

Лабораторна робота.

Створення простої служби windows communication foundation

Мета: отримати базові знання та навички роботи з windows communication foundation (WCF).

Основні відомості

Windows Communication Foundation (WCF) – програмний фреймворк, який використовується для обміну даними між додатками, що входять до складу .NET Framework. WCF робить можливим побудову безпечних і надійних транзакційних систем через спрощену уніфіковану програмну модель міжплатформної взаємодії. Або простіше: додатків для віддаленого міжплатформного обміну даними.

WCF реалізує таку базову модель комунікацій між клієнтським і серверним додатками:

- на стороні сервера вказується, які методи можуть бути викликані віддаленими клієнтськими додатками;
- на стороні клієнта вказуються сигнатури серверних методів, які повинні викликатися;
- на сторонах сервера та клієнта вибираються транспортний та комунікаційний протокол (в WCF це робиться через прив'язку);
- клієнт встановлює підключення до сервера;
- клієнт викликає віддалений метод, який прозоро виконується на сервері.

Є декілька шляхів створення служби WCF, можна, наприклад, обрати в якості початкової точки стандартний шаблон проекту Class Library (Бібліотека класів) і вручну додати посилання на збірки WCF.

Але більш простим і зрозумілим є альтернативний шлях: обрати у Visual Studio шаблон проекту WCF Service Library (Бібліотека служб WCF). Цей тип проекту автоматично встановлює посилання на необхідні збірки WCF, однак його недоліком є те, що при цьому генерується деякий об'єм початкового коду, який згодом доволі часто видаляється.

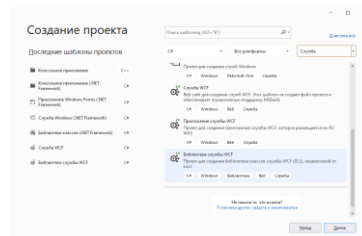
Однією з переваг вибору цього шаблону є створення файлу App.config. Корисність цього файлу у тому, що при налагодженні або запуску проекту WCF Service Library інтегроване середовище розробки Visual Studio автоматично завантажить додаток WCF Test Client (Тестовий клієнт WCF). Програма WcfTestClient.exe читає налаштування із файлу App.config, тому може використовуватись для тестування служби.

У Visual Studio доступний ще один шаблон проекту WCF Service, пов'язаний з WCF, який знаходиться в діалоговому вікні New WebSite. Даний шаблон зручний у випадку, якщо заздалегідь відомо, що служба WCF має використовувати протоколи на основі HTTP. На противагу цьому, служба створена із застосуванням шаблону WCF Service Library дає можливість розміщення служби декількома способами.

Приклад. Нехай наша WCF служба буде повертати кількість днів, що залишилися до нового року.

Для створення додатку необхідно виконати наступні дії:

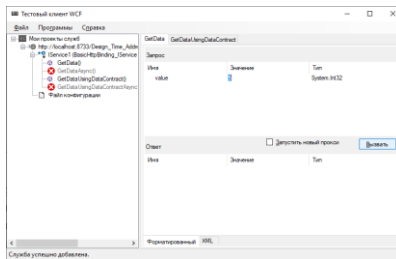
1. Створення служби WCF



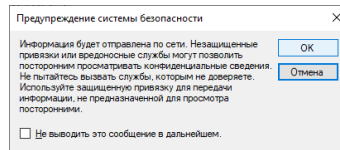
Вибрати директорію для збереження проекту та натиснути «Далее». Таким чином буде створена працююча служба, яку можна буде тестувати та використовувати.

2. Тестування служби WCF

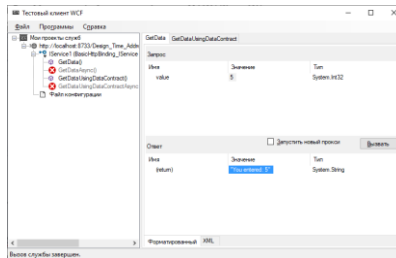
Натиснути F5 для запуску служби. На екрані з'явиться форма WCF Test Client та завантажить службу. Двічі натиснути метод GetData() під вузлом IService1. З'явиться вкладка GetData.



В області запиту ввести значення та натиснути кнопку «Вызвать» (Invoke)

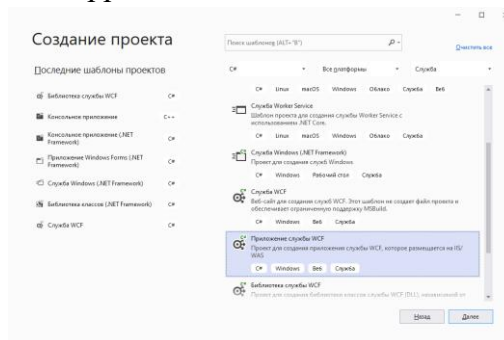


Результат буде виведено в області відповіді



3. Створення проекту сервісу.

