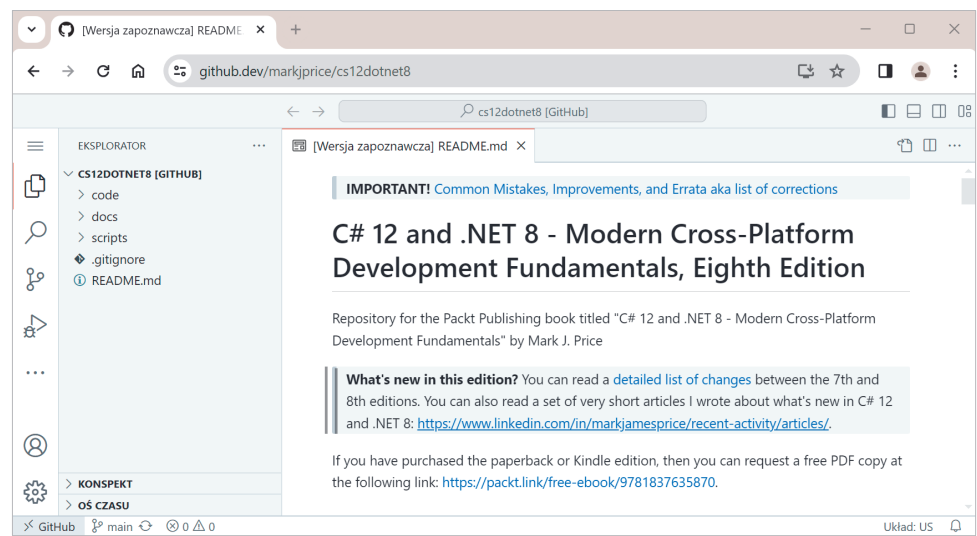
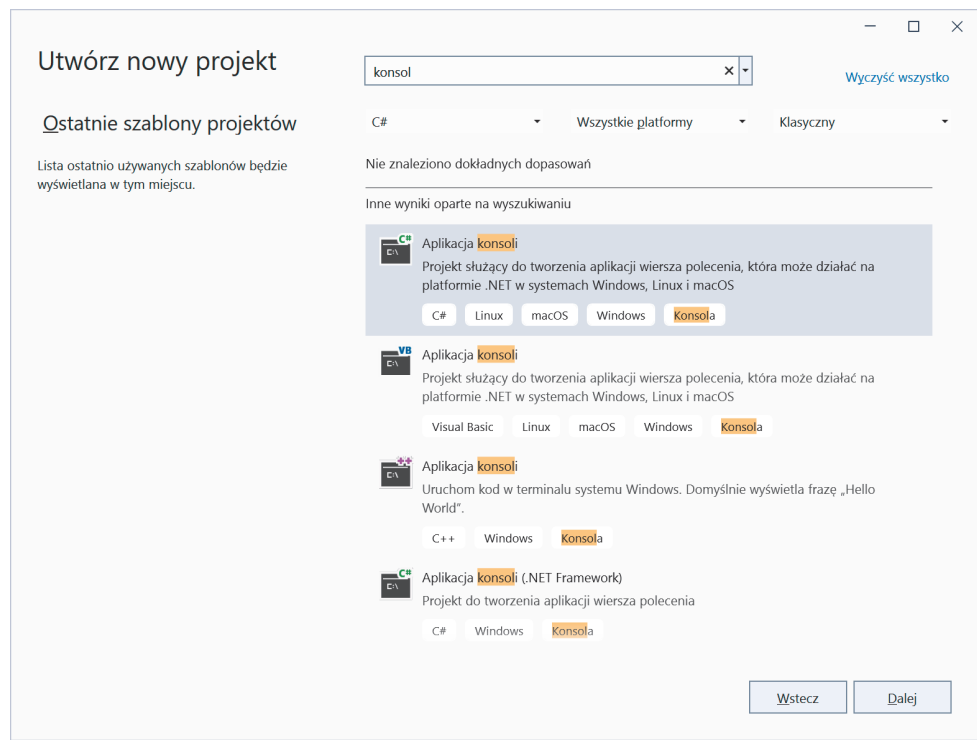


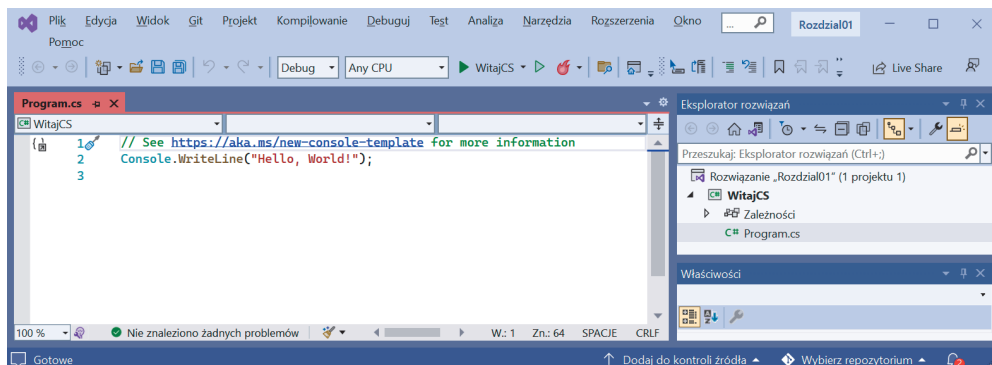
# Rozdział 1. Cześć, C#! Witaj, .NET!



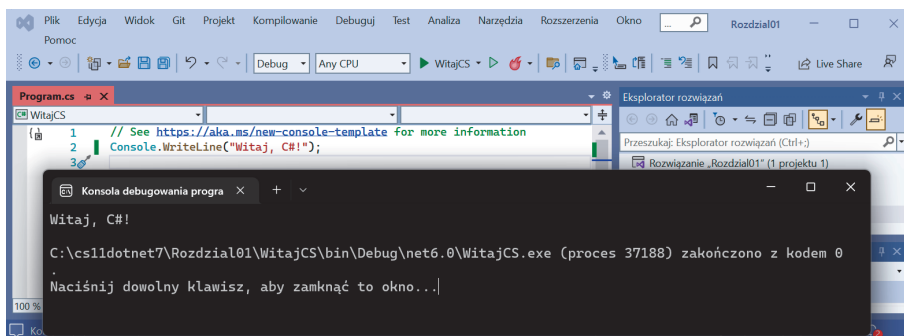
Rysunek 1.1. Edytor Visual Studio Code for Web z repozytorium GitHuba dla tej książki



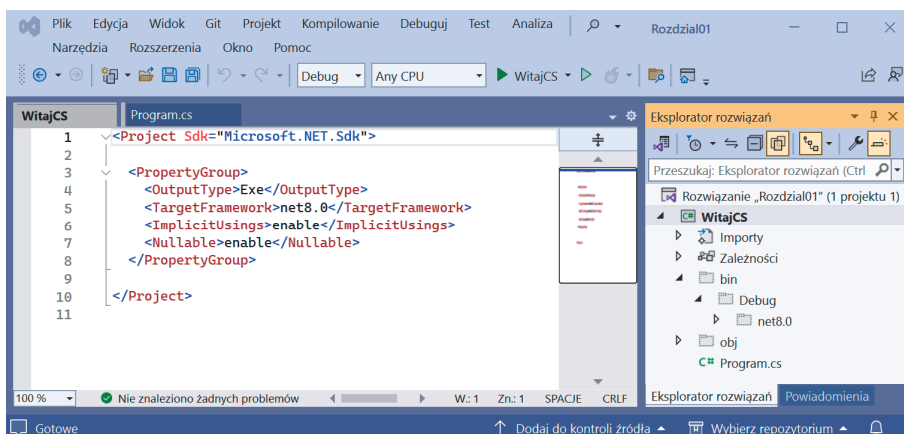
Rysunek 1.3. Wybieranie szablonu projektu aplikacji konsoli



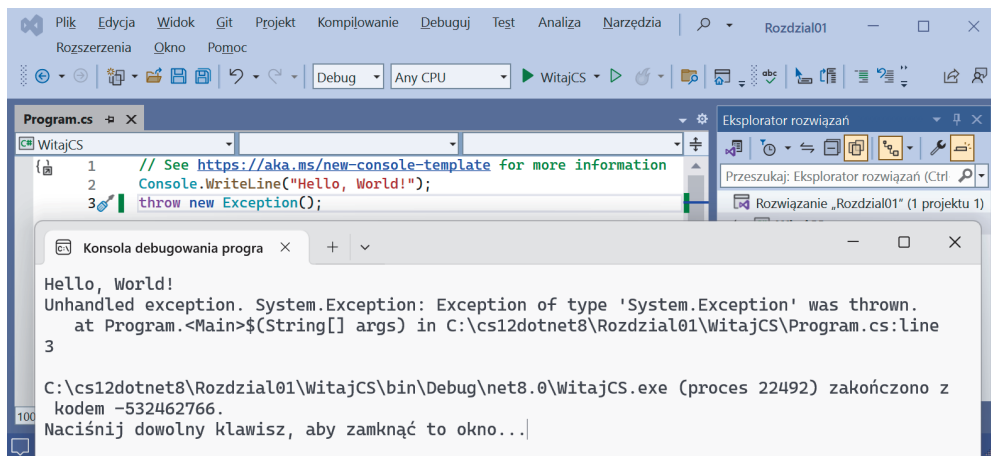
Rysunek 1.4. Edytowanie pliku Program.cs w Visual Studio 2022



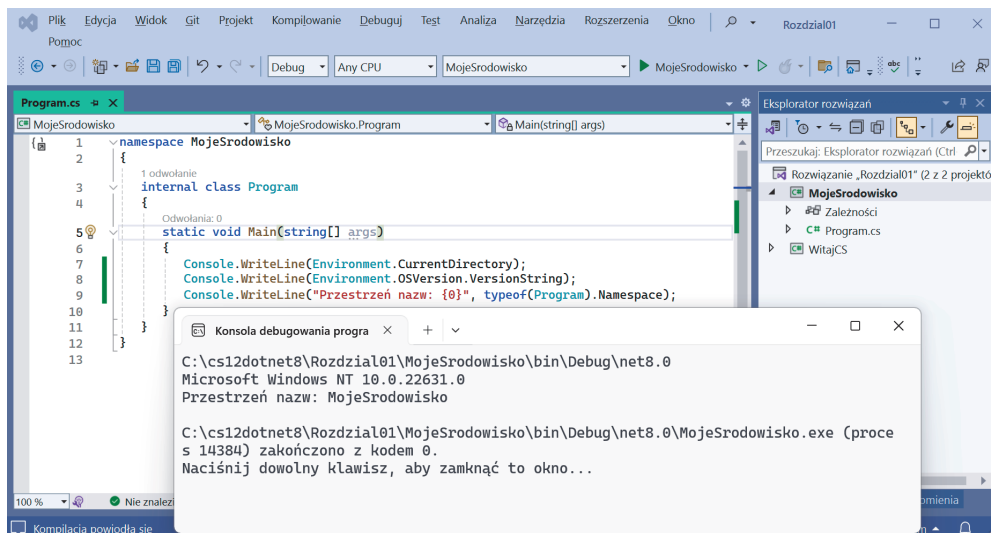
Rysunek 1.5. Uruchomienie aplikacji konsoli w systemie Windows



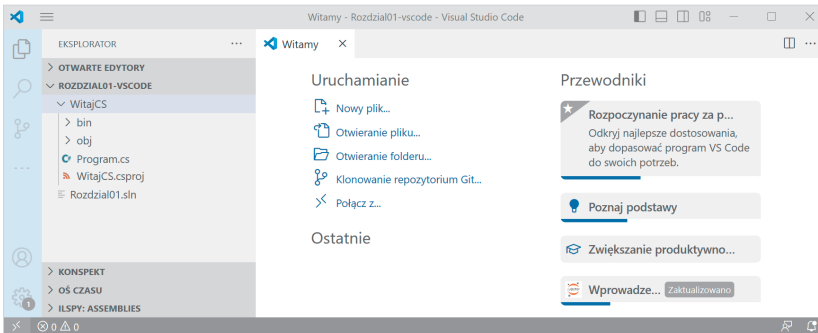
Rysunek 1.6. Wyświetlanie folderów i plików generowanych przez kompilator



