Internet технологии Сети – транспортный уровень

- ♦ Обеспечивает эффективную, надёжную и дешёвую службу доставки
- ♦ Использует программную модель сокетов
- ♦ Порт идентифицирует приложение
 - Хорошо известные порты соответствуют стандартным службам (например HTTP использует порт 80, SMTP использует порт 25)
- ♦ Протокол управления передачей (TCP)
 - Обеспечивает надёжную, ориентированную на соединение передачу потока байтов
- ♦ Протокол датаграмм пользователя UDP
 - Передача без подтверждения, ненадёжен

33

Internet технологии Сети – прикладной уровень

- ♦ Telnet: протокол виртуального терминала
- ♦ Протокол передачи файлов (FTP)
- ♦ Протокол передачи сетевых новостей (NNTP)
- ♦ Простой протокол управления сетью (SNMP)
- ♦ Простой протокол передачи электронной почты (SMTP)
- ♦ Почтовый протокол (POP3)
- ♦ Протокол доступа к сообщениям в сети Internet (IMAP)

34

Internet технологии Сети - Domain Name System (DNS)

- ♦ Создаёт удобные для пользователя имена доменов, например `www.msn.com`
- ♦ Иерархическое пространство имён с ограниченным набором корневых имён

▪ .com	▪ .org
▪ .net	▪ .mil
▪ .gov	▪ .jp
▪ .edu	▪ .de
- ♦ DNS серверы отображают имена доменов в IP адреса

35

Internet технологии Расширяемый язык разметки гипертекста (XML)

- ♦ Описывает иерархические структуры данных
- ♦ Мета язык: язык для описания других языков
- ♦ Расширяемый
- ♦ Полезен для обмена данными и трансформации данных
- ♦ Упрощённая версия стандартного обобщённого языка разметки SGML

36

План лекций

- ♦ Internet технологии
- ♦ **Языки и принципы программирования**
- ♦ Программирование для Web
- ♦ Обзор .NET

37

Языки программирования

- ♦ Машинный код
- ♦ Язык ассемблер
- ♦ Языки высокого уровня
 - Фортран, Лисп, Кобол
 - С, Паскаль, Бейсик, Smalltalk
 - C++, Eiffel
 - Java, C#
- ♦ Языки для написания скриптов
 - Shell scripts, Perl, TCL, Python, JavaScript, VBScript

38

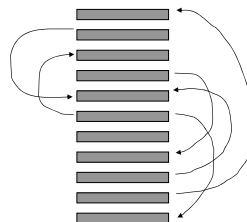
Принципы программирования

- ♦ Неструктурное программирование
- ♦ Структурное программирование
- ♦ Объектно-ориентированное программирование
- ♦ Компонентное программирование
- ♦ Программирование событий

39

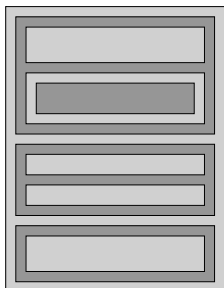
Принципы Неструктурное программирование

- ♦ Смотри "Go To Statement Considered Harmful"
<http://www.acm.org/classics/oct95/>



40

Принципы Структурное программирование



- ♦ Последовательность
- ♦ Условия
 - если то иначе
 - выбор
- ♦ Циклы
 - от i равно 1 до n
 - пока
 - до того как
- ♦ Функции
- ♦ Обработка исключений

41

Принципы Объектно-ориентированное программирование

- ♦ Объекты объединяют данные и поведение
 - Данные: члены, поля, переменные, слоты, свойства
 - Поведение: методы, функции, процедуры
- ♦ Работа с объектами проста
 - Сначала создаётся экземпляр объекта нужного типа
 - Затем вызываются его методы и читаются/задаются его свойства
- ♦ Разработка новых типов объектов может быть довольно сложна
 - Цели разработки часто вступают в противоречие :простота, функциональность, повторное использование, производительность

42

Принципы Объектно-ориентированное программирование

- ♦ Ключевые принципы ООП
 - Идентификация
 - Инкапсуляция
 - Данные + поведение
 - Скрытие информации (абстракция)
 - Классы и экземпляры объектов
 - Полиморфизм
 - Интерфейсы
 - Делегирование, агрегирование
 - Наследование
 - Модели

43

Принципы Компонентное программирование

- ♦ Компоненты
 - Независимо размещаемые и повторно используемые модули
 - Более крупная структура по сравнению с объектами (объекты это языковые конструкции)
 - Содержат несколько классов
 - Часто не зависят от языка программирования
- ♦ В общем случае разработчик и пользователь компонента не знакомы, не работают в одной компании, и не используют один и тот же язык программирования

