

Курс: Разработка приложений с использованием WCF

ЗАДАНИЕ №1

Создать WCF службу под условным названием **DiskInfo** Контракт службы должен быть создан в интерфейсе и содержать метод

```
string[] GetDiskInfo(string path)
```

Клиент будет вызывать этот метод, передавая в строке *path* какой-нибудь путь. Метод должен будет вернуть клиенту в ответ имена файлов и папок, расположенных по пути *path* на сервере. При передаче клиентом неверного пути, служба должна возвращать *null*.

Внимание! Передать клиенту надо имена объектов, расположенных только в указанной папке. Рекурсию использовать не надо и вложенные папки читать не надо, надо только передать их имена.

WCF служба *DiskInfo* должна хоститься в консольном приложении. Конечная точка и привязка службы должны задаваться вызовом метода *AddServiceEndpoint()*. Файл конфигурации службы создавать не надо.

Клиент должен представлять собой тоже консольное приложение. Канал связи со службой надо создать с помощью явного описания интерфейса с контрактом службы и явного использования класса *ChannelFactory<>*.

Клиент должен выводить в консоль полученную от сервера информацию. Например, в таком виде:

```
Content for path: F:\STEP\C#  
Page1.cs      File  
Readme.txt    File  
Program.cs    File  
Docs          Directory  
Images        Directory
```

Проверьте, будет ли работать служба с привязкой `wsHttpBinding`? А с привязкой `netTcpBinding`? Чем отличается применение службы при использовании каждой из этих привязок?

Курс: Разработка приложений с использованием WCF

ЗАДАНИЕ №2

Надо модифицировать клиента и службу, которые созданы в первом домашнем задании. Уберите на стороне службы явный вызов метода **AddServiceEndpoint()**. Задайте адрес конечной точки и привязку службы в конфигурационном файле. Также опишите в конфигурационном файле *тех* точку.

Клиента переделайте в WinForms приложение. Уберите на стороне клиента явное использование класса **ChannelFactory<>** и явное описание интерфейса с контрактом службы. Поскольку у службы теперь есть *тех* точка, на стороне клиента можно создать прокси-класс. Прокси-класс создается командой «Добавить ссылку на службу».

Выводить полученный от службы список объектов в окне клиента, например в *ListBox*. Вспомните, какую функцию выполняет у WCF службы конечная точка с именем *тех*? Вспомните, для чего применяется системная утилита *svcutil.exe*?

Курс: Разработка приложений с использованием WCF

ЗАДАНИЕ №3

Создать WCF службу под условным названием Converter.

Контракт службы должен быть создан в интерфейсе и содержать методы:

- `ConvertedUnits LinearMeasure(double meters);`
- `ConvertedUnits CelsiusToFahrenheit(double c);`
- `ConvertedUnits FahrenheitToCelsius(double f);`

Первый метод конвертирует переданное значение в метрах в дюймы, футы и ярды, два других метода преобразуют температура из шкалы Цельсия и в шкалу Фаренгейта и обратно. Особенность этой службы – класс `ConvertedUnits`.

Он может иметь такой вид и является контейнером для пересылки конвертированных значений.

```
Class ConvertedUnits
{
    public double inch;
    public double foot;
    public double yard;
    public double Celsius;
    public double Fahrenheit;
}
```

Каждый из методов службы заполняет в этом классе только свои поля. Например, метод ***LinearMeasure()***, получив значение в метрах, переведет его в дюймы, футы и ярды и запишет в первые три свой-

ства. Температурные свойства этот метод не трогает. Аналогично, два других метода не трогают линейные свойства этого класса. Может сконструировать класс другим способом. Лишь бы он передавал все данные.

Задайте адрес конечной точки и привязку службы в конфигурационном файле. Также опишите в конфигурационном файле tech точку. Клиент должен быть WinForms или WPF приложением. На стороне клиента создайте прокси-класс. Прокси-класс создается командой «Добавить ссылку на службу». Создайте приятный интерфейс для ввода значений для конвертации и для вывода полученных от службы результатов в окне клиентского приложения.