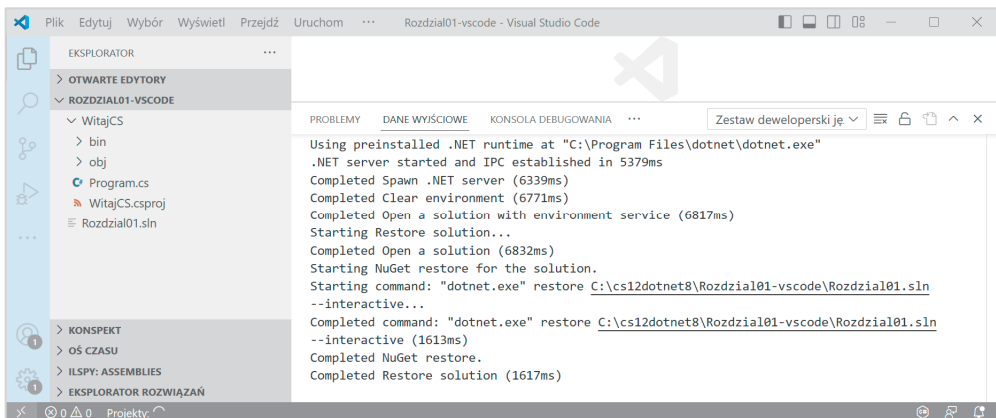
Rysunek 1.7. Rzucenie wyjątku ujawnia ukrytą metodę Program.<Main>\$



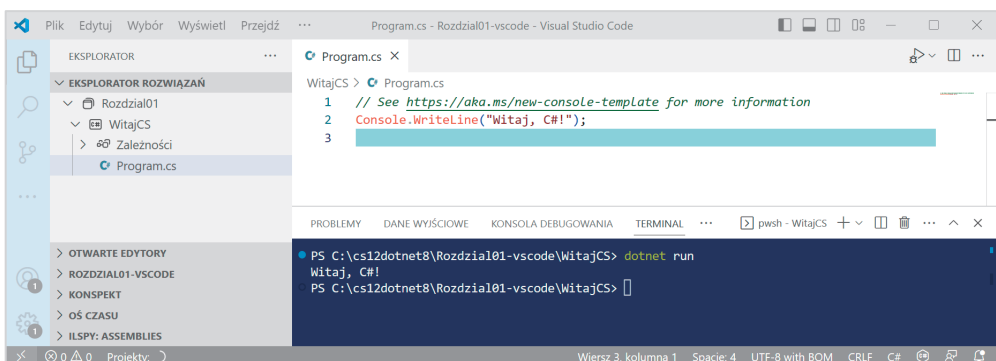
Rysunek 1.8. Uruchomienie programu najwyższego poziomu w rozwiązaniu składającym się z dwóch projektów



Rysunek 1.9. W panelu Eksplorator witać, że utworzono już dwa pliki i jeden folder

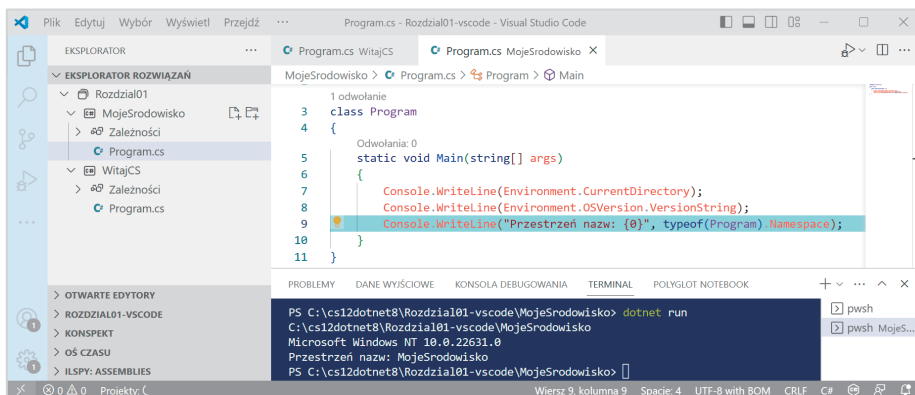


Rysunek 1.10. Rozszerzenie C# Dev Kit analizuje plik rozwiązania

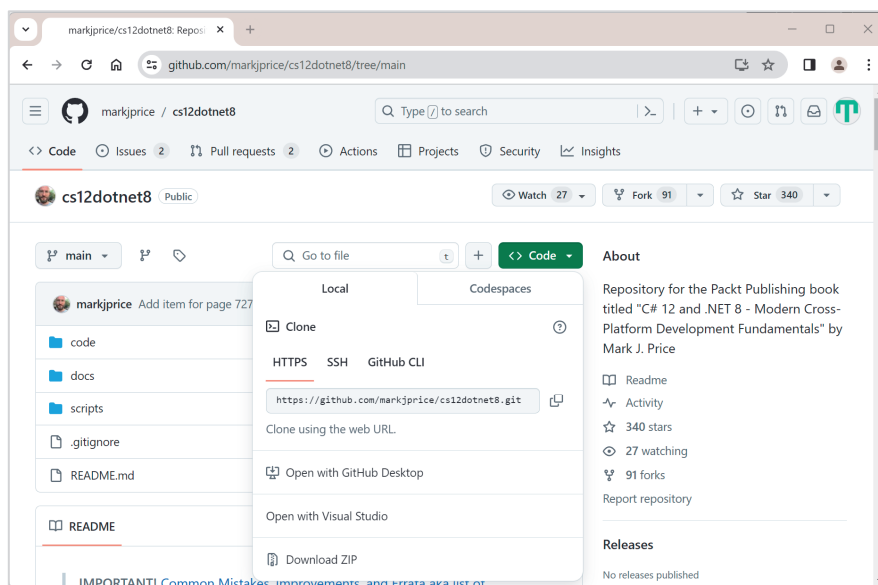


Rysunek 1.11. Wynik uruchomienia Twojej pierwszej aplikacji konsoli w Visual Studio Code

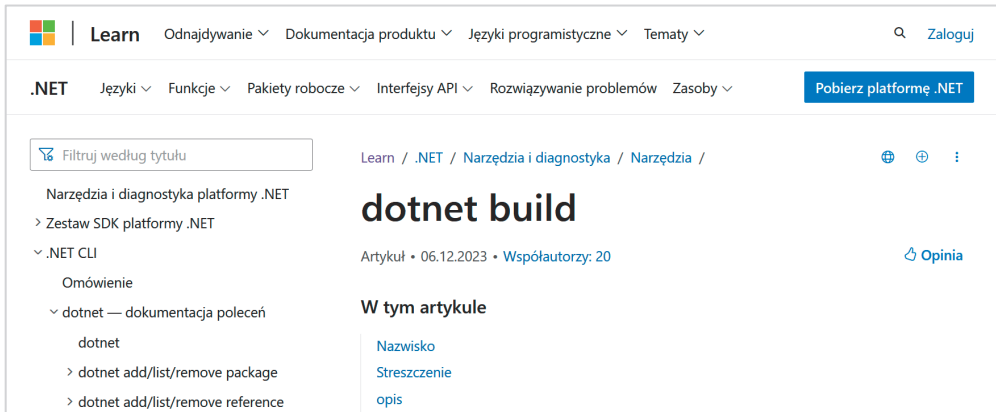




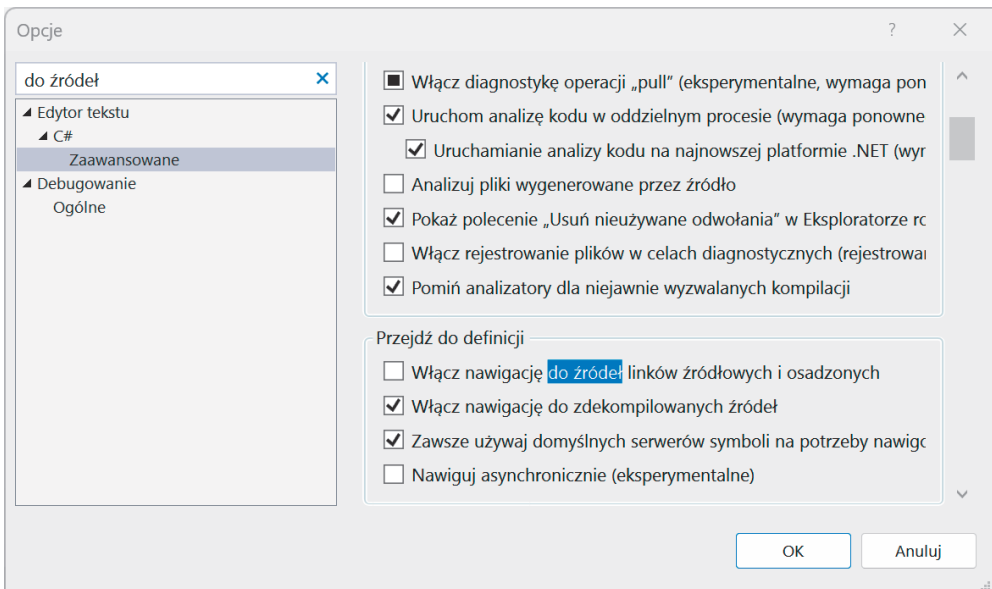
Rysunek 1.12. Uruchomienie programu najwyższego poziomu w Visual Studio Code dwoma projektami



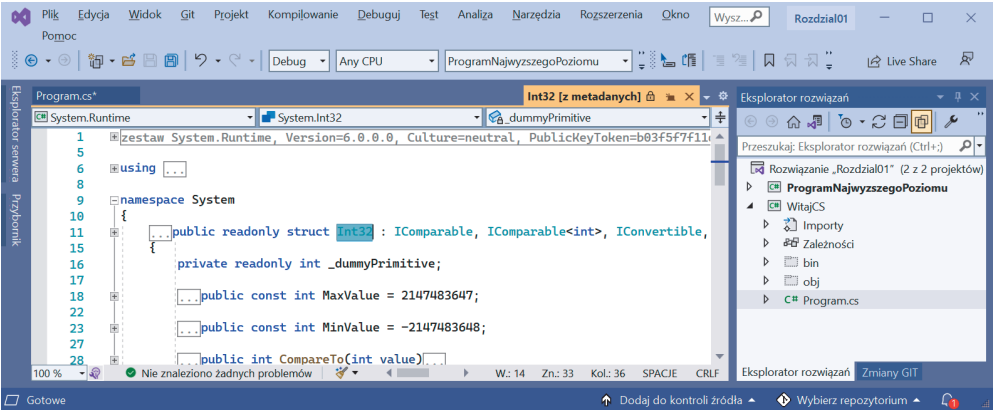
Rysunek 1.13. Pobieranie repozytorium jako pliku ZIP



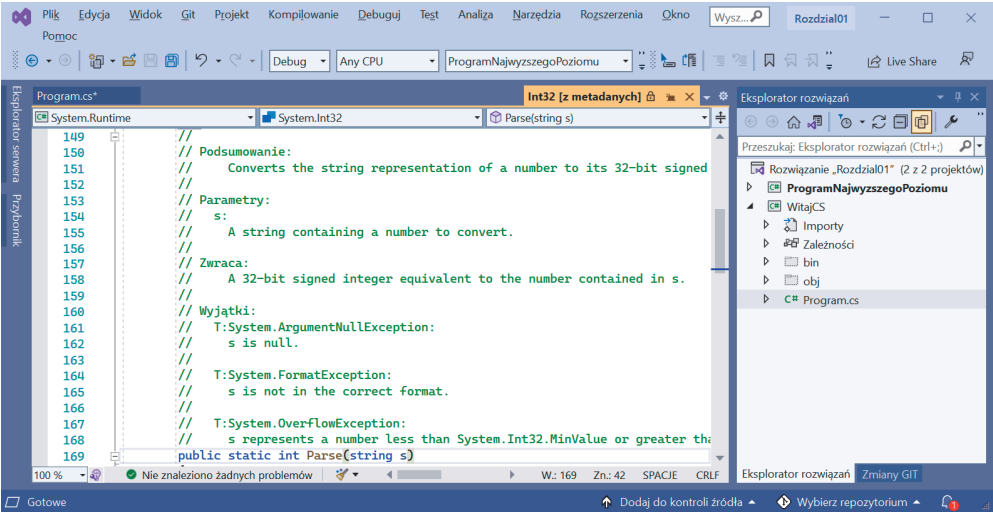
Rysunek 1.14. Strona WWW z dokumentacją polecenia dotnet build