44

Принципы Компонентное программирование

- ♦ Объектная модель компонентов (COM)
 - Первоначальный стандарт компонентов Microsoft
 - Определяет протоколы загрузки и использования компонентов: внутри процесса, между процессами или между машинами
 - Основа ActiveX, OLE, и многих других технологий
 - Может использоваться в Visual Basic, C++, .NET, ...
- ♦ Java Beans
 - Стандарт компонентов Java
 - Зависим от языка программирования

45

Принципы Программирование событий

- ♦ Когда происходит какое либо интересное нас событие в приложении вызывается процедура обработчик этого события
- ♦ События позволяют вам в вашем собственном коде перехватить действия другой системы
- ♦ Event = callback
- ♦ Пользовательские интерфейсы построены на событиях
 - onClick, onMouseOver, onMouseMove...
- ♦ События могут порождаться таймером, сетью, операционной системой, другими приложениями и т.п.

46

Программа лекций

- ♦ Internet технологии
- ♦ Языки и принципы программирования
- ♦ Программирование для Web
- ♦ Обзор .NET

47

Программирование для Web Клиентский код

- ♦ Что такое клиентский код?
 - Программы загружаемые с Web сервера браузером и затем исполняющиеся на клиенте
- ♦ Зачем нужен клиентский код?
 - Лучшая масштабируемость: меньше работы на сервере
 - Лучшее соотношение производительность/опыт оператора
 - Можно создавать конструкции пользовательского интерфейса не возможные в HTML
 - Выпадающие и всплывающие меню
 - Закладки в диалоговых окнах
 - Дополнительные эффекты, например, анимация
 - Проверка данных

48

Программирование для Web Клиентский код

- ♦ DHTML/JavaScript
- ♦ COM
 - Элементы управления ActiveX
 - Компоненты COM
 - Служба удалённого доступа к данным (RDS)
- ♦ Java
- ♦ Plug-ins
- ♦ Помощники (Helpers)
- ♦ Удалённое исполнение скриптов (Remote Scripting)

49

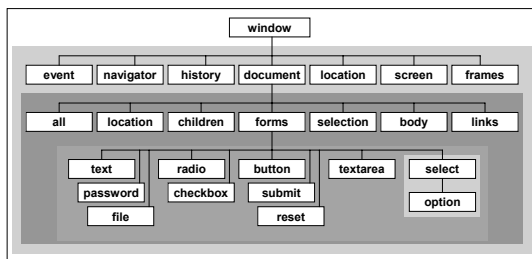
Программирование для Web Динамический HTML (DHTML)

- ♦ Скрипт встроенный в HTML страницу
- ♦ Обычно написанный для совместимости на JavaScript (ECMAScript, JScript)
 - Internet Explorer так же поддерживает VBScript другие языки скриптов
- ♦ Каждый элемент HTML становится объектом с набором событий (например onClick)
- ♦ Скрипт содержит код обработчиков событий браузера

50

Программирование для Web DHTML

- ♦ Объектная модель документа DHTML (DOM)



51

Программирование для Web ActiveX

- ♦ Основа - технология COM
- ♦ Встроенная поддержка только в Internet Explorer
 - Поддерживается в Netscape с помощью plug-in
- ♦ Хорошая технология если вы знаете своих пользователей (например в intranet) или можете указать им какой браузер использовать
- ♦ Небольшой эффективный код

52

Программирование для Web Java Applet

- ♦ Основа - технологии Java bytecode
- ♦ Многообещающий способ компактной и лёгкой загрузки клиентского кода:
 - "Один раз пишется, работает везде"
- ♦ Безопасный: код выполняется в "ящике с песком"
- ♦ Вопросы совместимости и производительность тормозят распространение

53

Программирование для Web Серверный код

- ♦ Что такое серверный код?
 - Программа работающая на сервере а не на клиенте
 - Воспринимает входную информацию через
 - Параметры URL
 - Данные HTML-формы
 - Cookies
 - Заголовки HTTP
 - Может обращаться к серверным базам данных, серверам электронной почты, файлам и т.п.
 - Динамически создаёт HTML ответ для клиента

54

Программирование для Web Серверный код

- ♦ Зачем нужен серверный код?
 - Доступность
 - Вы можете войти в Internet с любого браузера, с любого устройства, в любое время, повсюду
 - Управляемость
 - Не требует распространения прикладного кода
 - Легко модифицировать код
 - Безопасность
 - Исходный текст программ не показан
 - Авторизация позволяет ограничить действия пользователя
 - Масштабируемость
 - Трёхуровневая (3-tier) структура построенная на основе Web технологий может расширяться