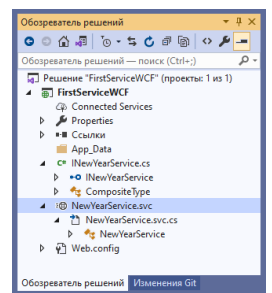
Створити проект типу WCF Service Application



Його структура буде такою:

IService1.cs містить опис інтерфейсу сервісу, тобто набір методів, які сервіс надає;

- **Service1.svc** складається з двох частин – реалізація сервісу (**Service1.svc.cs**) і розмітка (**Markup**);
- **Web.config** – конфігурація сервісу;
- папка **App_Data**.



Змінити назви файлів сервісу та його інтерфейсу, надавши їм імена

NewYearService та **INewYearService** відповідно. Перевірити наступний рядок у розмітці сервісу (файл **NewYearService.svc**).

```
<%@ ServiceHost Language="C#" Debug="true" Service="FirstServiceWCF.NewYearService" CodeBehind="NewYearService.svc.cs" %>
```

Розглянемо інтерфейс **INewYearService**. Для початку необхідно в тілі інтерфейсу оголосити метод, який буде надавати служба для виклику. Для цього його необхідно позначити атрибутом **[OperationContract]**.

```
[ServiceContract]
public interface INewYearService
{
    [OperationContract]
    TimeToNewYear GetDaysToNewYear(DateTime start);
}
```

!!! Всі методи інтерфейсу повинні бути позначені атрибутом [OperationContract], інакше вони не будуть видимі клієнтам.

Метод повертає екземпляр класу TimeToNewYear – допоміжний клас, що містить значення часу до нового року. Для того, щоб даний клас можна було використовувати в якості аргументу що повертається, його необхідно позначити атрибутом [DataContract], а властивості, доступні для читання клієнту в повернутому значенні позначаються атрибутом [DataMember].

```
[DataContract]
public class TimeToNewYear
{
    private DateTime start;           // момент часу, від якого відраховується час
    private DateTime newYear;        // дата нового року
    [DataMember]
    public int Days { get; private set; } // кількість днів до нового року
    [DataMember]
    public int Hours { get; private set; } // кількість годин до нового року
    [DataMember]
    public int Minutes { get; private set; } // кількість хвилин до нового року
    [DataMember]
    public int Seconds { get; private set; } // кількість секунд до нового року
    // створити новий екземпляр відліку часу до нового року (момент часу, від якого ведеться відлік)
    public TimeToNewYear(DateTime point)
    {
        start = point;
        newYear = DateTime.Parse($"{{point.Year}}-12-31 23:59:59.999");
        SetPeriods();
    }
    private void SetPeriods()         // метод, що встановлює тимчасові проміжки до нового року
    {
        var interval = newYear - start;
        Days = (int)interval.TotalDays;
        Hours = (int)interval.TotalHours;
        Minutes = (int)interval.TotalMinutes;
        Seconds = (int)interval.TotalSeconds;
    }
}
```

Тепер потрібно реалізувати інтерфейс служби в класі NewYearService.svc.cs наступним чином:

```
public TimeToNewYear GetDaysToNewYear(DateTime start)
{
    var timeToNewYear = new TimeToNewYear(start);
    return timeToNewYear;
}
```

4. Розміщення сервісу

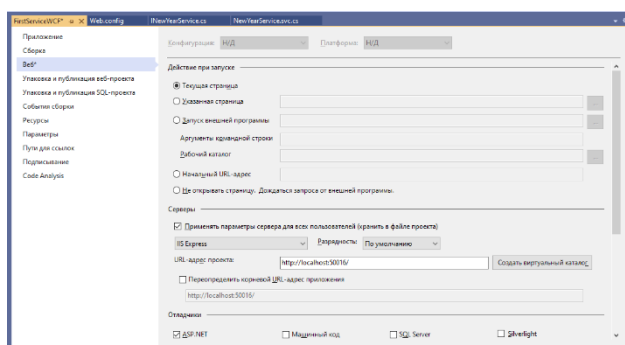
Сам по собі сервіс являє собою бібліотеку та для його запуску у вигляді сервісу необхідно використовувати один із методів WCF:

- хостинг на IIS;
- запуск у вигляді служби Windows;
- Self hosting (сервіс виконаний у вигляді консольного додатку, що запускає сервіс).