Rysunek 1.15. Wyłączanie nawigacji do źródeł dla funkcji Przejdź do definicji



Rysunek 1.16. Metadane dla typu int



Rysunek 1.17. Komentarze do metody Parse z parametrem typu string

Visual Studio 2022

Visual Studio Code

Wskazówki w tekście

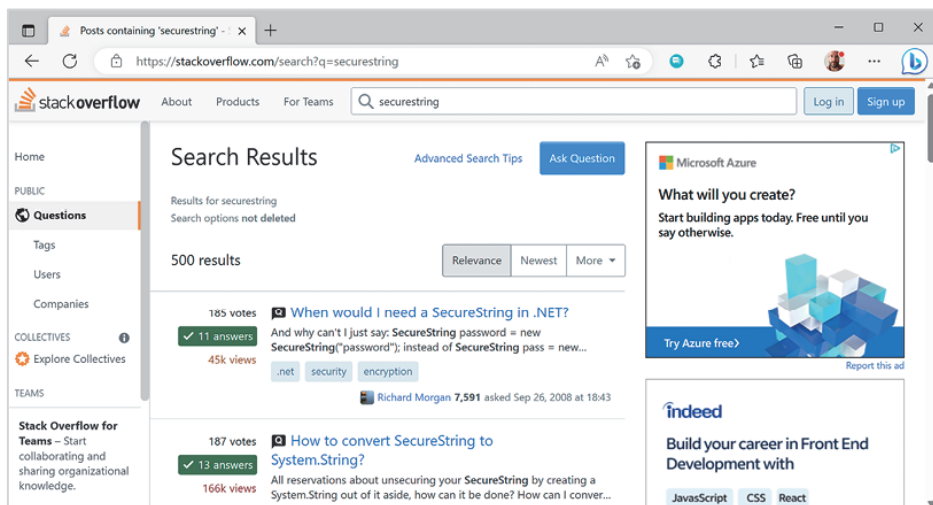
- ☒ Wyświetl wszystkie wskazówki po naciśnięciu klawiszy Alt+F1
- ☒ Wskazówki kolorów
- ☒ Wyświetl wskazówki w tekście dla nazw parametrów

Dotnet > Inlay Hints: Enable Inlay Hints For Parameters

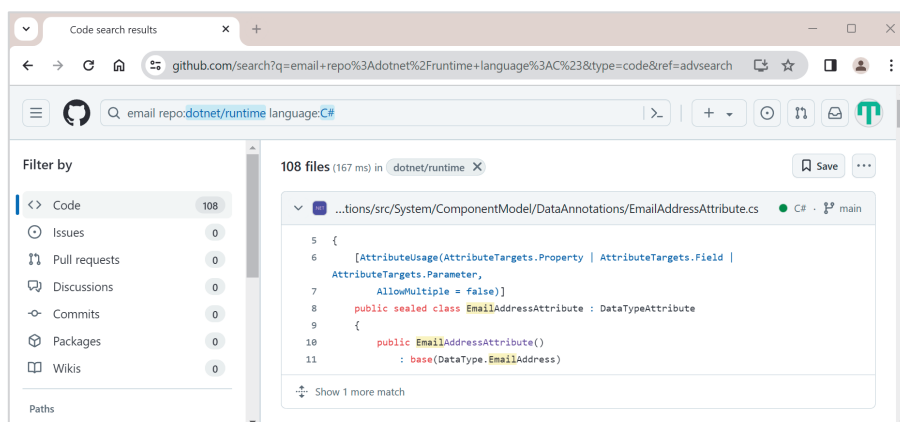
- ☒ Display inline parameter name hints

Console.WriteLine(format: "Wartość to {0}", arg0: 19.8);

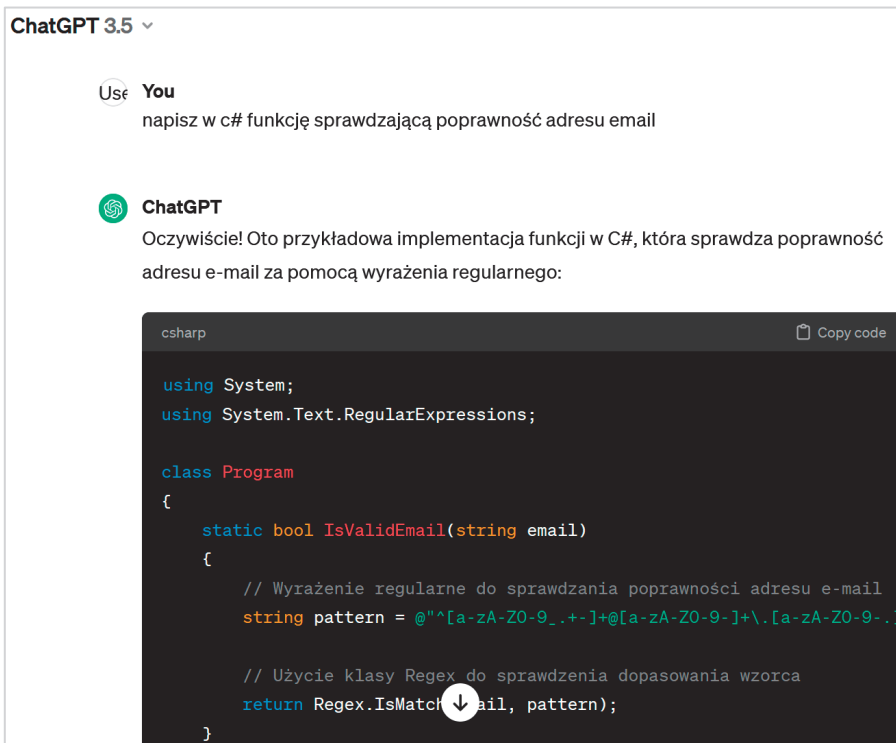
Rysunek 1.18. Konfigurowanie wskazówek w tekście



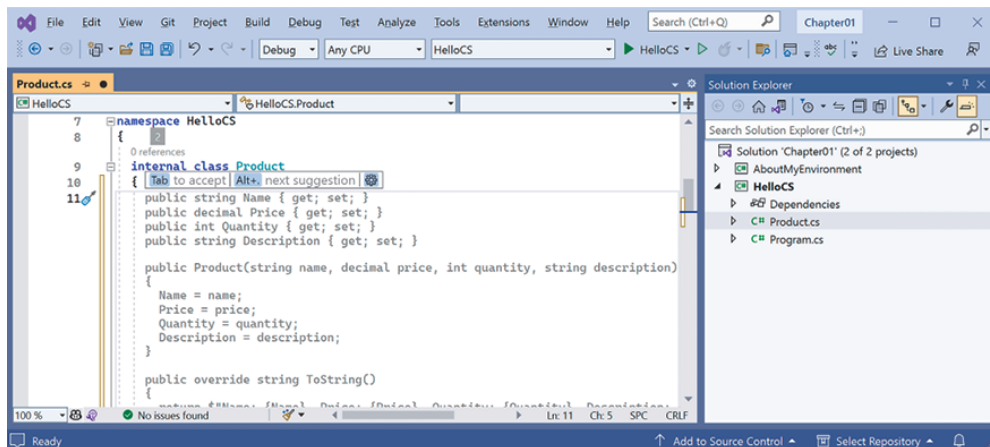
Rysunek 1.19. Wyniki wyszukiwania słowa **securestring** w serwisie Stack Overflow



Rysunek 1.20. Zaawansowane wyszukiwanie słowa „email” w repozytorium dotnet/runtime



Rysunek 1.21. ChatGPT tworzy funkcję sprawdzającą adres e-mail

Rysunek 1.22. GitHub Copilot sugeruje, jak zdefiniować klasę `Product`

https://github.com/dotnet/installer?tab=readme-ov-file#table

READMECode of conductMIT licenseSecurity

Table

Platform	main (9.0.x Runtime)	9.0.1xx-preview3 (9.0- preview3 Runtime)	8.0.3xx (8.0 Runtime)	Release/7.0.4xx (7.0.x Runtime)
Windows x64	<div>version9.0.100-preview.4.24175.2</div> <div><a href="#">Installer</a> - <a href="#">Checksum</a> <a href="#">zip</a> - <a href="#">Checksum</a></div>	<div>version9.0.100-preview.3.24175.2</div> <div><a href="#">Installer</a> - <a href="#">Checksum</a> <a href="#">zip</a> - <a href="#">Checksum</a></div>	<div>version8.0.300-preview.24175.2</div> <div><a href="#">Installer</a> - <a href="#">Checksum</a> <a href="#">zip</a> - <a href="#">Checksum</a></div>	<div>version7.0.400</div> <div><a href="#">Installer</a> - <a href="#">Checksum</a> <a href="#">zip</a> - <a href="#">Checksum</a></div>

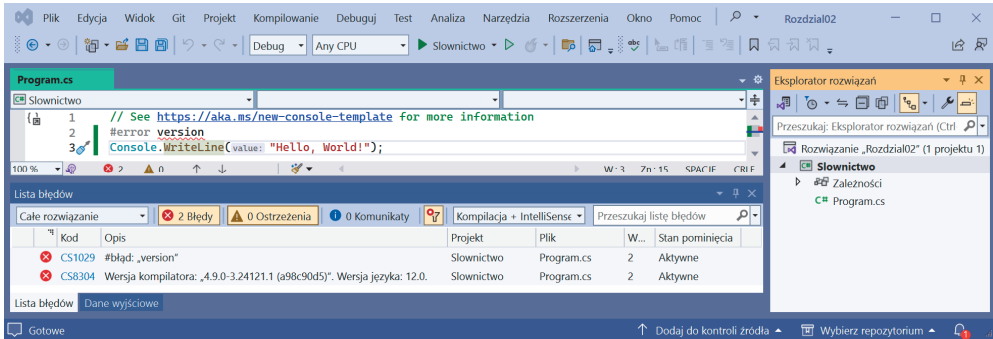
Pobrane

Plik dotnet-sdk-9.0.100-preview.4.24175.23-win-x64.exe nie jest często pobierany. Przed otwarciem upewnij się, że...

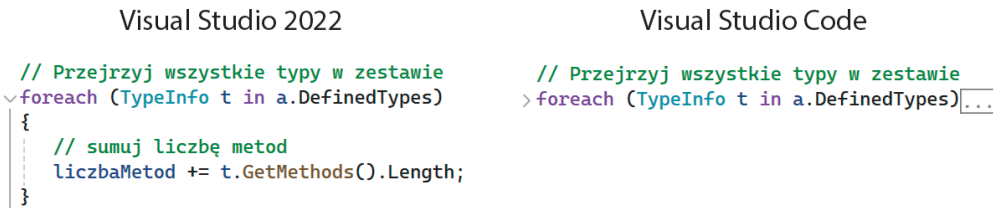
Zobacz więcej

Rysunek 1.23. Strona pobierania wersji alfa platformy .NET

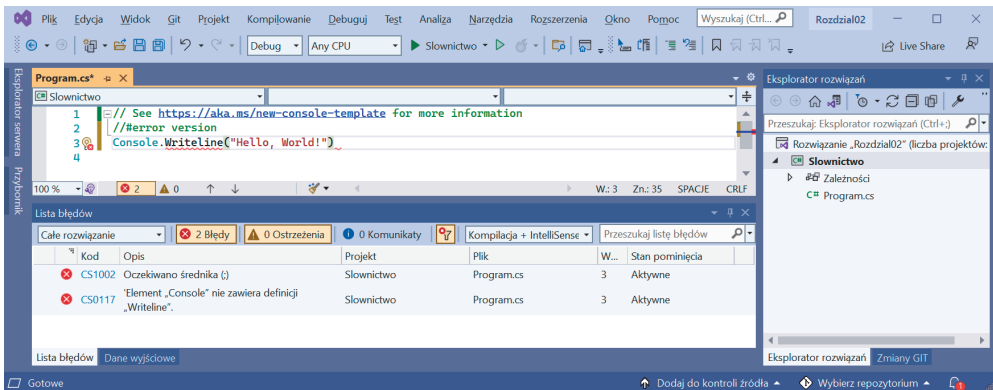
# Rozdział 2. Mówimy w C#



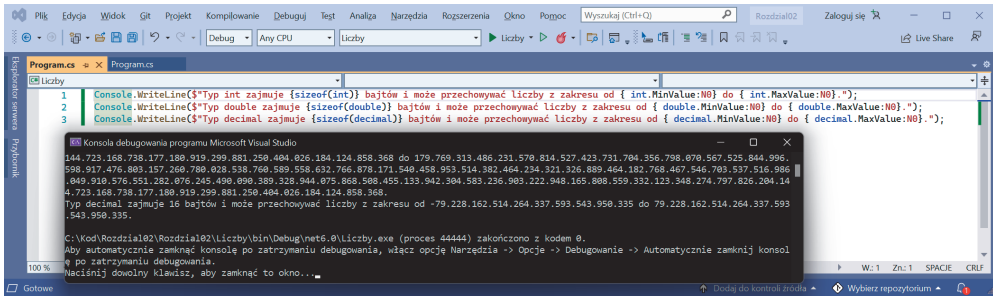
Rysunek 2.1. Błąd kompilatora podający numer wersji języka C#



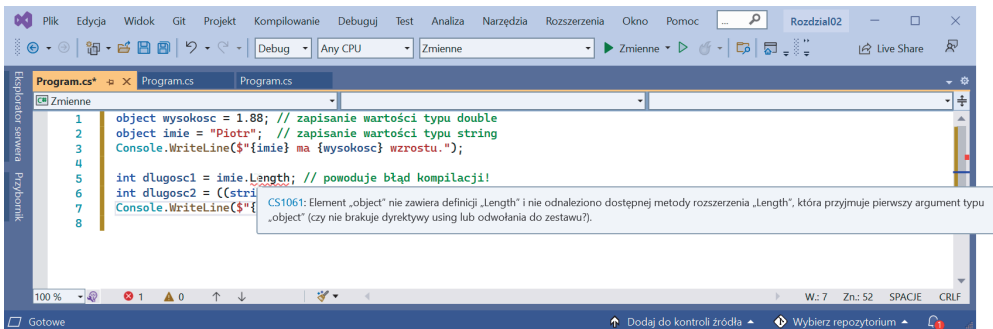
Rysunek 2.2. Edytor kodu z rozwiniętymi i zwiniętymi blokami



