

Облачные вычисления

№	Темы практические занятий:
	Краткий обзор платформ виртуализации: сравнение основных характеристик виртуальных машин.
	Azure Services Platform: подготовка рабочего места Visual Studio: установка и настройка программного обеспечения, знакомство с инструментарием Azure Services Platform, создание первого облачного приложения
	Azure Services Platform: создание проекта Visual Studio: создание и настройка простого облачного приложения, свойства рабочей роли и веб-роли, запуск приложения в режиме эмуляции
	Структурированное хранилище данных Windows Azure Table: изучение процедуры настройки хранилища данных Windows Azure Table и принципов работы с ним в Visual Studio при помощи Compute Emulator.

Лабораторно-практическая работа №1

Тема: Краткий обзор платформ виртуализации.

Цель: провести сравнение основных характеристик виртуальных машин.

Задание: необходимо выбрать и сравнить 3-5 виртуальные машины. Выбирать необходимо только те виртуальные машины, которые являются гипервизорами (эмуляторами аппаратного обеспечения). Критерии, по которым необходимо провести сравнение и пример рассмотрены в разделе «Ход работы».

Ход работы

Виртуальная машина (VM, virtual machine) — программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы (target — целевая, или гостевая платформа) и исполняющая программы для target-платформы на host-платформе (host — хост-платформа, платформа-хозяин) или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы.

Пример:

Название	<i>AlphaVM-Free</i>	<i>VirtualBox</i>	<i>Hyper-V</i>
----------	---------------------	-------------------	----------------

Создатель	<i>EmuVM</i>	<i>Innotek, подразделение Sun Microsystems (в 2010 Sun куплена Oracle)</i>	<i>Microsoft</i>
Процессор машины-носителя	<i>x86, x86-64</i>	<i>Intel x86, AMD64</i>	<i>Intel x86-64 (обязательно Intel VT-x; VT-d требуется лишь в Hyper-V 2012 для использования SR-IOV), AMD64 (обязательно AMD Pacifica)</i>
Гостевой процессор	<i>DEC Alphaserpver</i>	<i>Такой же, как и на машине-носителе. ver. 5.0.14</i>	<i>Intel x86, AMD64</i>
ОС машины-носителя	<i>Windows, Linux, FreeBSD</i>	<i>MS Windows, Linux, Solaris, OpenSolaris, Mac OS X, FreeBSD</i>	<i>Нет (также может работать под Windows Server 2008 (x64))</i>
Официально поддерживаемые гостевые ОС	<i>OpenVMS, Tru64</i>	<i>DOS, OS/2, MS Windows, Linux, Solaris, OpenSolaris, OpenBSD, FreeBSD, NetBSD, Netware, QNX, L4, Mac OS X</i>	<i>Windows NT, 2000, 2003, 2008, XP, Vista, Linux (Red Hat and SUSE)</i>
Поддержка любой ОС	<i>Нет</i>	<i>Есть</i>	<i>Есть</i>
Поддержка драйверов в гостевой ОС	<i>Нет данных</i>	<i>Есть</i>	<i>Есть</i>
Принцип действия	<i>Нет данных</i>	<i>Динамическая recompilation (основана на QEMU)</i>	<i>Аппаратная виртуализация</i>
Лицензия	<i>freeware</i>	<i>Свободная и проприетарная версии (GPL, PUEL)</i>	<i>Проприетарная (Free)</i>
Типичное применение	<i>Исследования, увлечение</i>	<i>Увлечение, разработка, тестирование</i>	<i>Консолидация серверов, организация непрерывной работы, тестирование</i>
Скорость работы гостевой ОС в сравнении с ОС носителя	<i>Медленная</i>	<i>Практически без потерь, если используются расширения</i>	<i>Близка к производительности системы-носителя / без потерь</i>

Требования к содержанию отчета:

титульный лист;

тема, цель, задание;

ход работы — сравнительная таблица выбранных виртуальных машин;

вывод — развернутая формулировка вывода из проделанной работы.