55

Программирование для Web Серверные технологии

- ♦ Common Gateway Interface (CGI)
- ♦ Internet Server API (ISAPI)
- ♦ Netscape Server API (NSAPI)
- ♦ Active Server Pages (ASP)
- ♦ Java Server Pages (JSP)
- ♦ Personal Home Page (PHP)
- ♦ Cold Fusion (CFM)
- ♦ ASP.NET

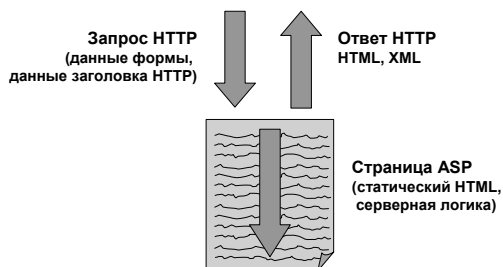
56

Программирование для Web Active Server Pages (ASP)

- ♦ Технология позволяющая легко создать серверное приложение
- ♦ ASP страницы пишутся на языке скриптов, обычно VBScript или JScript
- ♦ Страница ASP содержит статическую HTML последовательность перемежающуюся с серверным кодом
- ♦ ASP скрипт обычно работает с базой данных

57

Программирование для Web ASP



58

Программа лекций

- ♦ Internet технологии
- ♦ Языки и принципы программирования
- ♦ Программирование для Web
- ♦ **Обзор .NET**

59

Обзор .NET

- ♦ Введение в .NET
- ♦ Web службы (Web Services)
- ♦ .NET Framework
- ♦ Среда периода исполнения Common Language Runtime
- ♦ Пользовательский интерфейс Windows Forms
- ♦ Пользовательский интерфейс Web Forms
- ♦ ADO.NET
- ♦ Языки программирования

60

Введение в .NET

Что такое .NET?

- ♦ Взгляд на то как будут развиваться информационные технологии
- ♦ Платформа которая поддерживает это взгляд
- ♦ Бизнес-модель программного обеспечения как службы

61

Введение в .NET

Что такое .NET?

- ♦ Взгляд на будущее информационных технологий
 - Web узлы будут объединяться Web службами
 - К персональным компьютерам добавятся новые устройства
 - Пользовательский интерфейс станет более адаптирующимся и настраиваемым

62

Введение в .NET

Что такое .NET?

- ♦ Платформа
 - .NET Framework
 - Visual Studio.NET
 - Серверы .NET Enterprise
 - Database, Messaging, Integration, Commerce, Proxy, Security, Mobility, Orchestration, Content Management
 - Службы строительных блоков .NET
 - Passport
 - .NET My Services ("Hailstorm")
 - Цель: значительно облегчить процесс создания мощных Web приложений и Web служб

} Предмет изучения
в этом курсе

63

Введение в .NET

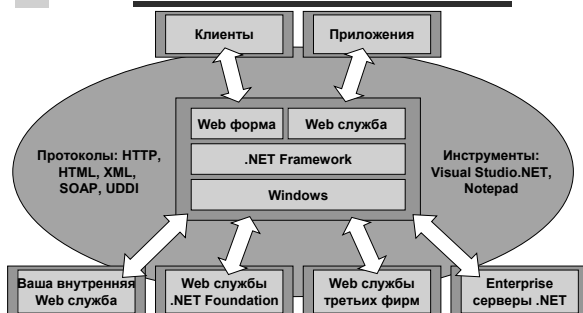
Что такое .NET?

- ♦ Бизнес модель
 - Программа как служба
 - Службы по подписке
 - Поддержка приложений, например, bCentral

64

Введение в .NET

Платформа .NET



65

Web службы

- ♦ Программируемый компонент приложения доступный по стандартным Web протоколам
- ♦ Центральная часть архитектуры .NET
- ♦ Предоставляет свои функции в Web
- ♦ Построено на существовавших ранее и новых технологиях
 - HTTP, XML, SOAP, UDDI, WSDL, ...

66

Web службы Эволюция Web



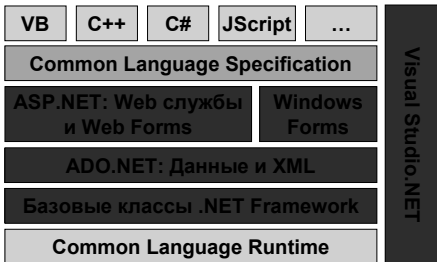
67

.NET Framework Что такое .NET Framework?