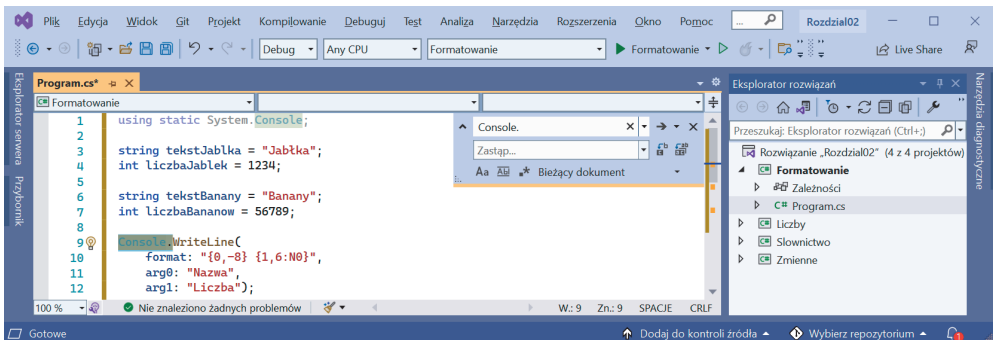
Rysunek 2.3. W okienku Lista błędów widoczne są dwa błędy kompilacji



Rysunek 2.5. Wielkość i zakres wartości dla typowych typów danych

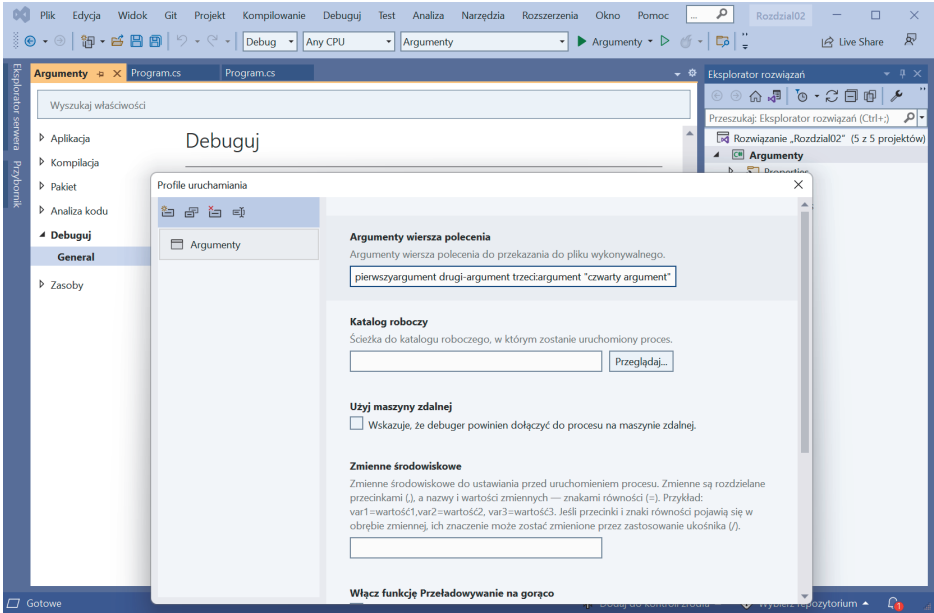


Rysunek 2.7. Typ object nie ma właściwości Length

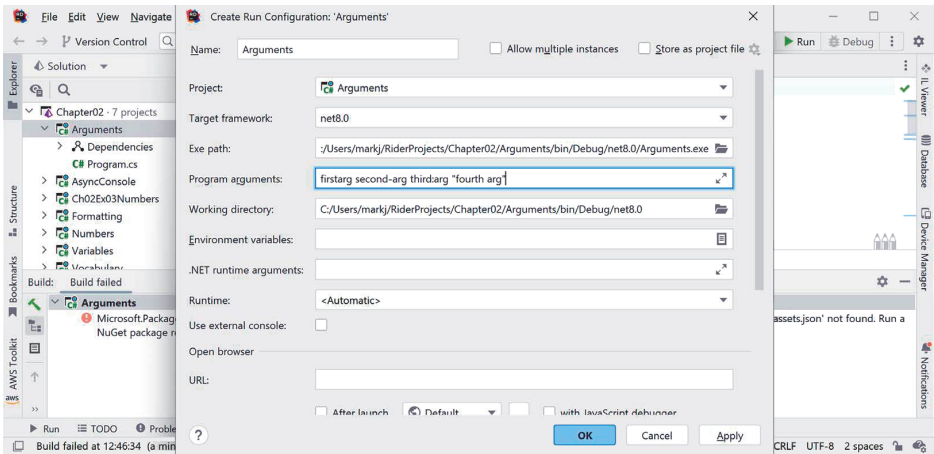


Rysunek 2.8. Użycie funkcji zamiany w Visual Studio w celu uproszczenia kodu

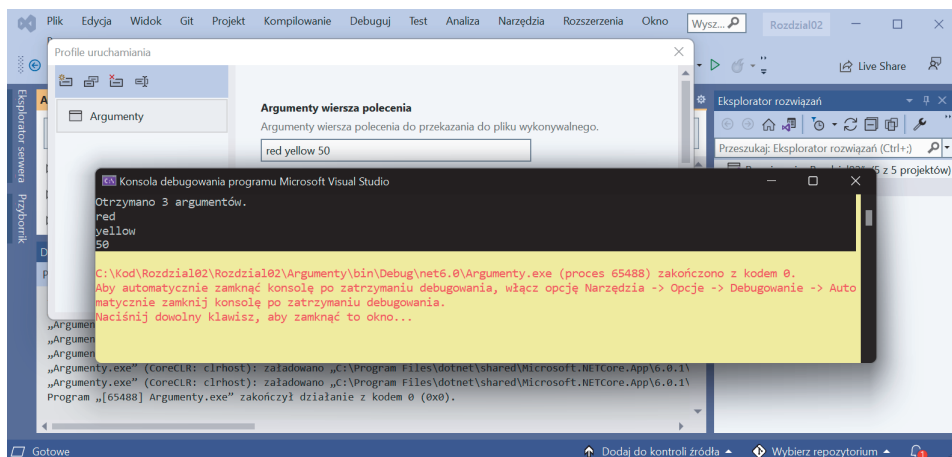




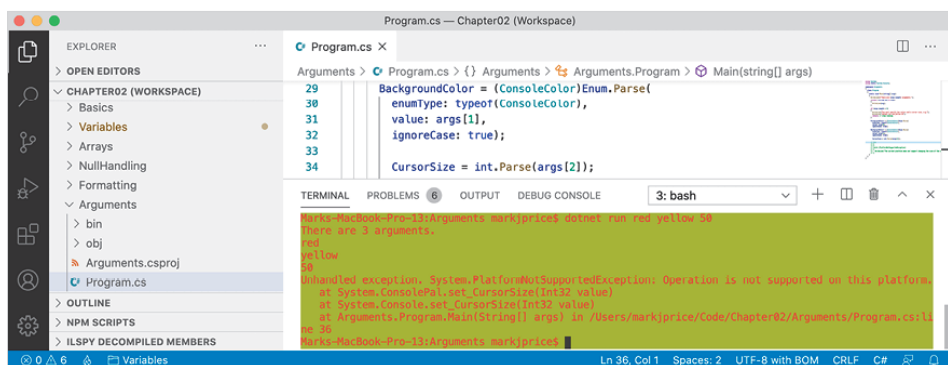
Rysunek 2.9. Wprowadzanie argumentów aplikacji we właściwościach projektu Visual Studio



Rysunek 2.10. Wprowadzanie argumentów wiersza poleceń w konfiguracji uruchamiania w JetBrains Rider

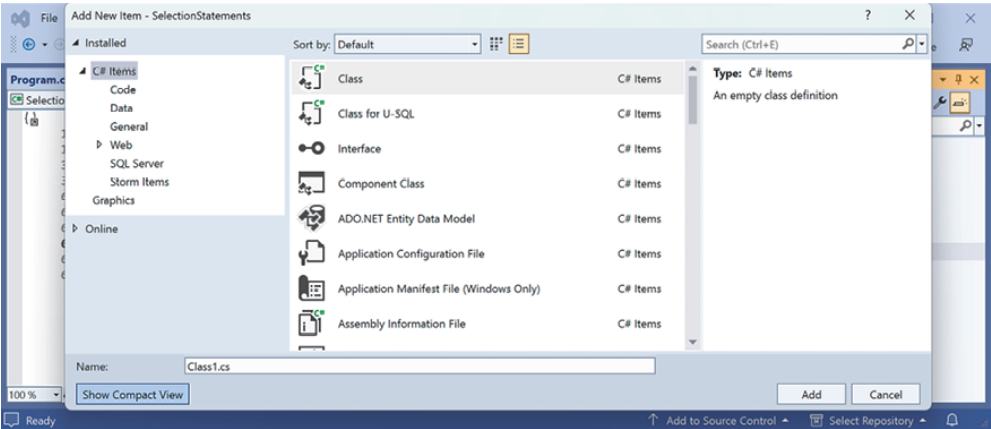


Rysunek 2.11. Ustalanie kolorów i wielkości kursora w systemie Windows

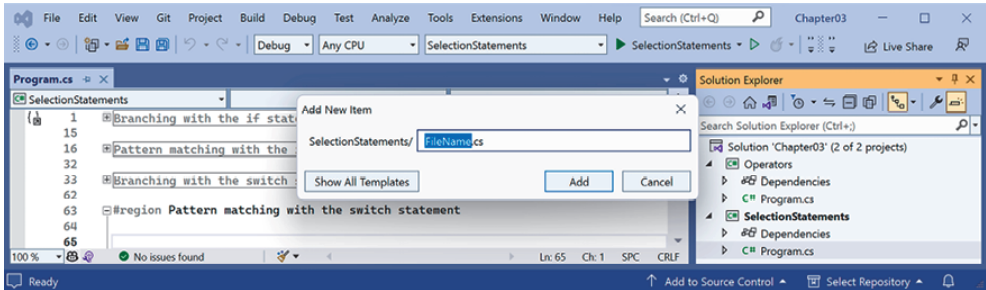


Rysunek 2.12. Nieobsłużony wyjątek w systemie macOS

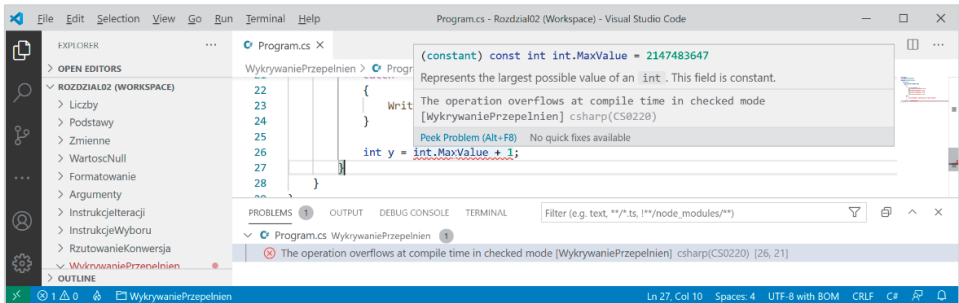
# Rozdział 3. Sterowanie przepływem, konwertowanie typów i obsługa wyjątków



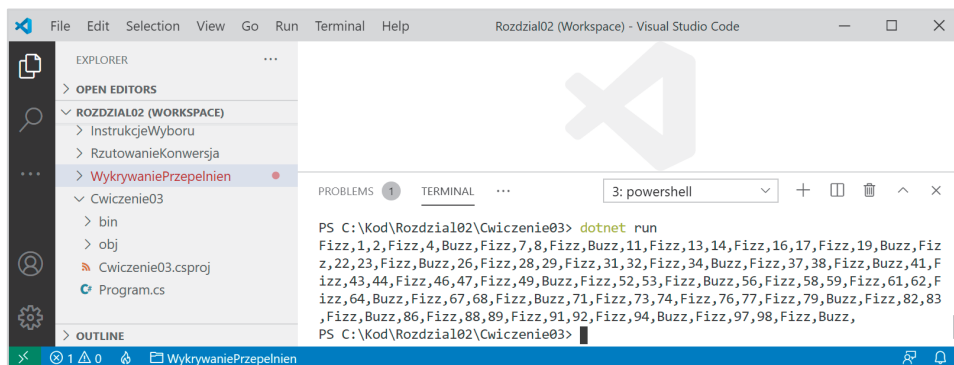
Rysunek 3.1. Tradycyjny widok dodawania nowego elementu



