**Paweł Frankowski U-19537**

**Programowanie w .NET.**

**Sprawozdanie z ćwiczenia praktycznego №6**

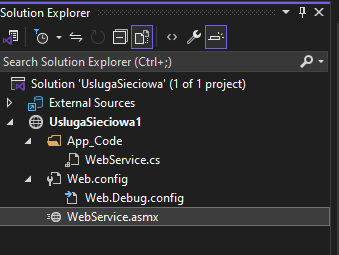
**Cel pracy:** Utworzenie i wykorzystanie rozproszonych systemów dystrybuowania danymi na platformie .NET, z wykorzystaniem WebService oraz WCF. Użycie elementów języka C# dostępnych od wersji .NET Framework 3.5.

1. **Zapoznać się z następnymi podstawami:**

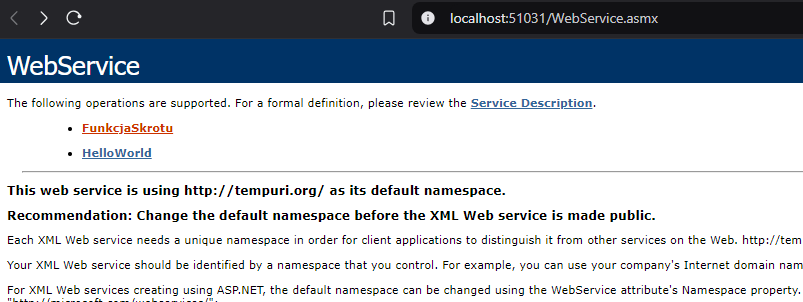
* Ogólne założenia systemów rozproszonych
  + System rozproszony to zbiór programów komputerowych, które wykorzystują zasoby obliczeniowe w wielu oddzielnych węzłach obliczeniowych, aby osiągnąć wspólny cel. Systemy rozproszone mają za zadanie eliminować z systemu wąskie gardła lub centralne punkty awarii.
* Udostępnianie usług i danych za pomocą WebService
  + WebService, czyli usługi sieciowe, to mechanizm pozwalający na wymianę danych między aplikacjami działającymi na różnych platformach i systemach. Opiera się najczęściej na protokole SOAP oraz formacie XML, dzięki czemu jest niezależny od technologii i umożliwia komunikację między różnymi językami programowania. Każda usługa jest opisana przez WSDL (Web Services Description Language), co umożliwia łatwe zrozumienie jej przez systemy zewnętrzne. WebService jest często wykorzystywany do komunikacji między mikroserwisami, aplikacjami mobilnymi i webowymi.
* Technologie systemów rozproszonych DCOM, .NET Remoting, WCF
  + DCOM (Distributed Component Object Model): To technologia opracowana przez Microsoft, umożliwiająca komunikację między aplikacjami na różnych komputerach, głównie w obrębie sieci lokalnych (LAN). DCOM opiera się na COM i korzysta z modelu wywoływania metod zdalnych.
  + .NET Remoting: Technologia Microsoftu stworzona z myślą o systemach rozproszonych na platformie .NET, umożliwiająca komunikację między aplikacjami, ale wymaga obu stron w ramach .NET Framework. W praktyce Remoting stał się przestarzały, szczególnie po wprowadzeniu WCF.
  + WCF (Windows Communication Foundation): Jest następcą .NET Remoting i DCOM, jednoczącym różne standardy komunikacji (SOAP, REST, TCP, MSMQ). WCF wspiera wiele sposobów komunikacji i umożliwia łatwiejszą konfigurację oraz dostosowanie usług sieciowych pod różne potrzeby (bezpieczeństwo, szybkość). WCF jest bardzo wszechstronny i jest stosowany w aplikacjach o wysokich wymaganiach dotyczących wydajności i bezpieczeństwa.
* Komunikacja międzyplatformowa
  + Komunikacja międzyplatformowa to zdolność różnych systemów operacyjnych i środowisk programistycznych do wymiany danych i interakcji ze sobą. Dzięki stosowaniu standardowych protokołów (jak HTTP) oraz formatów (XML, JSON), systemy zbudowane w różnych językach i na różnych platformach mogą wymieniać dane. W kontekście .NET, WCF oraz WebService umożliwiają tworzenie usług kompatybilnych z wieloma technologiami, dzięki czemu systemy działające w sieci są niezależne od technologii wykorzystywanej po stronie klienta.
* Protokoły transmisji danych
  + HTTP(S): Najpopularniejszy protokół komunikacji między aplikacjami w Internecie, wykorzystywany przez większość WebService i WCF.
  + TCP: Protokół niskopoziomowy zapewniający niezawodność przesyłania danych między komputerami.
  + SOAP i REST: Protokoły stosowane w usługach sieciowych. SOAP jest bardziej strukturalny i działa głównie z XML, podczas gdy REST jest bardziej elastyczny i korzysta z formatu JSON, co czyni go lżejszym.