Лабораторно-практическая работа №2

Тема: Azure Services Platform: подготовка рабочего места Visual Studio

Цель: подготовка рабочего места для *stand-alone* разработки облачных приложений; знакомство с основными инструментами разработчика.

Задание: выполнить установку необходимых системных и служебных программ и инструментальных средств для реализации облачных вычислений.

Ход работы

Загрузить инструменты для работы можно по следующей ссылке: <http://www.microsoft.com/downloads/ru-ru/details.aspx?FamilyID=7a1089b6-4050-4307-86c4-9dadaa5ed018#QuickDetails>

Требования к установке Windows Azure SDK

Операционная система.

Windows Azure SDK 1.3 от ноября 2010 года официально поддерживается следующими операционными системами:

Windows 7 (в т.ч. Enterprise; Home Premiumx86, x64; Professional x86, x64; *Ultimate* x86, x64);

Windows Server 2008 + SP2, Windows Server 2008 R2;

Windows Vista + SP2.

Программное обеспечение для Windows Azure SDK:

1. .Net Framework 3.5 + SP1
2. IIS 7.0
3. Microsoft SQL Server 2005 Express
4. *Microsoft SQL Server 2008* (в т.ч. Express)
5. Исправление WCF <http://support.microsoft.com/kb/971842> (при отсутствии SP1 для Windows 7 и Windows Server 2008 R2 - <http://www.microsoft.com/downloads/ru-ru/details.aspx?FamilyID=c3202ce6-4056-4059-8a1b-3a9b77cdfdda>)
6. Windows PowerShell (не для Windows 7). <http://support.microsoft.com/kb/968929>
7. Модуль URL Rewrite <http://www.iis.net/download/URLRewrite>

Если все указанные требования соблюдены, то можно приступать к непосредственной установке инструментов для работы с Visual Studio 2010.

Подготовка рабочего места:

1. Установить VS 2010 и MS SQL Server 2008 (или MS SQL Server 2008 R2). Установка данного инструментария подробно описывается во множестве ресурсов и, как правило, не вызывает затруднений.

2. Настройка IIS.

для Windows 7

Открыть панель управления (Пуск - Панель управления)

Выбрать "Программы и компоненты"

Раскрыть узел "Microsoft .Net Framework 3.5" и включить элемент "Windows Communication Foundation HTTP Activation"

Последовательно раскрыть узлы "Службы IIS", "Службы Интернета" и "Компоненты разработки приложений", отметить элементы "ASP.NET" и "CGI".

В узле "Службы IIS" развернуть "Службы Интернета" и "Общие функции HTTP". Отметить элемент "Статическое содержимое".

В узле "Службы IIS" развернуть "Средства управления веб-сайтом" и отметить "Консоль управления IIS".

После этого нажать "ОК" и дождаться завершения процесса установки отмеченных компонент.

Для Windows 2008

Запустить диспетчер сервера (Пуск - Администрирование - Диспетчер сервера).

Щелкнуть правой кнопкой мыши на узле "Компоненты" и выбрать "Добавить компоненты"

В списке компонентов, в узле "Возможности .Net Framework 3.0" отметить ".Net Framework 3.0".

Аналогичным образом выбрать элемент "Активация HTTP" в узле "Активация WCF" и нажать "Далее". В случае, если появится диалоговое окно установки служб для данных компонент, установить их.

Дождаться окончания установки и перейдите к узлу "Роли" диспетчера сервера.

Установить роль "Веб-сервер", нажав "Добавить роли".

Последовательно нажимая Далее" установить необходимую роль.

В узле "Роли" выберите элемент "Веб-сервер(IIS) и нажать "Добавить службы ролей"

Выделить службы "Статическое содержимое", "ASP.NET" и "CGI" в узлах "Веб-сервер" и "Разработка приложений" соответственно, если они еще не установлены. Нажать "Далее", затем "Установить".

Дождаться завершения процесса установки.

3. Установить VSCloudService.exe или Windows Azure SDK 1.3 (SDK входит в состав VSCloudService.exe).

4. В случае использования 32-битной ОС, установить исправление №5 списка требуемого программного обеспечения (см. выше).

5. Установить исправления №6 и №7 из списка требуемого программного обеспечения (см. выше).