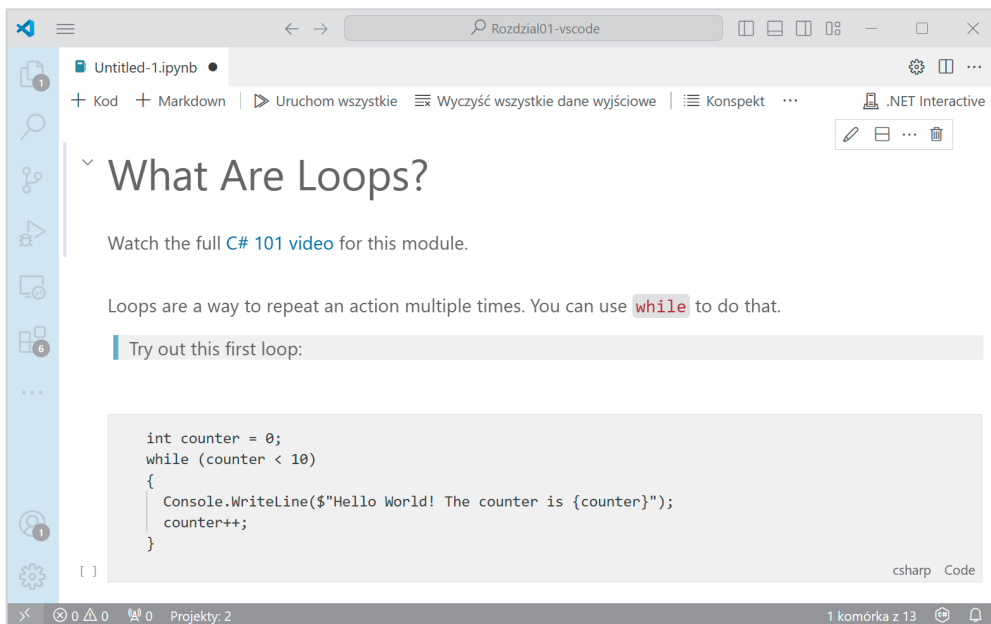
Rysunek 3.2. Uproszczony widok dodawania nowego elementu



Rysunek 3.4. Kontrola przepełnień wykonywana w czasie kompilacji

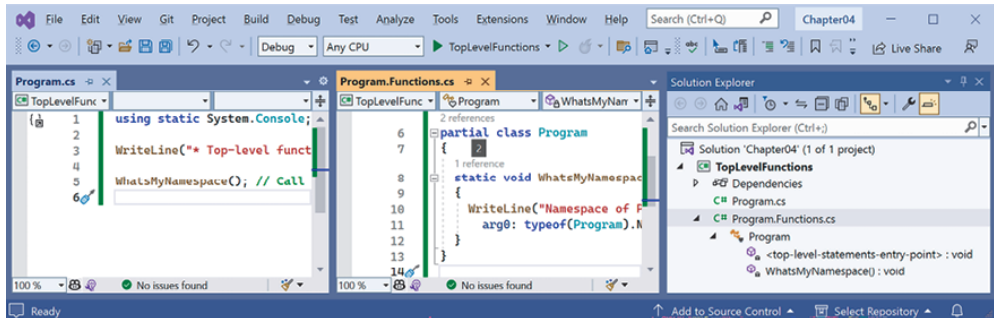


Rysunek 3.5. Wynik działania programu grającego w FizzBuzz

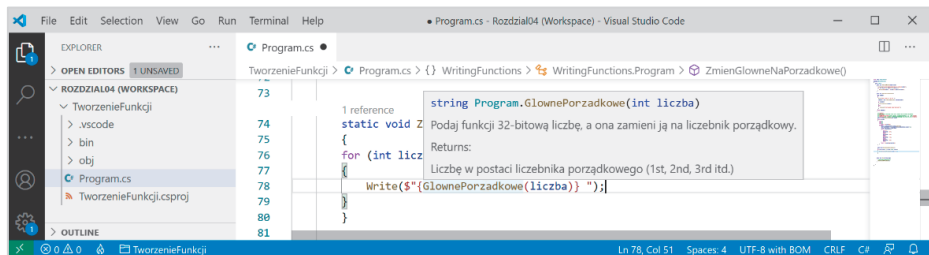


Rysunek 3.6. Przykładowy notatnik zatytułowany Czym są pętle?

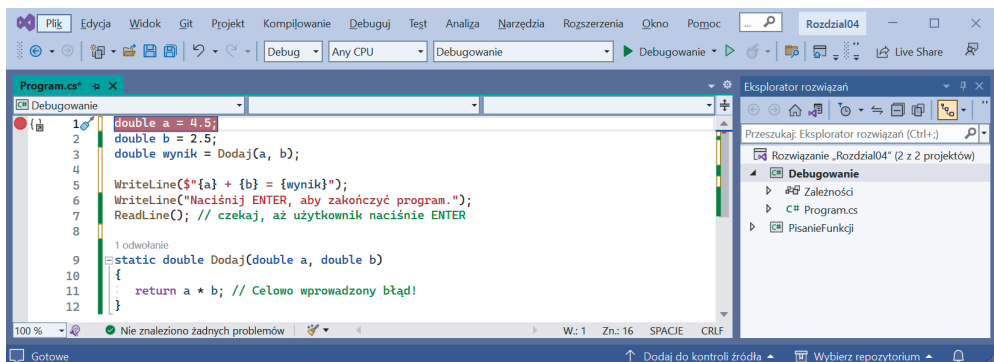
# Rozdział 4. Pisanie, debugowanie i testowanie funkcji



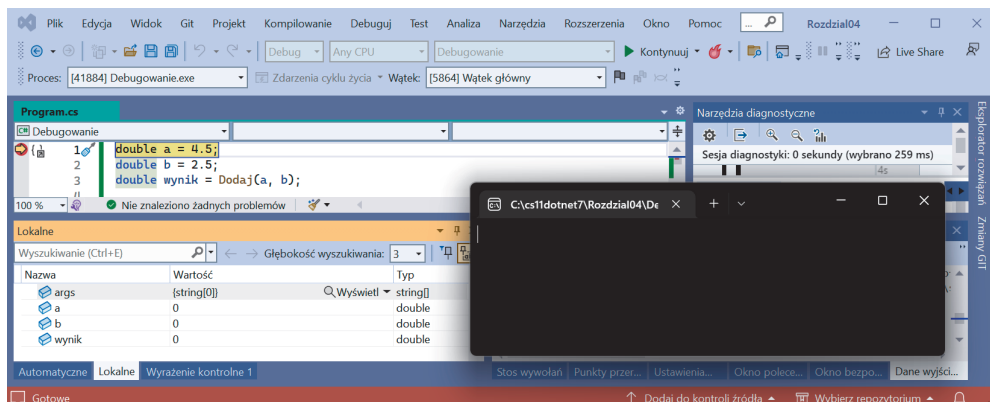
Rysunek 4.1. Eksplorator rozwiązań pokazuje złączenie klasy Program



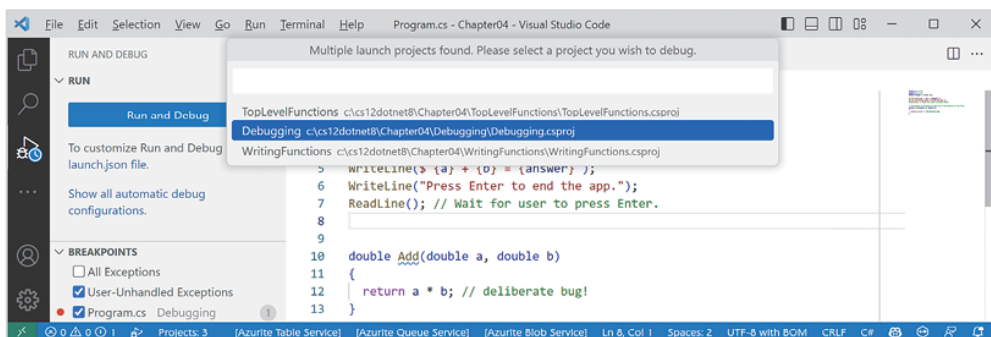
Rysunek 4.2. Podpowiedź zawiera więcej informacji o sygnaturze metody



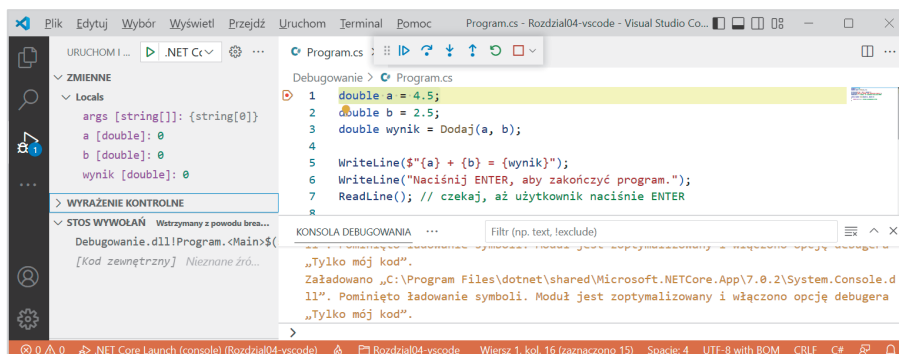
Rysunek 4.3. Przełączanie punktu przerwania w Visual Studio 2022



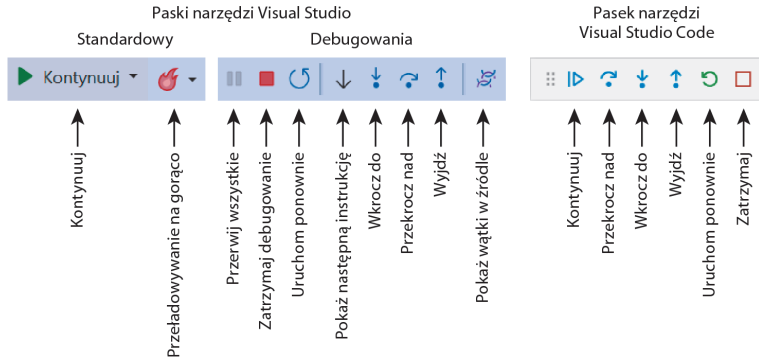
Rysunek 4.4. Tryb przerwania w Visual Studio 2022



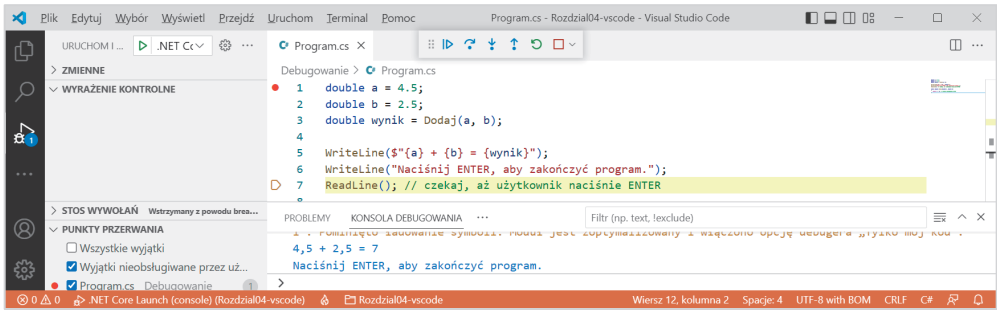
Rysunek 4.5. Wybieranie projektu do debugowania w Visual Studio Code



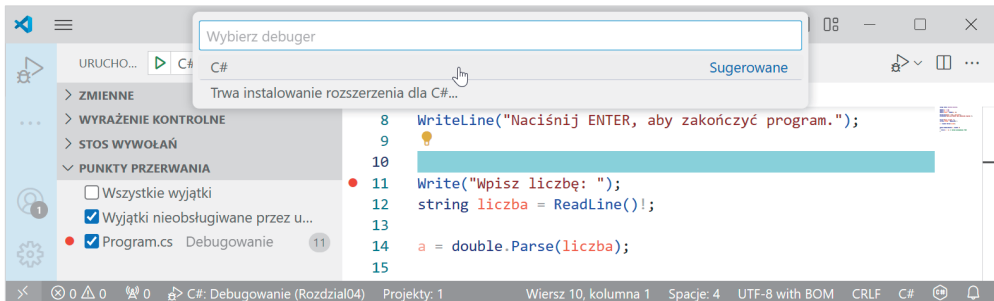
Rysunek 4.6. Tryb przerwania w Visual Studio Code