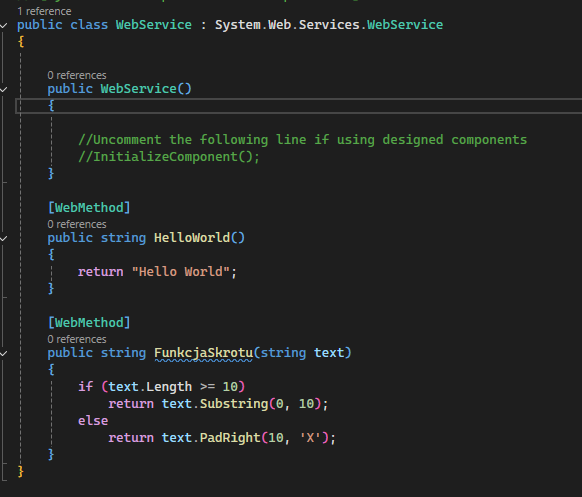
2. **Aplikacja**

Utworzono nową solucję WebService „UsługaSieciowa” na bazie Framework .NET 3.5.

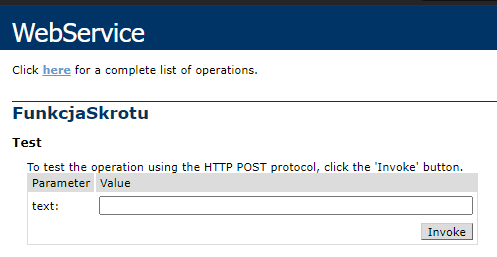


Rys. 1. Utworzenie solucji

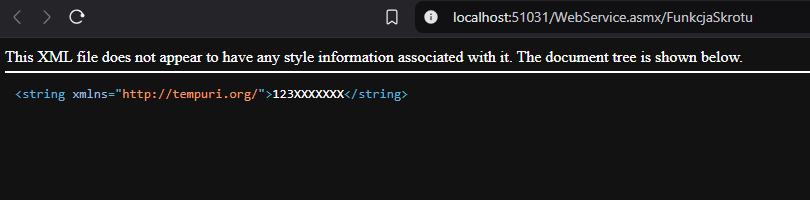
Rys. 2. Wyświetlenie dostępnych metod

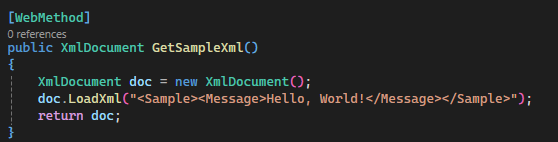


Rys. 3. Dodanie metody FunkcjaSkrotu do klasy WebService.