На этом установку инструментария *stand-alone* разработки облачных приложений можно считать завершенной.

Знакомство с инструментарием

Эмуляторы *Compute Emulator* (Development Fabric) и *Storage Emulator* (development Storage) можно найти в папке Windows Azure SDK\v1.3 (по умолчанию C:\Program Files\Windows Azure SDK\v1.3\). Эмуляторы располагаются в подкаталоге bin.

Если установка инструментария завершена корректно, то в списке проектов VS2010 появится шаблон *Cloud* . Выбор проекта *Windows Azure* приведет к появлению списка доступных ролей.

Таблица 1.1. Список поддерживаемых ролей:	
Роль	Описание
Веб-роль ASP.NET	Основанное на ASP.NET приложение с веб-интерфейсом
Веб-роль ASP.NET MVC 2	Основанное на ASP.NET MVC 2 приложение с веб-интерфейсом

Веб-роль ASP.NET службы WCF	WCF - сервис
Рабочая роль	Создание фоновой задачи
Веб-роль CGI	Хостинг приложения с использованием FastCGI

Создание первого облачного приложения будет рассмотрено в следующей практической работе.

Требования к содержанию отчета:

- титульный лист;
- тема, цель, задание;
- ход работы — иллюстрации основных этапов выполнения работы с краткими пояснениями;
- вывод — развернутая формулировка вывода из проделанной работы.

Перечень полезных ссылок

Установка Visual Studio 2010

1. <http://msdn.microsoft.com/ru-ru/library/e2h7fzkw.aspx>

Установка MS SQL Server 2008

1. <http://www.alt.ru/mssqlserver2008.php>
2. [http://itband.ru/2010/07/install-microsoft-sql-server-2008-r2/\(версииR2\)](http://itband.ru/2010/07/install-microsoft-sql-server-2008-r2/(версииR2))

Руководства по устранению неполадок

1. <http://msdn.microsoft.com/ru-ru/library/ee460770.aspx>

Лабораторно-практическая работа №3

Тема: Azure Services Platform: создание проекта Visual Studio

Цель: освоение принципов создания проекта облачного решения и изучение особенности его запуска и контроля состояния при помощи Compute Emulator.

Задание: выполнить установку необходимых системных и служебных программ и инструментальных средств для реализации облачных

вычислений.

Ход работы

Для работы инструментов создания облачных решений необходимо запустить VS с правами администратора:

Запустить VS 2010 и создать проект облачной службы с именем WA_first.

Добавим к проекту решения веб-роль ASP.NET и рабочую роль:

Результатом будет решение, состоящее из трех проектов:

Файлы *ServiceConfiguration.cscfg* и *ServiceDefinition.csdef* являются конфигурационными и используются для определения характеристик облачного решения и его ролей. Конфигурационные файлы упаковываются вместе с кодом и разворачиваются в Windows Azure.

ServiceDefinition.csdef - используется для описания приложения и ролей, хранит настройки одинаковые для всех экземпляров ролей. После запуска приложения содержимое данного файла не может быть изменено.

Файл *ServiceConfiguration.cscfg* задает значения настроек, описанных в файле *ServiceDefinition.csdef*, указывает число экземпляров каждой роли. Содержимое данного файла может быть изменено и после запуска роли.

Страницу свойств роли Windows Azure можно открыть, щелкнув правой кнопкой мыши на роли и выбрав "Свойства":

Рассмотрим свойства нашей рабочей роли:

В разделе "Конфигурация" можно указать уровень доверия .NET. Полное доверие необходимо для выполнения собственного кода приложений *FastCGI*. Частичное доверие отключает возможность загрузки и использования клиентских библиотек *Windows Azure*.

Значение настройки "число экземпляров" определяет количество экземпляров, которые должны быть запущены для службы этой роли.

Свойства размер виртуальной машины (ВМ) определяет характеристики автоматически создаваемой виртуальной машины.

Поскольку мы создаем приложение локально, в учетных данных хранилища в разделе "Диагностика" нужно отметить пункт "Использование эмулятора хранилища Windows Azure", либо вручную ввести строку "UseDevelopmentStorage=true".