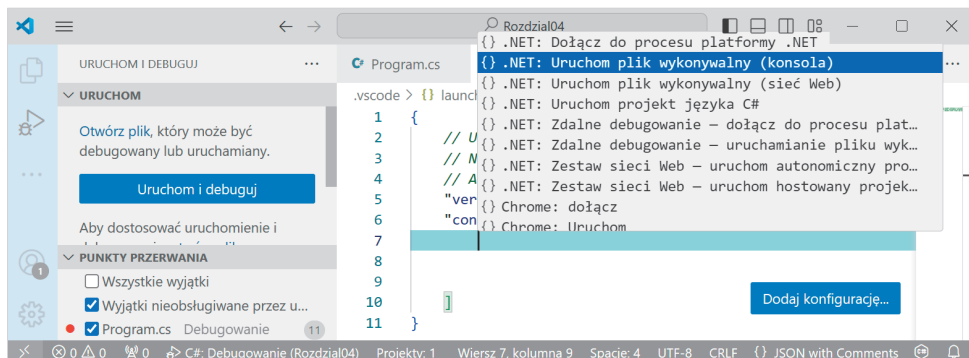
Rysunek 4.7. Paski narzędzi debugowania w Visual Studio 2022 i Visual Studio Code



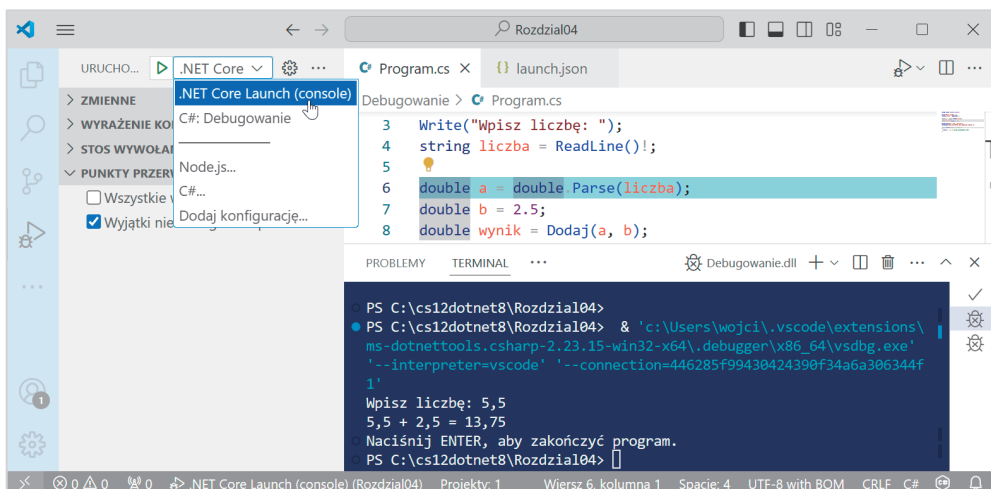
Rysunek 4.8. W trakcie debugowania komunikaty są wypisywane w okienku Konsola debugowania



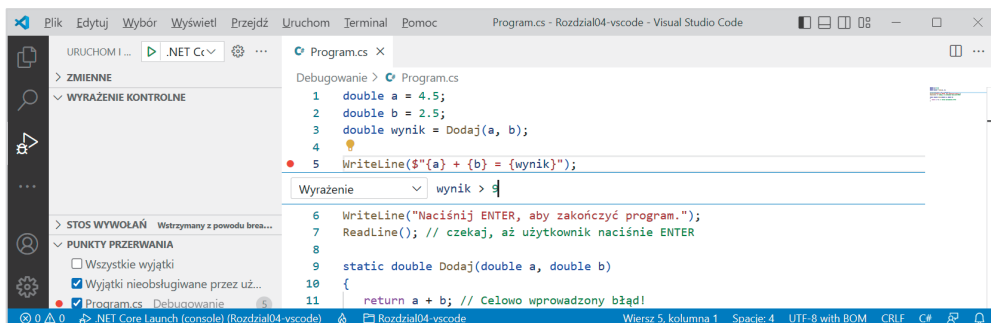
Rysunek 4.9. Wybieranie debugera dla pliku launch.json



Rysunek 4.10. Dodawanie konfiguracji uruchamiania dla aplikacji konsoli

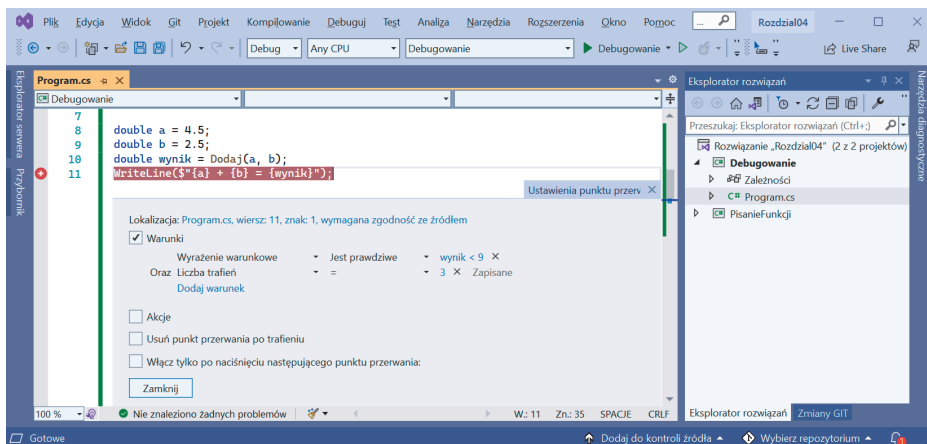


Rysunek 4.11. Wybieranie konfiguracji uruchamiania

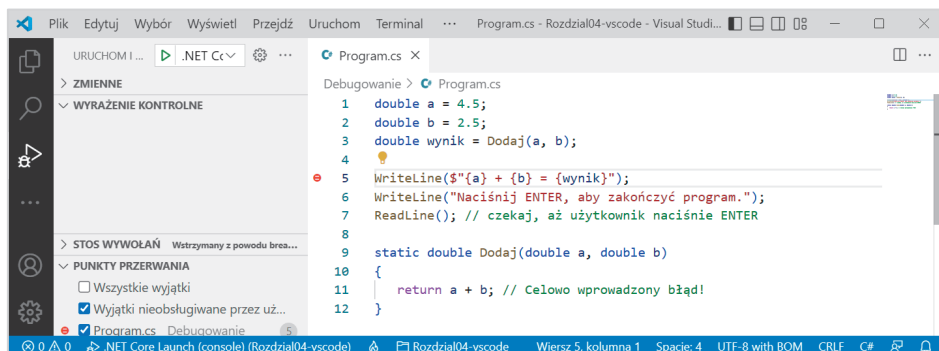


Rysunek 4.12. Dostosowywanie punktu przerwania za pomocą wyrażenia (w Visual Studio Code)

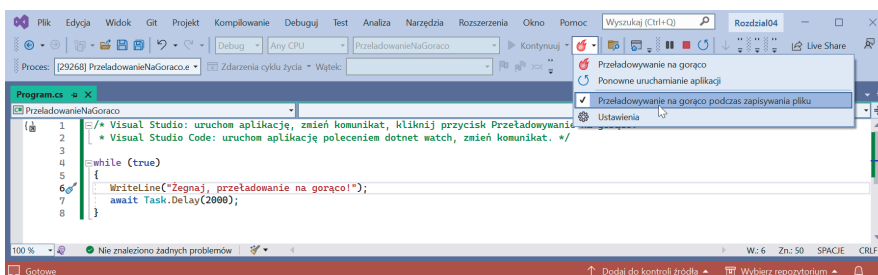




Rysunek 4.13. Dostosowywanie punktu przerwania za pomocą wyrażenia i licznika przejść w Visual Studio 2022

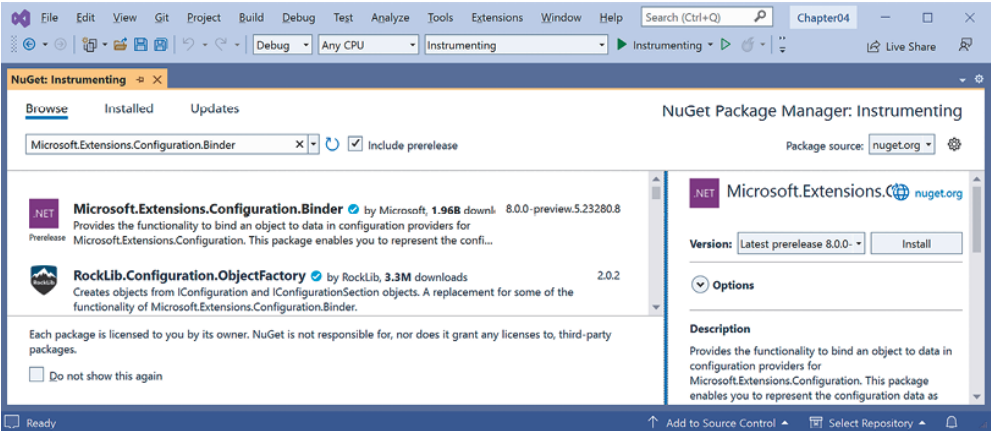


Rysunek 4.14. Podsumowanie dostosowanego punktu przerwania w Visual Studio Code

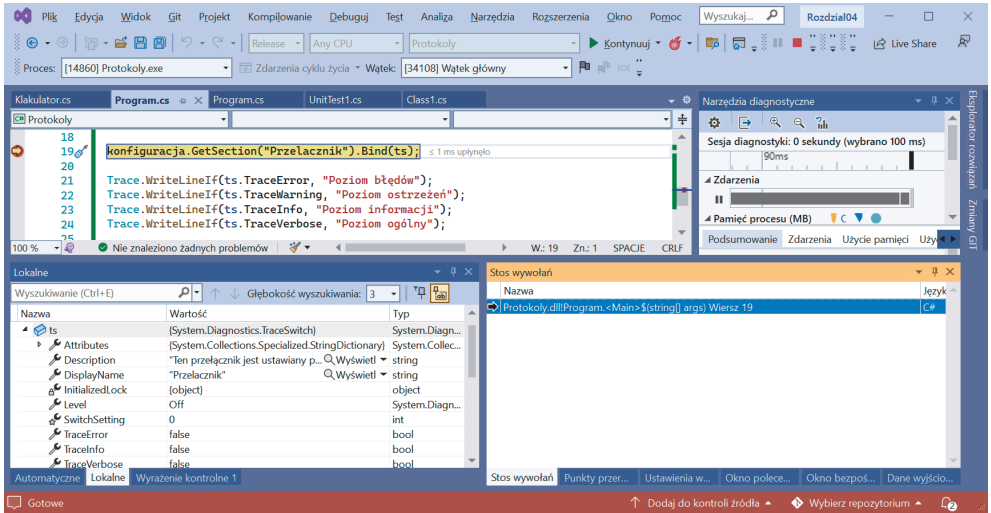


Rysunek 4.15. Opcje dla funkcji Przeladowywanie na gorąco

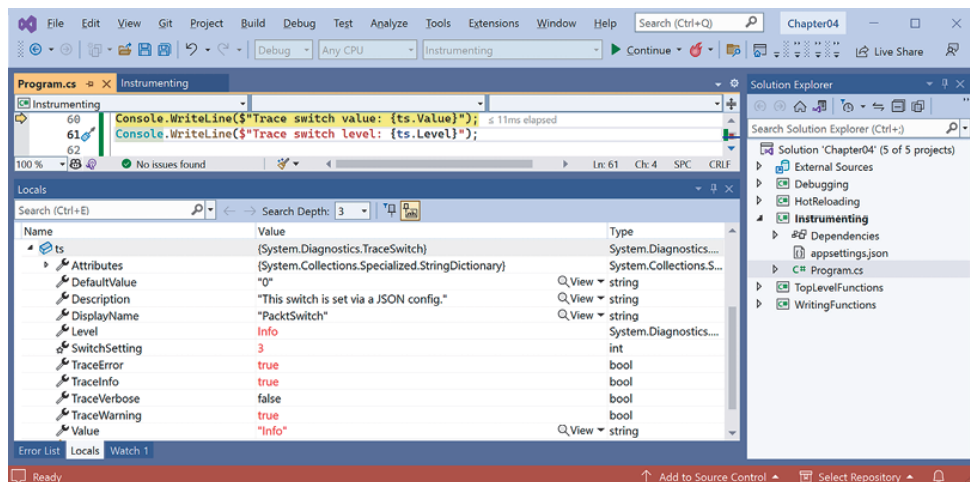




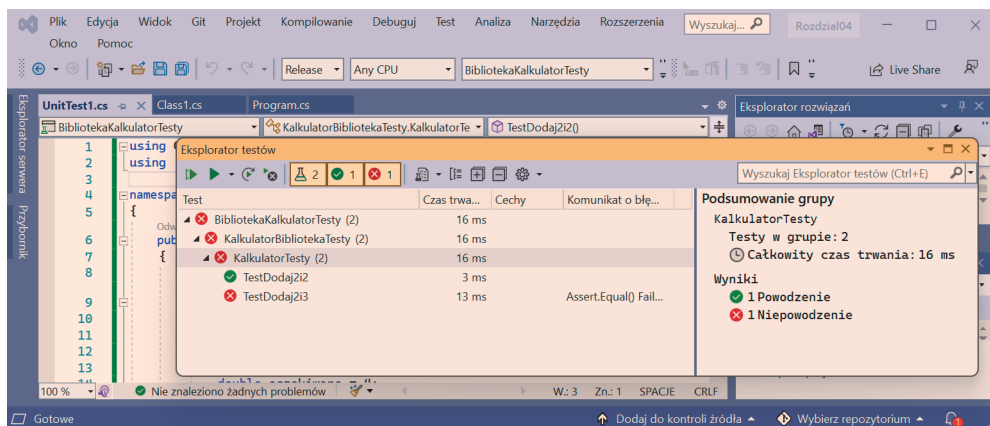
Rysunek 4.19. Instalowanie pakietów NuGet w Visual Studio 2022