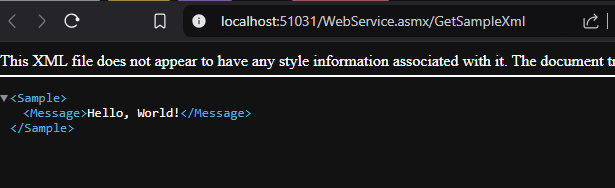


Rys. 4. Widok metody FunkcjaSkrotu

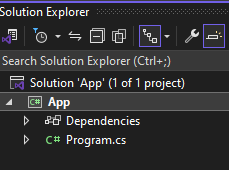
Rys. 5. Wynik metody FunkcjaSkrotu



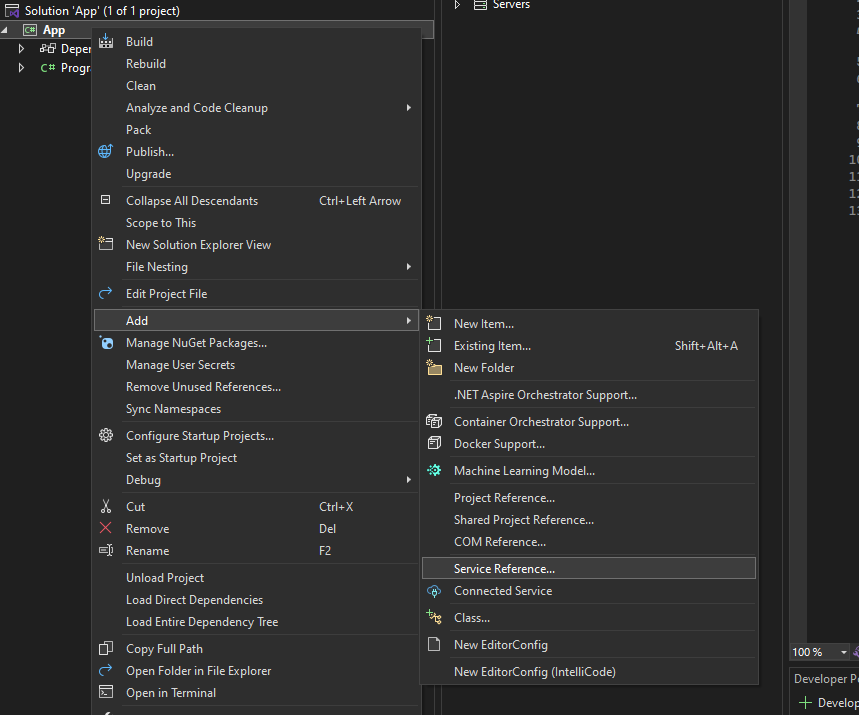
Rys. 6. Dodanie metody zwracającej obiekt XmlDocument