Запуск приложения в режиме эмуляции:

Нажав клавишу F5, дождемся окончания построения облачного решения.

Результатом будет веб-страница на локальном сервере.

В правой части панели задач должна появиться иконка *Windows Azure*. Для просмотра информации о работе приложения в режиме эмуляции, нужно

щелкнуть правой кнопкой мыши по значку *Windows Azure* и выбрать "*Show Compute Emulator UI*"

Появится окно для просмотра статусов ролей нашего приложения:

Compute emulator – это основной компонент *Windows Azure*, можно сказать, ядро "облачной" операционной системы, отвечающий за управление виртуальными машинами и экземплярами ролей. Развертывание и старт экземпляра роли начинают ее жизненный цикл.

Подробнее работа по созданию более сложных приложений будет рассмотрена в следующих лабораторных работах.

Требования к содержанию отчета:

титульный лист;

тема, цель, задание;

ход работы — иллюстрации основных этапов выполнения работы с краткими пояснениями;

вывод — развернутая формулировка вывода из проделанной работы.

Перечень полезных ссылок

Настройка облачной службы

1. <http://msdn.microsoft.com/ru-ru/library/ee405486.aspx>

Уровни доверия .Net

1. <http://msdn.microsoft.com/ru-ru/library/dd573355.aspx>
2. <http://msdn.microsoft.com/ru-ru/library/dd573345.aspx>

Руководства по устранению неполадок

1. <http://msdn.microsoft.com/ru-ru/library/ee460770.aspx>

Лабораторно-практическая работа №4

Тема: Структурированное хранилище данных Windows Azure Table

Цель: изучение процедуры настройки хранилища данных Windows Azure Table и принципов работы с ним в Visual Studio 2010 при помощи Compute Emulator.

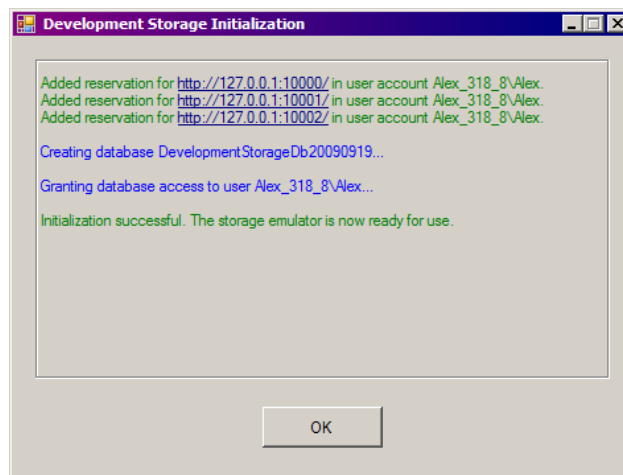
Ход работы

Рассмотрим более подробно работу с Storage Emulator. По умолчанию

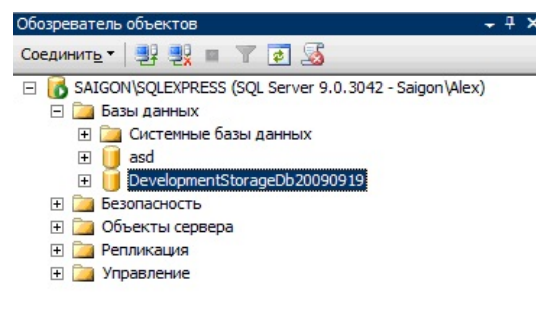
Storage Emulator устанавливается в папку devstore подкаталога bin, папки Windows Azure SDK. Либо в папку devstore подкаталога Emulator, папки Windows Azure.