- ♦ Ряд технологий для разработки и использования компонентов предназначенных для создания:
 - Web форм
 - Web служб
 - Приложений Windows
- ♦ Поддерживает жизненный цикл ПО
 - Разработка
 - Отладка
 - Установка
 - Сопровождение

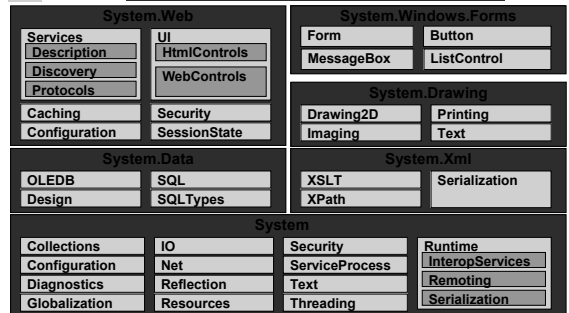
68

.NET Framework .NET Framework и Visual Studio.NET



69

.NET Framework Классы .NET Framework



70

Common Language Runtime Задачи

- ♦ Службы поддержки проектирования
 - Глубокая межязыковая функциональная совместимость
 - Улучшенная производительность
- ♦ Службы поддержки установки
 - Простая и надёжная установка
 - Меньше проблем с версиями – КОНЕЦ 'DLL HELL'
- ♦ Службы поддержки периода исполнения
 - Производительность
 - Надёжность
 - Масштабируемость
 - Защита
 - Доступность
 - Безопасность

71

Common Language Runtime Компиляция



72

Common Language Runtime Сборки (Assemblies)

- Сборка
 - Логическая единица установки
 - Содержит Манифест, Метаданные, MSIL и ресурсы
- Манифест
 - Метаданные о компонентах в сборке (версия, типы, зависимости, и т.п.)
- Метаданные Типов
 - Полностью описывают все типы имеющиеся в сборке: свойства, методы, аргументы, возвращаемые значения, атрибуты, базовые классы, ...

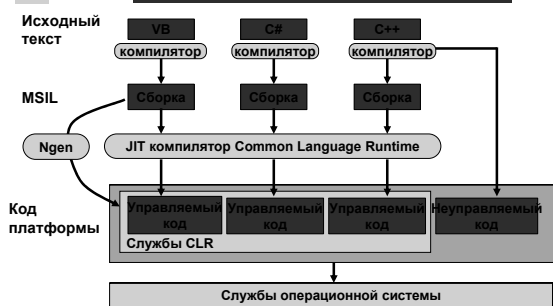
73

Common Language Runtime Сборки

- ♦ Промежуточный язык Microsoft (MSIL, IL)
 - Все языки компилируются в IL (управляемый код)
 - IL всегда перед исполнением компилируется в код платформы
- ♦ Ресурсы
 - например .bmp, .jpg

74

Common Language Runtime Модель исполнения



75

Common Language Runtime Службы

- ♦ Управление кодом
- ♦ Преобразование MSIL в код платформы
- ♦ Загрузка и исполнение управляемого кода
- ♦ Создание и управление метаданными
- ♦ Проверка безопасности типов
- ♦ Вставка и исполнение проверок защиты
- ♦ Управление памятью и изоляция
- ♦ Обработка исключений в разных языках
- ♦ Взаимодействие между объектами .NET Framework, объектами COM и Win32 DLL
- ♦ Автоматизация структуры объекта для позднего связывания
- ♦ Службы поддержки разработки (профилирование, отладка, и т.п.)

76

Common Language Runtime Многоязыковая поддержка

- Общая система типов (CTS)
 - Множество типов включающее все использующиеся в большинстве современных языков программирования
- Общая спецификация языка (CLS)
 - Подмножество CTS которое позволяет взаимодействовать коду написанному на разных языках
- Какие языки?
 - Microsoft: C++, Visual Basic, C#, JScript
 - Третьих фирм: Cobol, Eiffel, Smalltalk, Scheme, Oberon, Haskell, Java, Python, Perl, ...