Принципової різниці між цими методами немає. Найпростіший варіант – розміщення на IIS. Для цього потрібно відкрити конфігурацію проекту («Свойства») та обрати вкладку «Веб» (Web).

У всіх випадках сервіс завантажується в режимі налагодження та поводить себе як звичайний додаток Visual Studio. Останній варіант зручний тим, що нічого не відкривається у браузері.

Розділ «Сервери» (Servers) визначає, де буде розміщений сервіс. За замовчуванням обраний IIS Express (вбудований у Visual Studio варіант IIS). Але можна розмістити сервіс і на звичайному IIS, для цього треба його встановити та увімкнути необхідні компоненти.

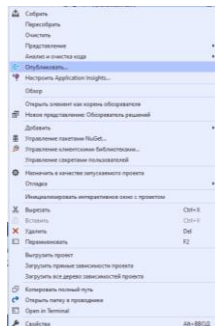


Вказати адресу сервісу в налаштуваннях та натиснути «Создать виртуальный каталог» (Create Virtual

Перевірити роботу додатку, відкривши в браузері рядок, вказаний у конфігурації проекту та додавши в кінці назву сервісу.

Під публікацією сервісу мається на увазі його компіляція для подальшого викладення збірки на сервер.

Створити профіль публікації: натиснути правою кнопкою миші на імені проекту та обрати пункт «Опублікувати...» (Publish...).



Обрати Custom, ввести назву профіля. Вибрати метод публікації File System та вказати шлях, за яким буде збережена збірка, краще створити для неї нову папку.

Для розміщення сервісу на сайті, необхідно скопіювати папку з додатком на сервер та підключити до IIS як додаток, таким чином, якби це був звичайний сайт (Default Web Site => Add Application...).

Використовувати створений додаток можна різними способами: виклик через клієнт на C#, звернення через SoapUI.

Створити у наявному розв'язку ще один проект типу Console Application та назвати його TestClient. Додати посилання на сервіс (Add Service Reference).

У вікні, що відкрилося, ввести локальну адресу сервісу. Також в цьому вікні вказується назва простору імен та додаткові налаштування, видимі методи, що використовує створений сервіс.

Після створення посилання на сервіс, в проєкті має з'явитися папка **Service References**, в якій буде знаходитись згенерований Visual Studio клієнт.

Написати код для підключення до сервісу та виклику арифметичних операцій, що ним надаються.

SoapUI – безкоштовна програма для тестування SOAP-сервісів. Поміж іншого, вона надає можливість перегляду внутрішньої структури повідомлень, що може бути корисним при вирішенні складних проблем, у випадку коли клієнт і сервіс використовують різні платформи.

Завантажити програму та створити новий проект. В полі адреси сервісу вказати WSDL та ввести назву проекту.

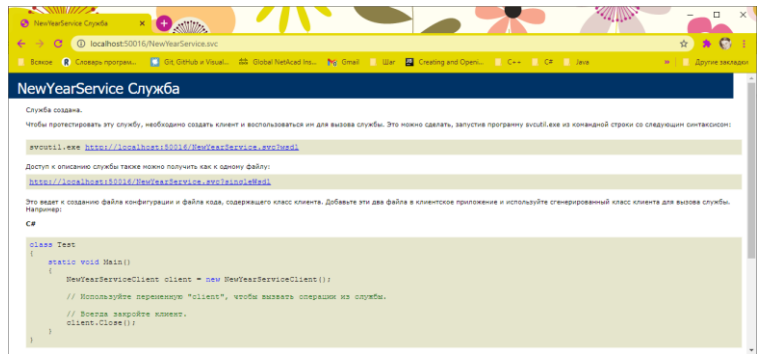
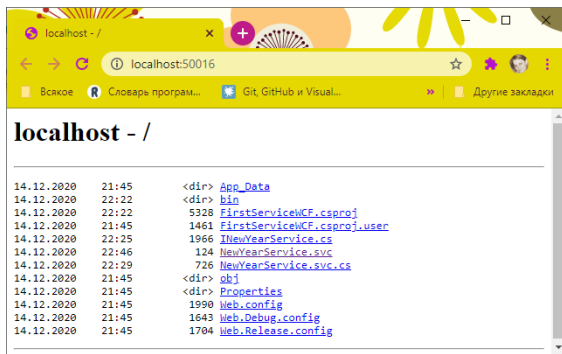
Програма відобразить основні інтерфейси сервісу та створить тестові запити для кожного з методів.