Здесь можно найти два .exe файла:

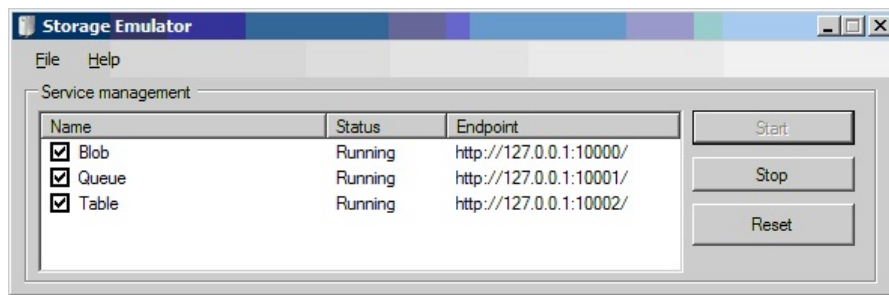
1. DSInit — инициализирует локальное хранилище и устанавливает права доступа к нему. Запустив этот файл, при отсутствии ошибок, должно появиться следующее окно:



Как видно, была создана локальная база данных для разработки, и зарезервированы порты 10000–10002. В том, что база создана можно также убедиться, запустив SQL Management Studio



2. DSService.exe — непосредственно запускает эмулятор облачного хранилища. Запустив его и подождя некоторое время, можно заметить, что в правом нижнем углу появился значок Windows Azure. Для того, чтобы открыть интерфейс эмулятора, нужно щелкнуть правой кнопкой мыши по значку Windows Azure и выбрать "Show Compute Emulator UI". При отсутствии ошибок, должно появиться следующее окно:



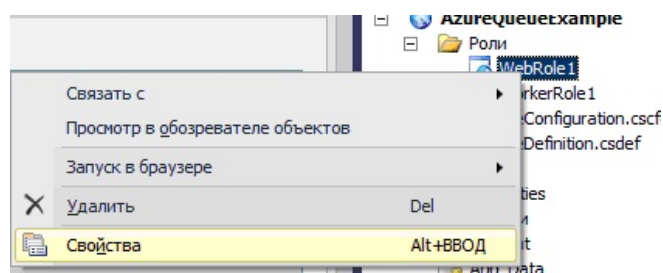
В окне Storage Emulator отображается состояние и конечные точки сервисов эмулятора: Blob, Queue и Table. Сервисы можно запустить, остановить, либо сбросить, с потерей данных, хранящихся в них.

Важно отследить следующий момент: порты, указанные для каждого из сервисов должны быть свободны. В случае, если при запуске хранилища разработки появляется ошибка.

Это скорее всего означает, что указанные порты "слушают" другие приложения.

Подключение к хранилищу разработки

Для подключения к эмулятору хранилища при разработке приложения (в среде Visual Studio 10 в нашем случае) после создания проекта необходимо перейти к свойствам роли



А затем во вкладке "Параметры" добавить параметр строки подключения и нажать кнопку "..."



И в появившемся окне задать параметры подключения и используемой учетной записи Windows Azure, например, такие:

Создание строки подключения к хранилищу

Подключиться с помощью:

- ☐ Эмулятор хранилища Windows Azure
- ☐ Ваша подписка
- ☒ Введенные вручную учетные данные

Чтобы создать строку подключения к хранилищу, введите имя и ключ учетной записи. Для использования настраиваемых конечных точек должна быть указана одна из конечных точек большого двоичного объекта, таблицы или хранилища.

Имя учетной записи:

Ключ учетной записи:

Подключение:

- ☒ Использовать HTTPS
- ☐ Использовать HTTP
- ☐ Укажите настраиваемые конечные точки

Предварительный просмотр строки подключения:

[Заявление о конфиденциальности](#)

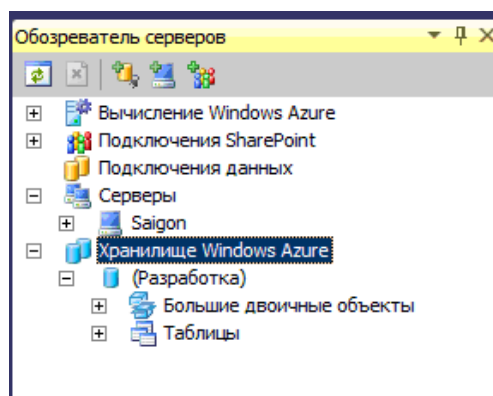
В результате в проекте будут такие параметры веб-роли:

Конфигурация службы:

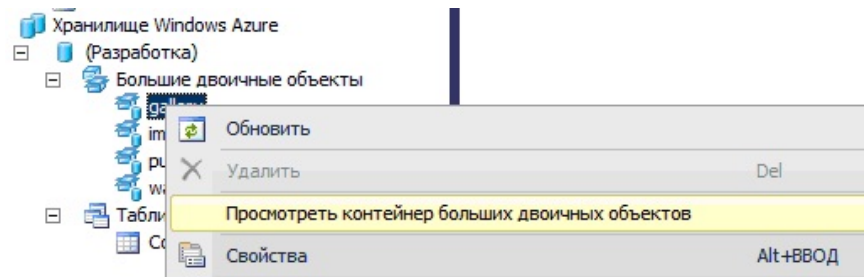
Добавление параметров конфигурации, доступных с помощью программных средств и допускающих динамическое обновление.

Имя	Тип	Значение
Microsoft.WindowsAzure.Plugins...	Строка подключен...	UseDevelopmentStorage=true
► DataConnectionString	Строка	DefaultEndpointsProtocol=https;AccountName=Alex;AccountKey=key1

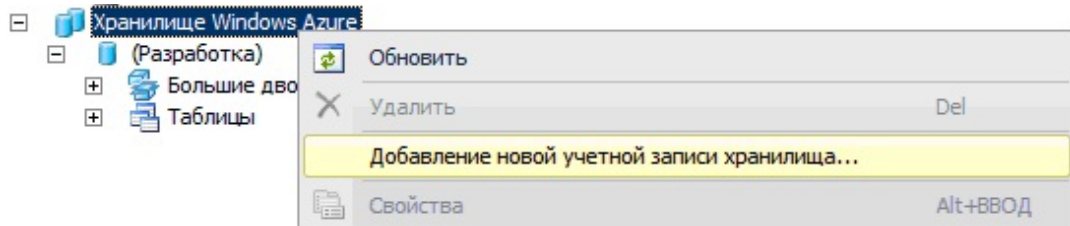
Запустив Обзорщик серверов (Меню "Вид" - Обзорщик серверов), увидим появившееся хранилище Windows Azure и хранилище "Разработка" - являющееся отображением эмулятора.



При помощи обзорщика серверов можно просматривать содержимое конкретных таблиц или контейнеров бинарных объектов.



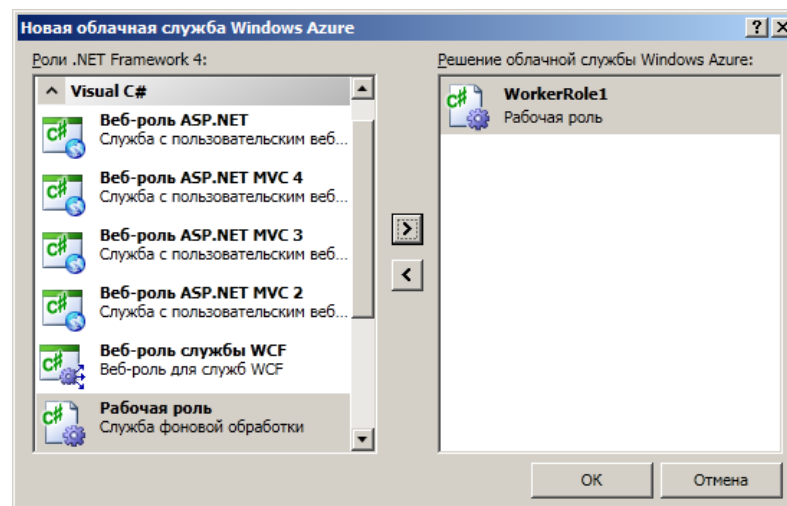
Также при помощи обозревателя серверов можно подключиться к хранилищу Windows Azure, добавив данные учетной записи.



Для того, чтобы продемонстрировать подключение к хранилищу данных, рассмотрим небольшой пример, по созданию хранилища с простой структурой - это будет таблица - список юридических лиц.