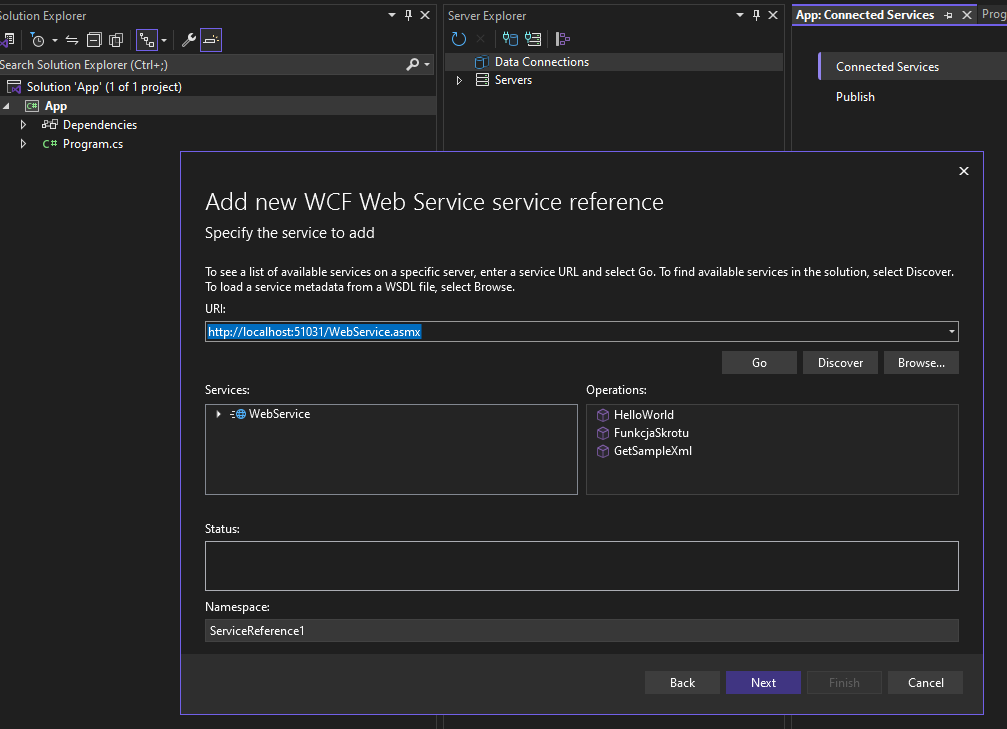


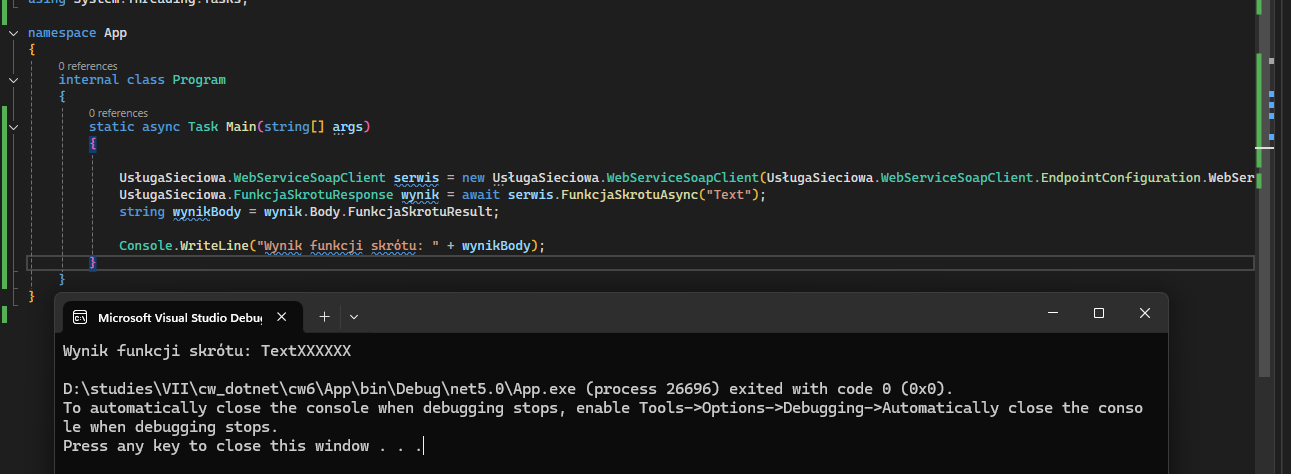
Rys. 7. Wynik metody GetSampleXml

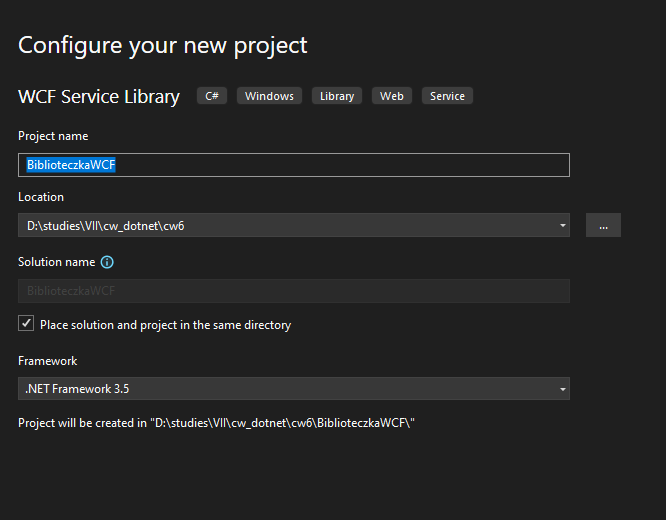


Rys. 8. Utworzenie nowej solucji