Замінити простір імен в атрибуті сервісу ServiceContract на бажаний. Оновити конфігурацію сервісу в тестовому клієнті (RemoteService => Update Service Reference), оновити опис в SoapUI (Update Definition), натиснувши F5 на імені точки підключення BasicHttpBinding ICalculator та оновити запит.

Повернемося до прикладу. Перевіримо роботу створеної служби WCF. Для цього запусимо налагодження.

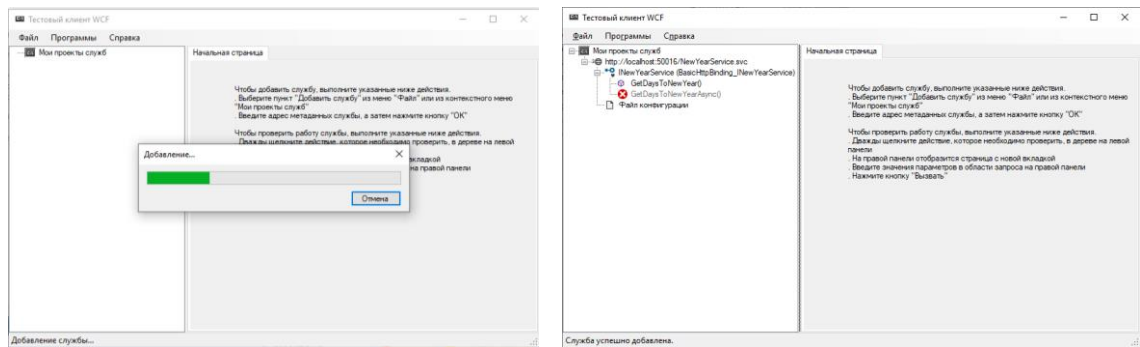
!!! Можливі два варіанти поведінки системи. Якщо ми почнемо налагодження перебуваючи в [NewYearService.svc](#), відкриється відладчик служби. У всіх інших випадках відкриється вікно браузера. Давайте розглянемо кожен з варіантів докладніше.

Браузер. Після запуску налагодження відобразиться браузер з файлової структури нашої служби WCF.

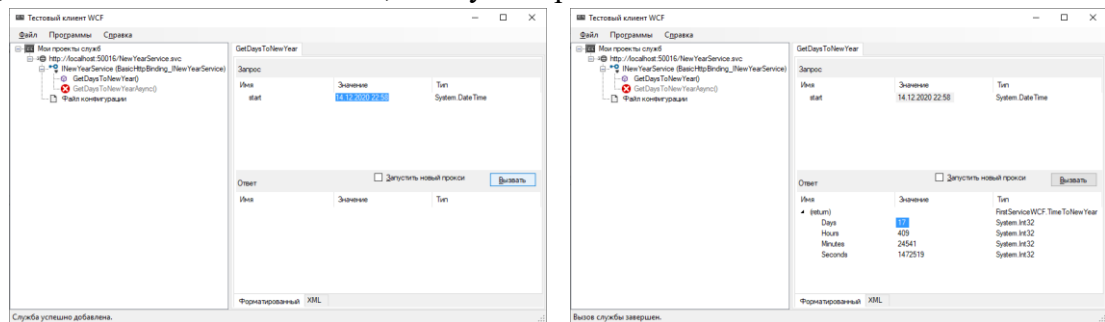


Необхідно натиснути на посилання з ім'ям служби **NewYearService.svc**. Якщо все працює коректно, то ми побачимо наступне вікно, інакше буде показано повідомлення з помилкою.

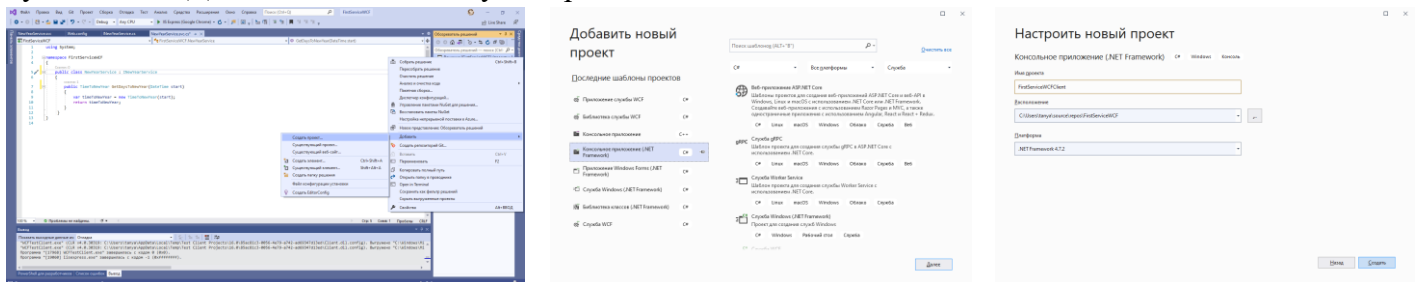
Тестовий клієнт WCF.