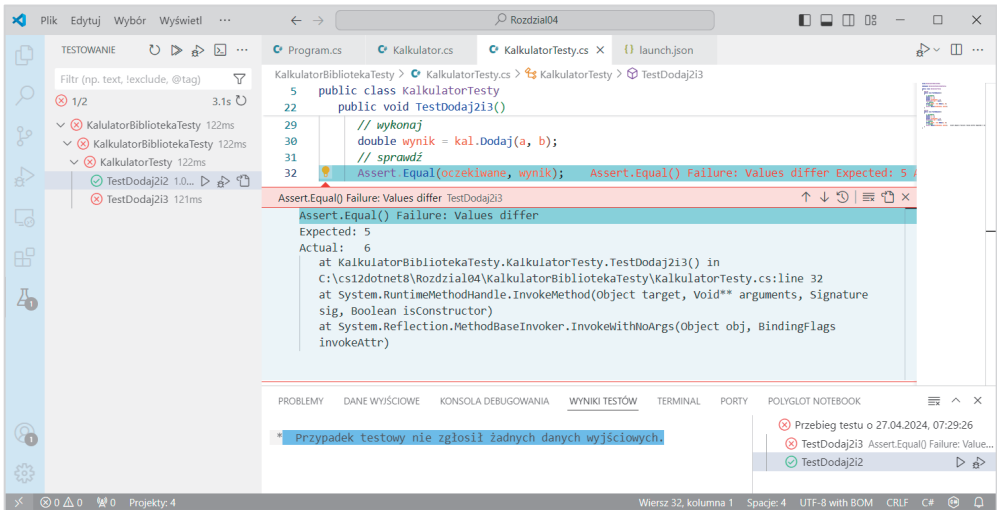
Rysunek 4.20. Przeglądanie wartości zmiennej przełącznika protokołowania w Visual Studio 2022



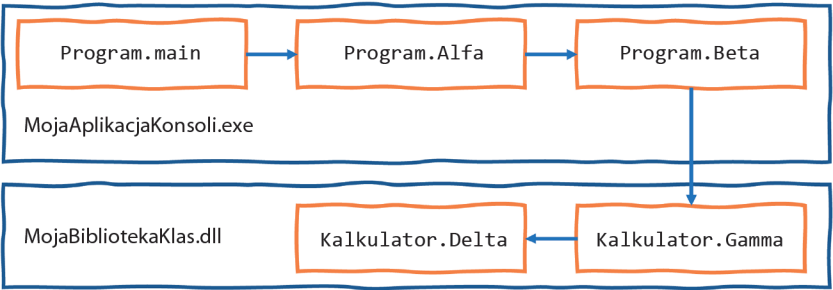
Rysunek 4.21. Włączone wszystkie poziomy śledzenia oprócz TraceVerbose



Rysunek 4.22. Wyniki testów jednostkowych w okienku Eksplorator testów w Visual Studio 2022



Rysunek 4.23. Wyniki testów jednostkowych w oknie WYNIKI TESTÓW w Visual Studio Code



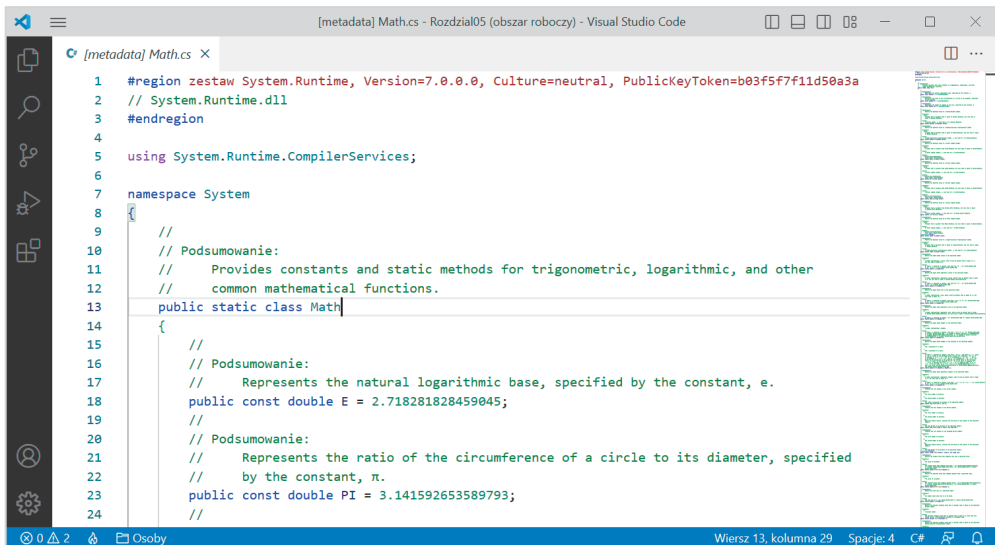
Rysunek 4.24. Łańcuch wywołań metod tworzy tak zwany stos wywołań

## Rozdział 5. Tworzenie własnych typów w programowaniu obiektowym

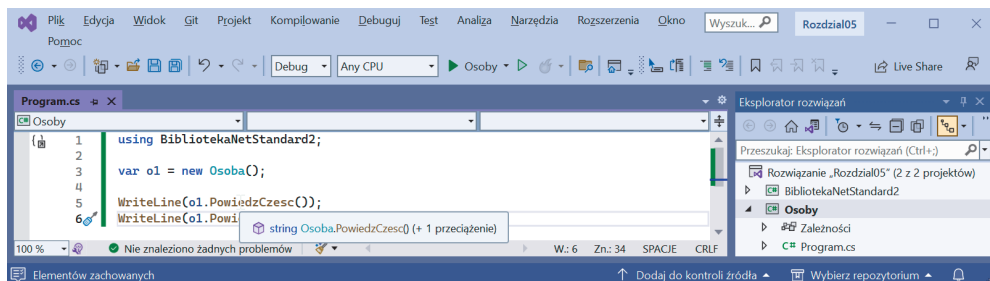
```
namespace System
{
    ...public class Object
    {
        ...public Object();
        ...~Object();

        ...public static bool Equals(Object objA, Object objB);
        ...public static bool ReferenceEquals(Object objA, Object objB);
        ...public virtual bool Equals(Object obj);
        ...public virtual int GetHashCode();
        ...public Type GetType();
        ...public virtual string ToString();
        ...protected Object MemberwiseClone();
    }
}
```

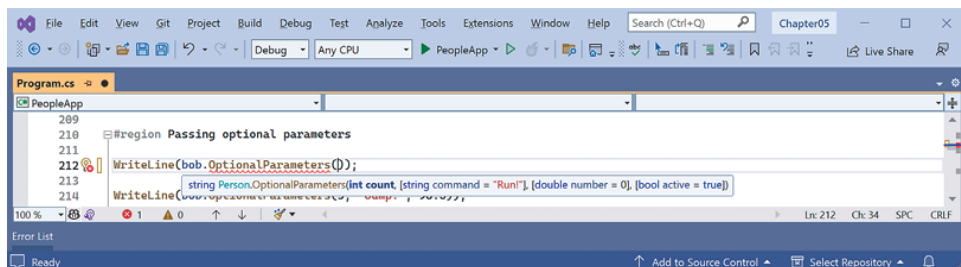
Rysunek 5.1. Definicja klasy System.Object w .NET Standard 2.0



Rysunek 5.2. Przykłady wartości stałych

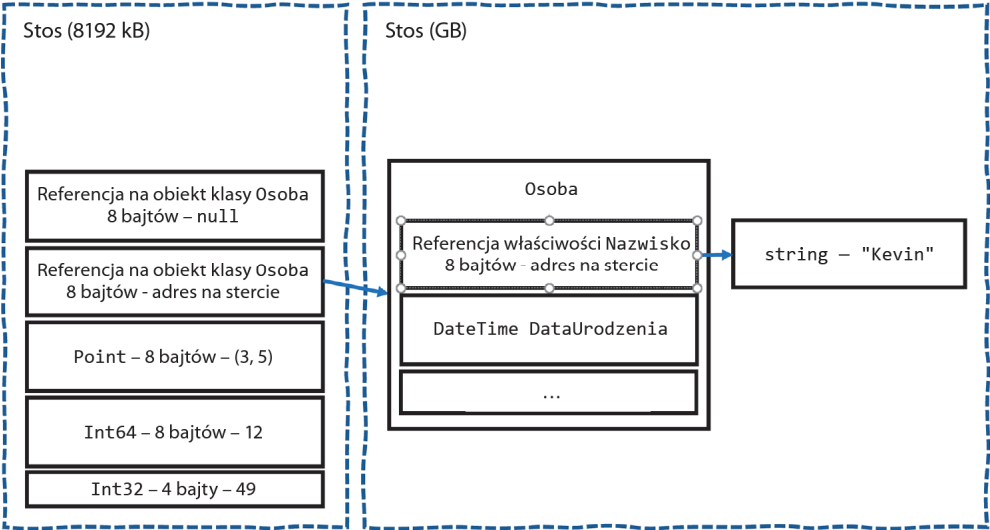


Rysunek 5.3. Podpowieź IntelliSense pokazująca metodę przeciążoną



Rysunek 5.4. IntelliSense wyświetla listę parametrów wymaganych i opcjonalnych

# Rozdział 6. Implementowanie interfejsów i dziedziczenie klas

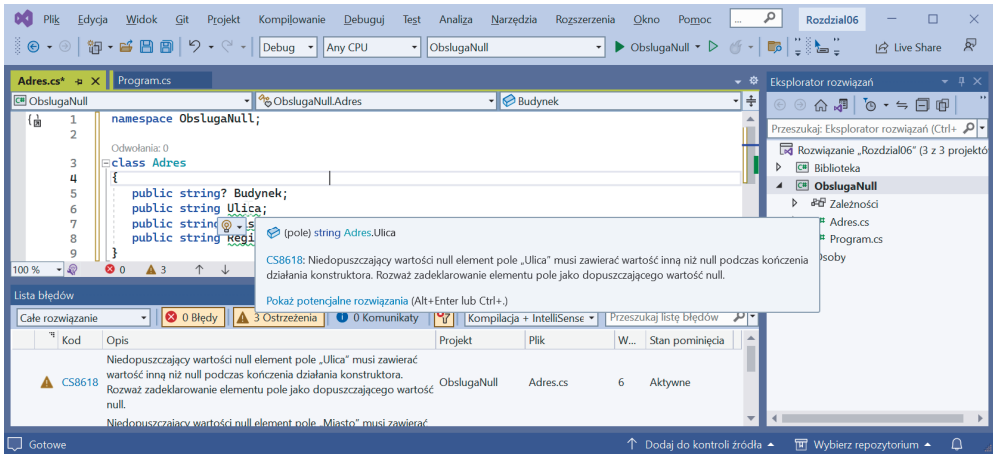


Rysunek 6.1. W taki sposób typy wartości i typy referencyjne zapisywane są na stosie i na stercie

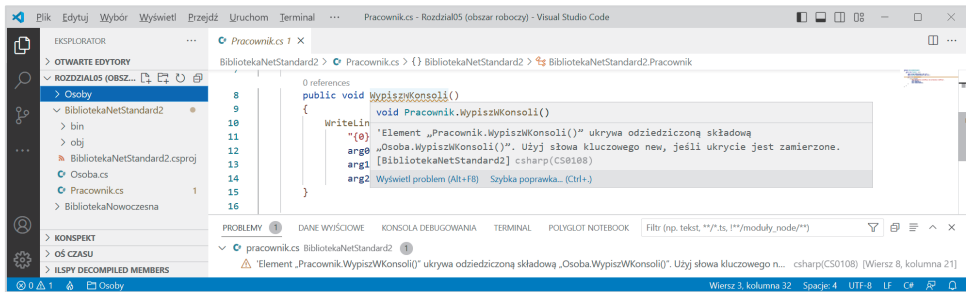
```
1  namespace System
2  {
3      public struct Nullable<T> where T : struct
4      {
5          public Nullable(T value);
6          public readonly bool HasValue { get; }
7          public readonly T Value { get; }
8          public override bool Equals(object? other);
9          public override int GetHashCode();
10         public readonly T GetValueOrDefault();
11         public readonly T GetValueOrDefault(T defaultValue);
12         public override string? ToString();
13         public static implicit operator T?(T value);
14         public static explicit operator T(T? value);
15     }
16 }
```