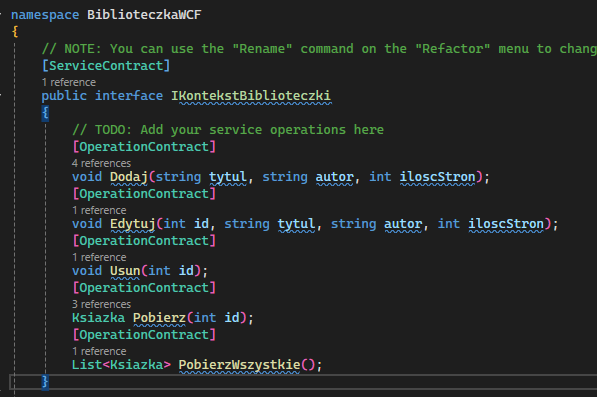
Rys. 9. Dodawanie referencji do projektu

Rys. 10. Wskazanie adresu serwisu

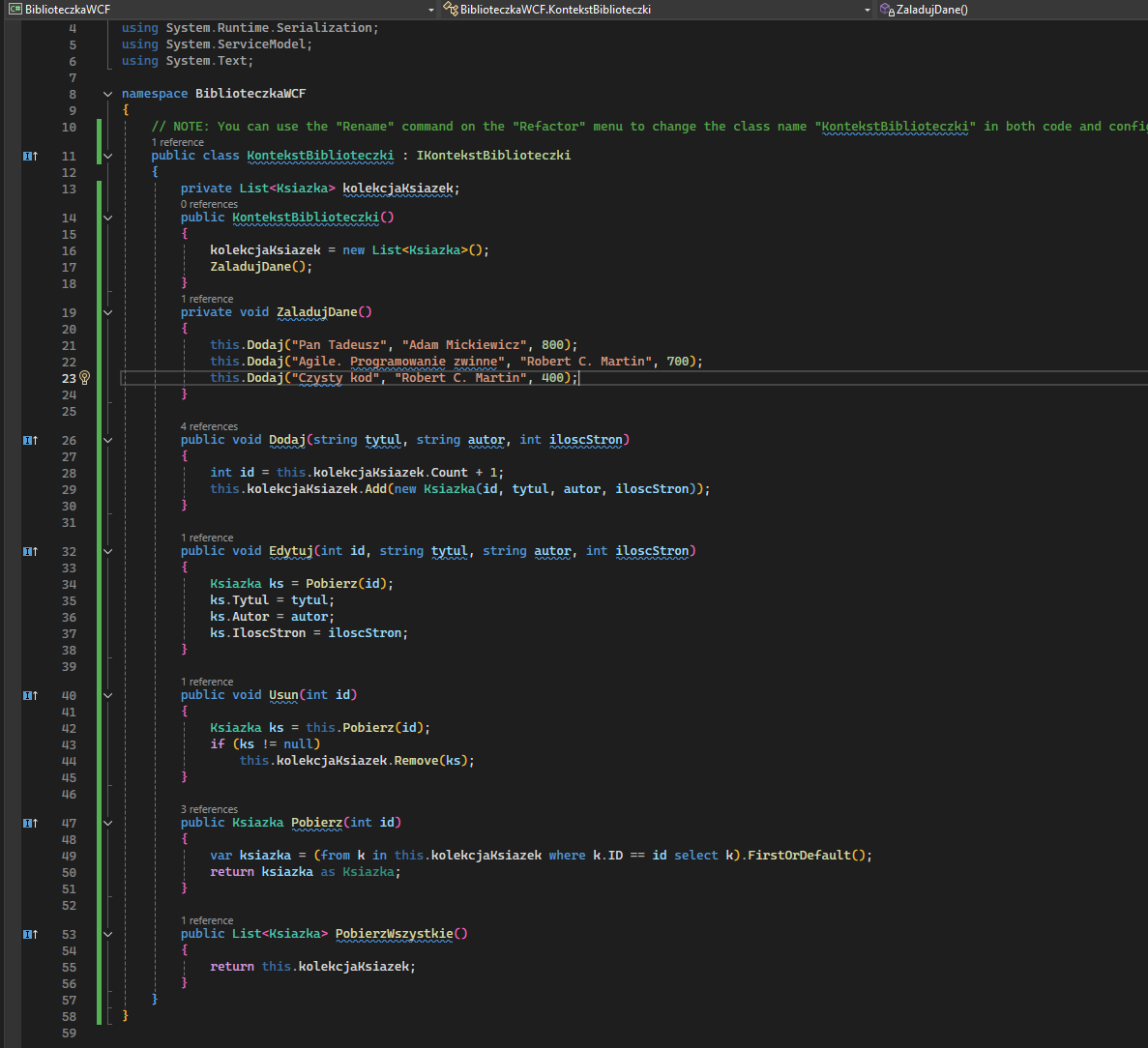
Rys. 11. Wywołanie metody z serwisu w aplikacji klienckiej