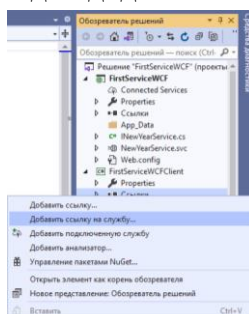
У лівій верхній частині відладчика можна побачити структура служби WCF. Для перевірки методу потрібно двічі клацнути лівою кнопкою миші по його імені. У правій частині відладчика відкриється форма запиту, де можна вказати значення, яке буде передано в метод.



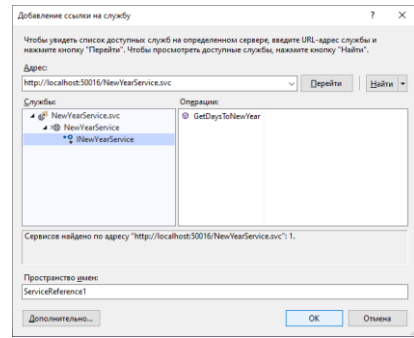
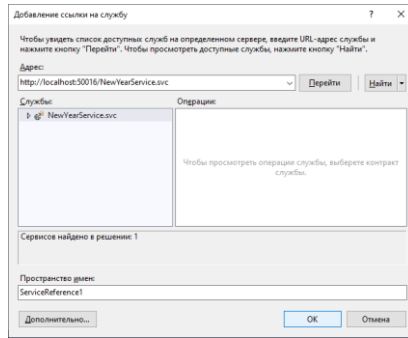
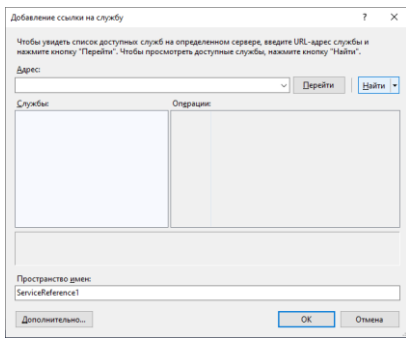
Консольний клієнт для WCF. Тепер необхідно створити клієнта, який буде звертатися до нашої служби WCF. Для цього для початку створимо нове консольний додаток.



У створеному консольному додатку необхідно додати посилання на службу WCF.



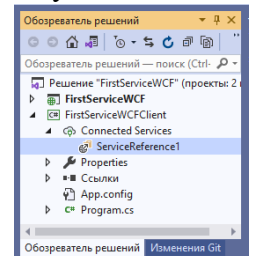
У вікні служби необхідно вказати ім'я служби WCF і ввести її адресу.



!!! Для простоти можна натиснути кнопку «Найти», тоді адреса служби WCF буде визначено автоматично.

Після цього необхідно розгорнути дерево WCF служби, щоб упевнитися у правильності вибору. У правій частині повинен бути відображений викликається метод.

Якщо налаштування пройшло коректно, то в браузері рішення в консольному додатку відобразиться посилання на нашу службу WCF.



Залишається тільки звернутися до нашої служби, щоб викликати метод і вивести результат на екран

`static void Main(string[] args)`

```
{ var client = new NewYearServiceClient(); // створимо клієнт для звернення до служби
  var result = client.GetDaysToNewYear(DateTime.Now); // викличемо метод служби і збережемо результат
  // виводимо результат на консоль
  WriteLine($"До нового року залишилося {result.Days} днів, або {result.Hours} годин,
  або {result.Minutes} хвилин, або {result.Seconds} секунд.");
  ReadKey();
}
```

!!! Перед початком налагодження встановіть автозавантаження консольного додатку.

Аналогічним чином організовується виведення результатів і в інших типах додатків (Windows, ASP тощо).

Завдання

Створити просту службу WCF. Перевірити коректність підключення та роботи служби.

Варіанти завдань (орієнтовний перелік)

1. Калькулятор (арифметичні операції, піднесення до степені (у тому числі корінь), тригонометричні функції).
2. Переведення радіанів в градуси та навпаки.
3. Конвертер валют.
4. Створити логи, які демонструють роботу з клієнтськими сесіями.

Контрольні питання

1. Яке призначення служби WCF?
2. Яким чином відбувається взаємодія компонентів через службу WCF?
3. Чим WCF відрізняється від .Net Remoting?
4. Які переваги та недоліки використання WCF?