77

Common Language Runtime Приложения

- ♦ Приложения состоят из одной или нескольких сборок
- ♦ Как сборки подключаются друг к другу?
 - Определяется метаданными и политикой
 - Локально (предпочтительно)
 - Глобальный кэш сборки
- ♦ Несколько версий сборки могут совместно существовать на одной машине
 - Легкая установка, обновление и удаление
 - Несколько версий сборки даже могут использоваться одним приложением

78

Common Language Runtime Защита

- ♦ Доказательная защита (аутентификация)
- ♦ Основана на идентификации пользователя и кода
- ♦ Конфигурируемые политики
- ♦ Императивные и декларативные интерфейсы

79

Windows Forms

- ♦ Основа для создания полнофункциональных клиентских приложений
- ♦ Надстройка над .NET Framework
- ♦ Быстрый цикл разработки приложений (RAD)
- ♦ Визуальное наследование
- ♦ Anchoring и docking
- ♦ Широкий набор элементов управления
- ♦ Расширяемые элементы управления
- ♦ Поддержка работы с данными
- ♦ Легко взаимодействует с Web службами
- ♦ Поддержка ActiveX
- ♦ Поддержка лицензирования
- ♦ Поддержка печати
- ♦ Улучшенная графика

80

Web Forms

- ♦ Создаются с помощью ASP.NET
 - Логическая эволюция ASP
 - Та же модель проектирования: отредактировать страницу и запустить
- ♦ Меньше программного кода
- ♦ Новая программная модель
 - Управляемые событиями серверные элементы управления
 - Сложные элементы управления (таблицы данных, проверка)
 - Связывание данных
 - Элементы генерируют специфичный для браузера код
 - Упрощенное управление состоянием страницы

81

Web Forms

- ♦ Позволяют разделить пользовательский интерфейс и логику работы
- ♦ Используют языки .NET
 - Не только скрипты
- ♦ Легко применяемые компоненты
- ♦ Установка XCOPY/FTP
- ♦ Простая конфигурация (на основе XML)

82

Web Forms

- ♦ Кэширование (страницы, фрагменты, по настройке)
- ♦ Масштабируемое управление состоянием сеанса
- ♦ Поддержка трассировки
- ♦ ASP.NET расширяемая система
 - Нет разделения технологий как в ISAPI / ASP
- ♦ Аутентификация с помощью форм

83

ADO.NET

- ♦ Аналогично ADO, но структура интерфейсов лучше
- ♦ Доступ к данным не зависит от языка
- ♦ Поддерживает два типа доступа к данным
 - Автономный
 - Доступ только на чтение и только вперед
- ♦ Поддерживает привязку данных
- ♦ DataSet: набор таблиц
- ♦ Может просматривать и обрабатывать реляционные (таблицы) или иерархические (XML) структуры данных

84

Языки программирования C#

- ♦ Новый язык созданный для .NET
- ♦ Безопасная, продуктивная эволюция C++
- ♦ Ключевые концепции:
 - Компоненто-ориентированный
 - Всё представлено в виде объектов
 - Стойкий и надёжный код
 - Сохраняет ваши вложения
- ♦ Подан на рассмотрение ECMA для стандартизации
- ♦ Использует классы .NET Framework

85

Языки программирования Visual Basic.NET

- ♦ Модернизированный и упрощённый Visual Basic
 - Наследование
 - Поток
 - Обработка исключений
- ♦ Поддержка позднего связывания
- ♦ Использует классы .NET Framework

86

Заключение

- ♦ Internet технологии
- ♦ Языки и принципы программирования
- ♦ Программирование для Web
- ♦ Обзор .NET

87

Дополнительные ресурсы

- ♦ HTTP
 - <http://msdn.microsoft.com/library/periodic/period96/protocol.htm>
 - *HTTP Essentials*, Stephen Thomas, 2001, Wiley, ISBN 0471-39823-3
- ♦ Cookies
 - <http://msdn.microsoft.com/library/default.asp?URL=/library/partbook/instantj/cookies.htm>
- ♦ MIME
 - http://www.ufaq.org/navcom/mime_tutorial.html
 - <http://www.irvine.com/~mime/>
- ♦ Сети
 - <http://msdn.microsoft.com/library/periodic/period99/ntp99b3.htm>

88

Дополнительные ресурсы

- ♦ XML
 - <http://msdn.microsoft.com/xml/default.asp>
 - <http://www.w3.org/XML/>
 - *Essential XML*, Don Box, Aaron Skonnard, John Lam, Addison Wesley, 2000, ISBN 0-201-70914-7
- ♦ .NET
 - <http://www.microsoft.com/net/>
 - <http://msdn.microsoft.com/net/>
 - <http://www.gotdotnet.com>
 - msnews.microsoft.com news server
 - microsoft.public.dotnet.general@newsgroup

89