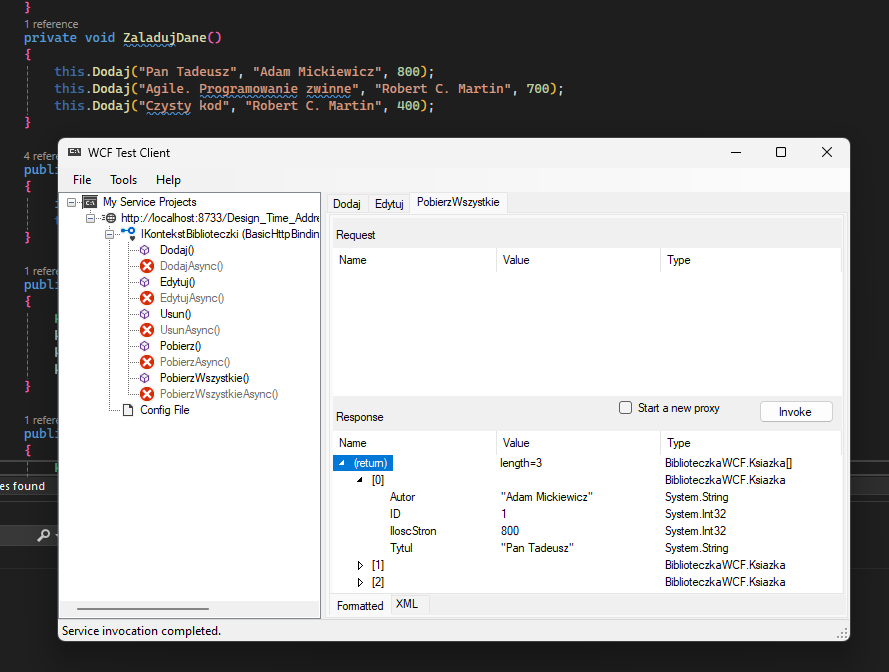


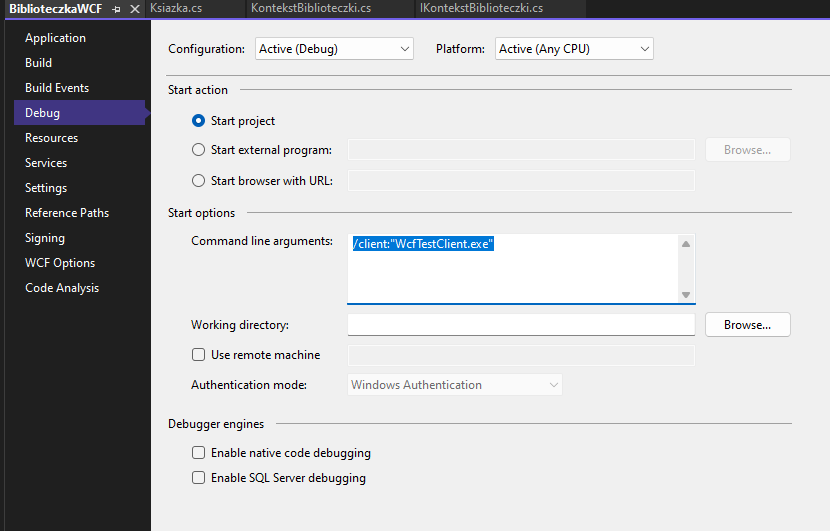
Rys. 12. Utworzenie solucji BiblioteczkaWCF na podstawie szablonu WCF Service Library