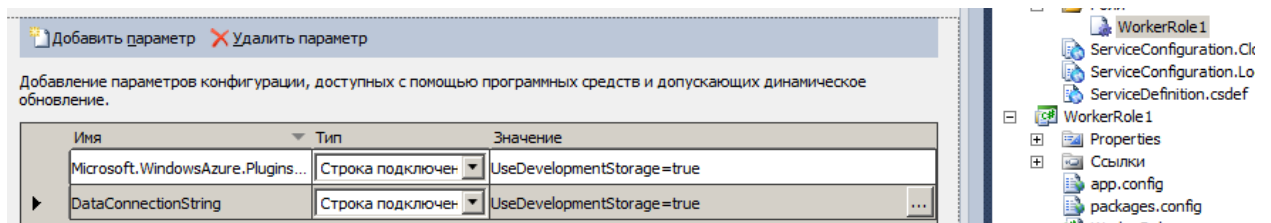
Создание хранилища с простой структурой данных.

1. Создадим проект облачной службы. SimpleDataStructure (Меню "Файл" - Создать - Проект). Добавим решению рабочую роль.



Наше приложение будет подключаться к эмулятору хранилища, создавать таблицу Students, если ее не существует и добавлять туда одну произвольную запись.

В свойствах рабочей роли определим строку подключения к эмулятору хранилища Azure: `UseDevelopmentStorage=True`



Далее, необходим класс, который будет описывать структуру сущности для нашей таблицы. Класс должен быть наследником к л а с с а - ш а б л о н а `Microsoft.WindowsAzure.StorageClient.TableServiceEntity`

```
class Student : TableServiceEntity
{
    public String FirstName { get; set; }
    public String LastName { get; set; }
    public Int32 Course { get; set; }
    public String Group { get; set; }
    public String Telephone { get; set; }
}
```

Для создания таблиц необходимо определить класс-контекст, при чем класс должен быть наследником `TableServiceContext`

Для каждой таблицы необходимо определить свойство типа `IQueryable`, где значение параметра `DataType` - тип сущностей, хранимых в таблице, в нашем случае - `Students`.

```
class StudentContext : TableServiceContext
{
    public IQueryable<Student> ContactData
    {
        get
        {
            return this.CreateQuery<Student>("Students");
        }
    }
    public StudentContext(Uri baseStudent, StorageCredentials
credentials) : base(baseStudent.AbsoluteUri, credentials) { }
}
```

Теперь обратим внимание на методы `Run` и `OnStart`. Первый содержит код выполняемый в ходе работы роли, второй - при ее запуске.

Для того, чтобы добавить данные в таблицу нам необходимо в методе `Run`:

- создать экземпляр класса - учетной записи
- создать экземпляр класса - контекста
- создать экземпляр класса - сущности и задать его параметры
- создать таблицу `Students`, если она не существует

- добавить сущность в таблицу

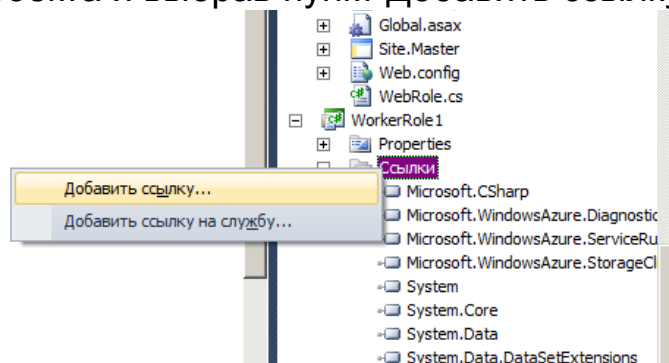
Для этого содержимое метода Run необходимо заменить следующим кодом:

```
CloudStorageAccount.SetConfigurationSettingPublisher(
    (configName, configSettingPublisher) =>
    {
        var connectionString =
            RoleEnvironment.GetConfigurationSettingValue(configName);
        configSettingPublisher(connectionString);
    }
);
//определение учетной записи
CloudStorageAccount account =
CloudStorageAccount.FromConfigurationSetting("DataConnectionString");

//создание таблицы Windows Azure Table
CloudTableClient _tc = null;
_tc = account.CreateCloudTableClient();
_tc.CreateTableIfNotExist("Students");
/*определение сущности, в том числе свойств ключ строки и ключ секции,
унаследованных от родительского TableServiceEntity*/
Student stud = new Student();
stud.PartitionKey = "Студент";
stud.RowKey = "40100173";
stud.FirstName = "Петр";
stud.LastName = "Иванов";
stud.Course = 2;
stud.Group = "ИИ-261";
stud.Telephone = "(913) 123-45-67";

//определение контекста
StudentContext context = new StudentContext(account.TableEndpoint,
account.Credentials);
//добавление сущности таблице Students
context.AddObject("Students", stud);
//сохранение изменений
context.SaveChanges();
```

В случае отсутствия ссылки `System.Data.Services.Client` необходимо ее добавить в проект `WorkerRole1`, вызвав контекстное меню на папке Ссылки данного проекта и выбрав пункт Добавить ссылку...

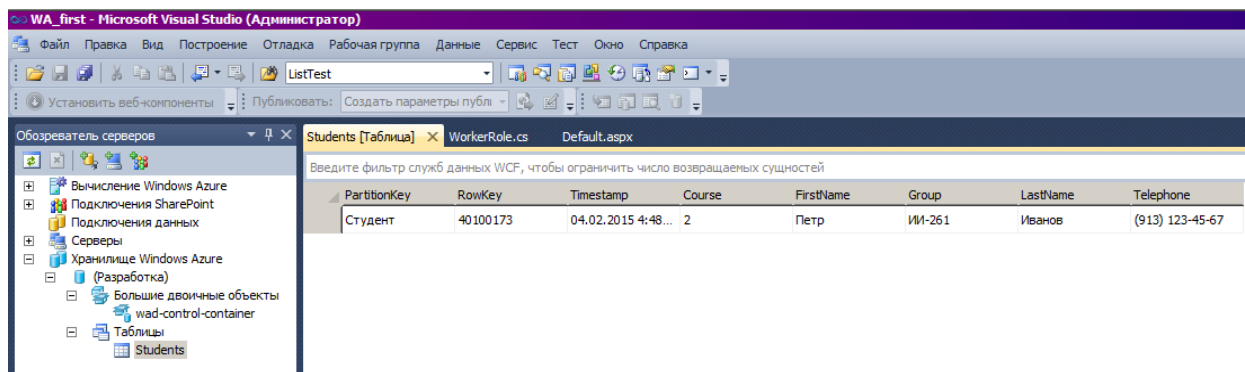


Далее на вкладке `.Net` выделить нужную ссылку и нажать кнопку `Ок`. Также необходимо подключить соответствующую библиотеку к проекту, добавив в верхней части модуля `WorkerRole1.cs` строку подключения библиотеки

```
using System.Data.Services.Client;
```

Запустите приложение, убедитесь в том, что оно выполнилось без ошибок и остановите его.

В диспетчере серверов, во вкладке "Хранилище Windows Azure" обновите вкладку "Таблицы", вы увидите созданную нашим приложением таблицу Students. Щелкните на таблице правой кнопкой мыши и выберите "Просмотреть данные". Вы увидите, что определенная нами сущность добавлена в таблицу.



Требования к содержанию отчета

титульный лист;

тема, цель, задание;

иллюстрации основных этапов настройки хранилища данных с краткими пояснениями;

листинг программы на языке C#;

результаты выполнения программы;

вывод — развернутая формулировка вывода из проделанной лабораторной работы.

Перечень полезных ссылок

Обзор Windows Azure Table:

1. <http://blogs.msdn.com/b/jnak/archive/2010/01/06/walkthrough-windows-azure-table-storage-nov-2009-and-later.aspx>

Windows Azure Table - подробный обзор:

1. <http://msdn.microsoft.com/ru-ru/library/ee872426.aspx>
2. <http://www.bandgap.cs.rice.edu/classes/comp410/resources/Using%20Azure/Using%20Azure%20Table%20Storage.aspx>

Руководства по устранению неполадок:

2. <http://msdn.microsoft.com/ru-ru/library/ee460770.aspx>