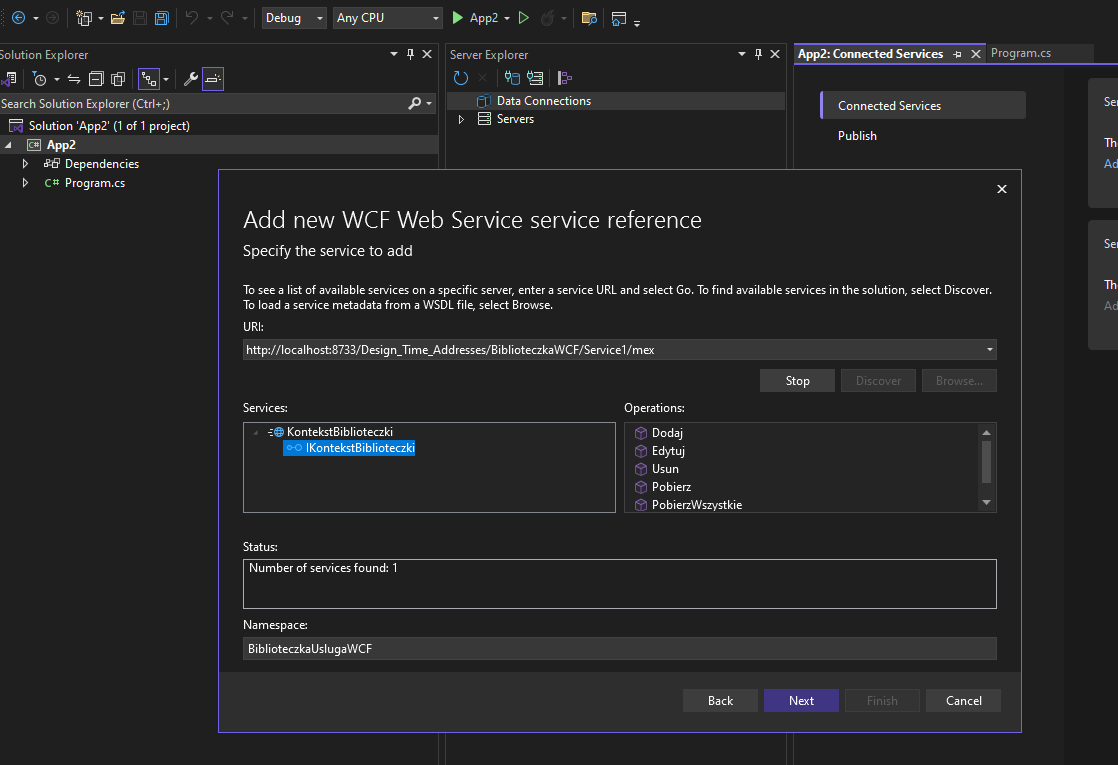


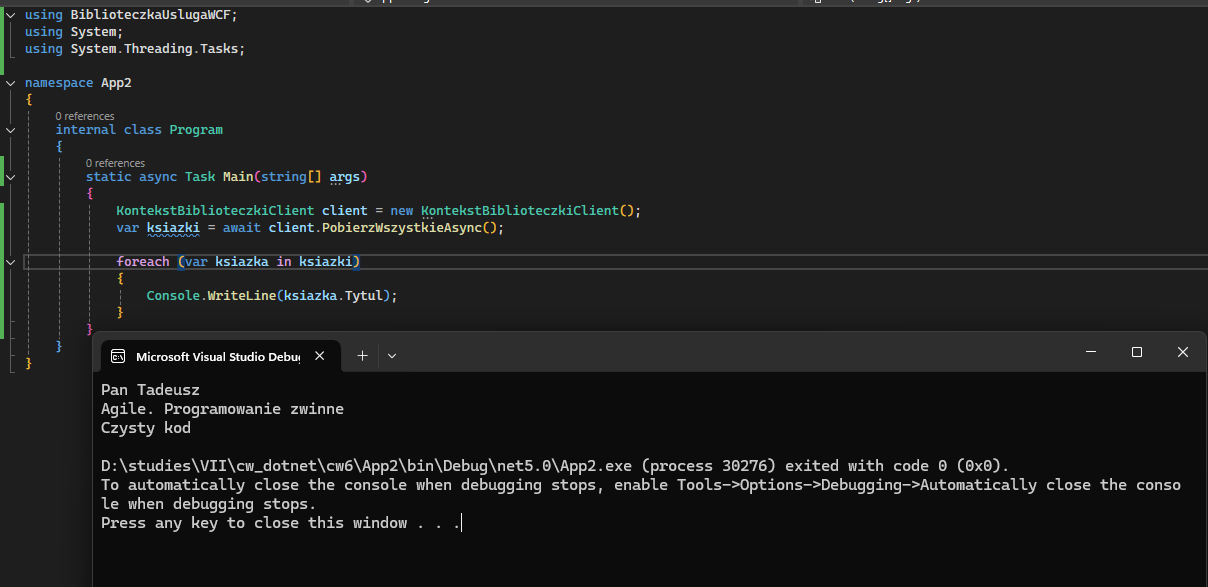
Rys. 13. Zdefiniowanie metod w interfejsie IkontekstBiblioteczki

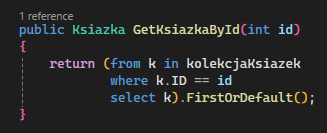
Rys. 14. Kod metod zdefiniowanych w interfejsie IKontekstBiblioteczki

Rys. 15. Testowanie metod za pomocą WCF Test Client

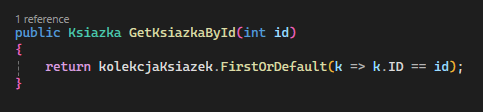
Rys. 16. Wykasowanie treści z pola Command Line Arguments

Rys. 17. Stworzenie nowej solucji oraz dodanie referencji do serwisu

Rys. 18. Wywołanie metody PobierzWszystkieAsync i wyświetlenie jej w projekcie klienckim



Rys. 19. Rozbudowanie aplikacji o metodę wykorzystującą zapytania LINQ



Rys. 20. Alternatywna forma metody GetKsiazkaById wykorzystująca wyrażenia lambda

3. **Podsumowanie**

W ramach ćwiczenia praktycznego nr 6, celem było stworzenie oraz wykorzystanie rozproszonych systemów dystrybuowania danymi na platformie .NET, z zastosowaniem technologii WebService oraz WCF. Na początku omówiono kluczowe koncepcje dotyczące systemów rozproszonych, w tym definicję, działanie oraz zasady komunikacji międzyplatformowej. Przeanalizowano również różnice pomiędzy DCOM, .NET Remoting a WCF, zwracając uwagę na zalety tej ostatniej technologii, która łączy różne standardy komunikacji oraz zapewnia lepszą wydajność i bezpieczeństwo.  
  
W praktycznej części ćwiczenia utworzono nową solucję WebService „UsługaSieciowa” na bazie .NET Framework 3.5, w której zdefiniowano metody do obliczeń oraz zwracania obiektów XML. Po dodaniu referencji do serwisu, przeprowadzono testy metod za pomocą aplikacji klienckiej, co pozwoliło na weryfikację poprawności działania stworzonych usług.  
  
Dodatkowo, w ramach drugiej części ćwiczenia, zbudowano solucję WCF „BiblioteczkaWCF”, definiując interfejs i metody, a następnie przetestowano je za pomocą WCF Test Client. Wykorzystano także zapytania LINQ oraz wyrażenia lambda, co umożliwiło bardziej eleganckie i zwięzłe przetwarzanie danych.  
  
Całość ćwiczenia pozwoliła na zdobycie praktycznej wiedzy z zakresu tworzenia i obsługi rozproszonych systemów na platformie .NET oraz zrozumienie zasady działania WebService i WCF. Wyniki oraz kod źródłowy zostały opublikowane w repozytorium na GitHubie.

**Link do repozytorium:** https://github.com/pablolambo/WebServices