Rysunek 6.2. Informacje o elementach typu Nullable<T>

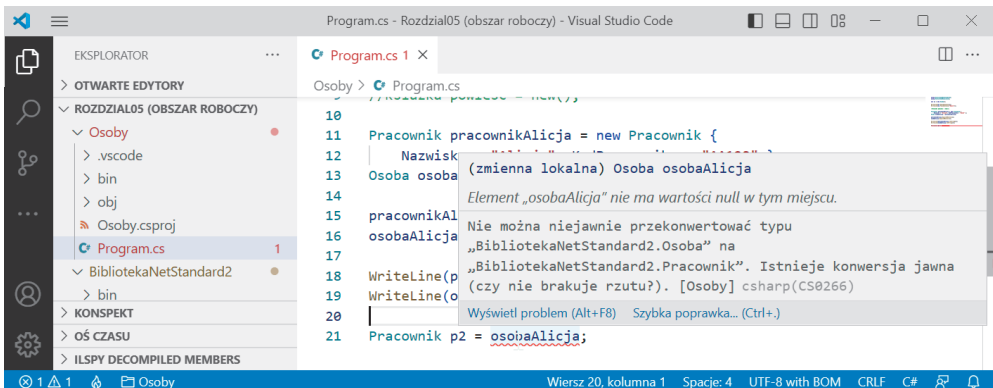




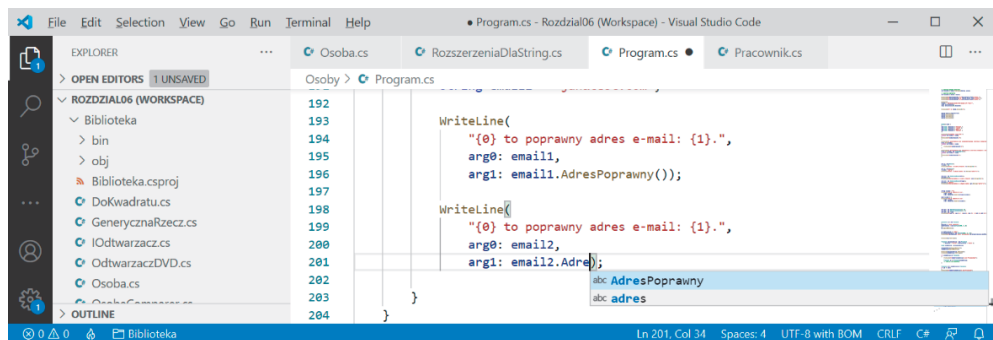
Rysunek 6.3. Ostrzeżenie związane z używaniem nienullowalnych typów danych



Rysunek 6.4. Ostrzeżenie o ukrywaniu metod

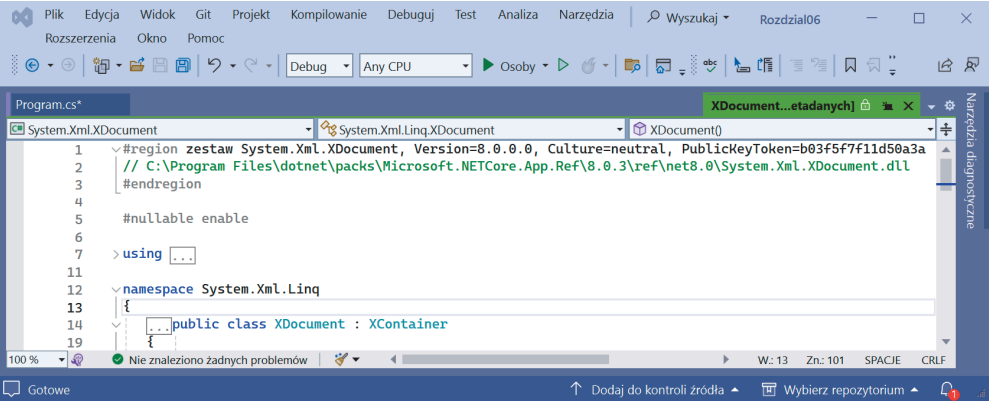


Rysunek 6.5. Błąd kompilacji wynikający z braku jawnego rzutowania

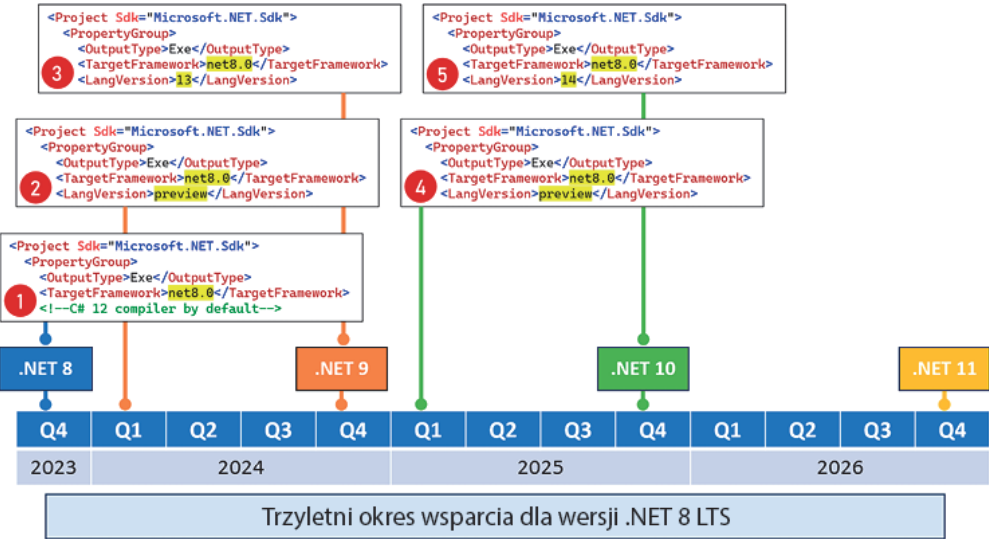


**Rysunek 6.6. Metoda rozszerzająca wyświetlona w okienku IntelliSense obok innych metod obiektu**

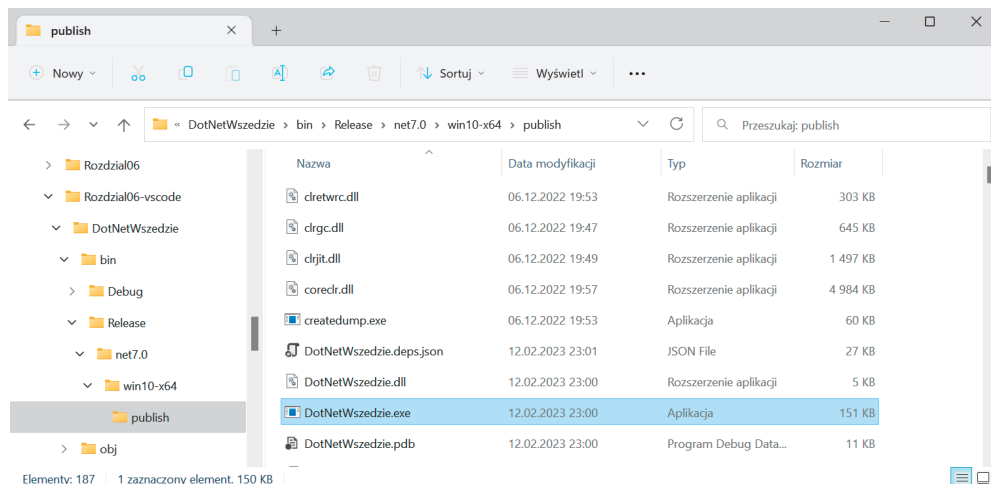
# Rozdział 7. Poznawanie typów .NET



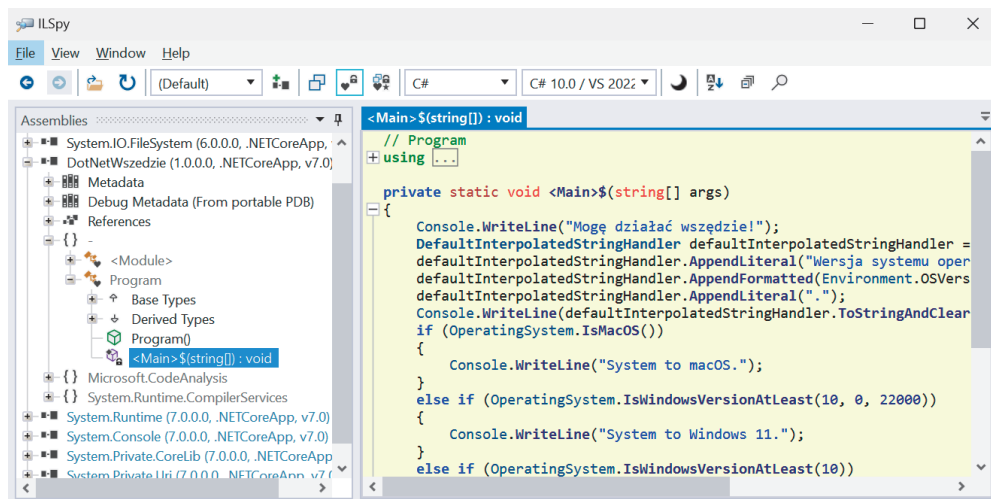
Rysunek 7.1. Zestaw i przestrzeń nazw zawierające typ XDocument



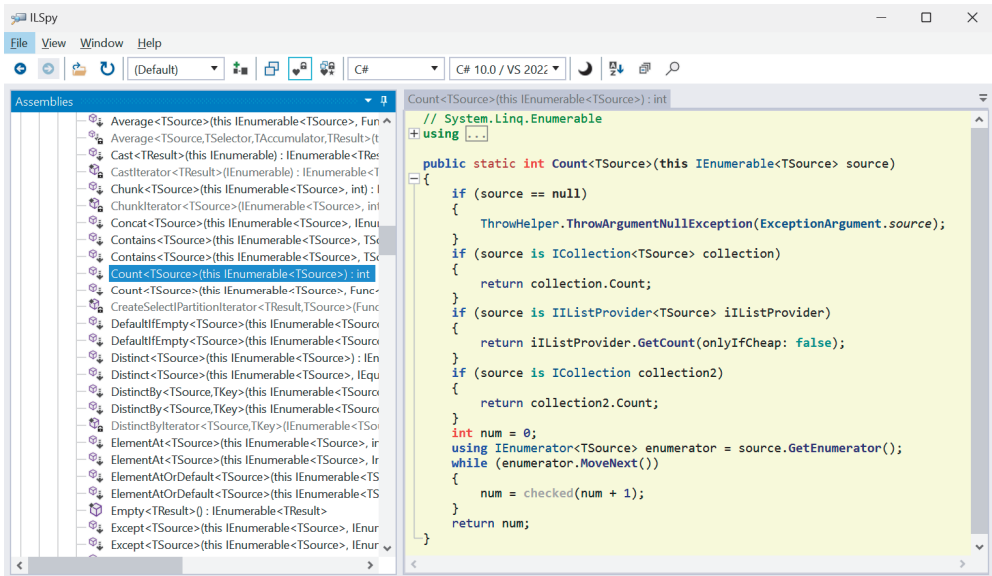
Rysunek 7.2. Tworzenie projektu .NET 8 z wykorzystaniem najnowszych kompilatorów C#



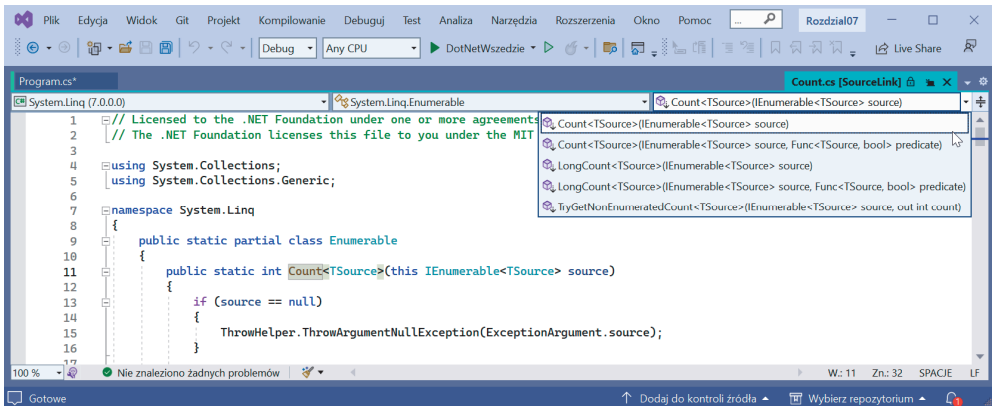
Rysunek 7.3. Plik wykonywalny DotNetWszedzie dla 64-bitowego systemu Windows 10



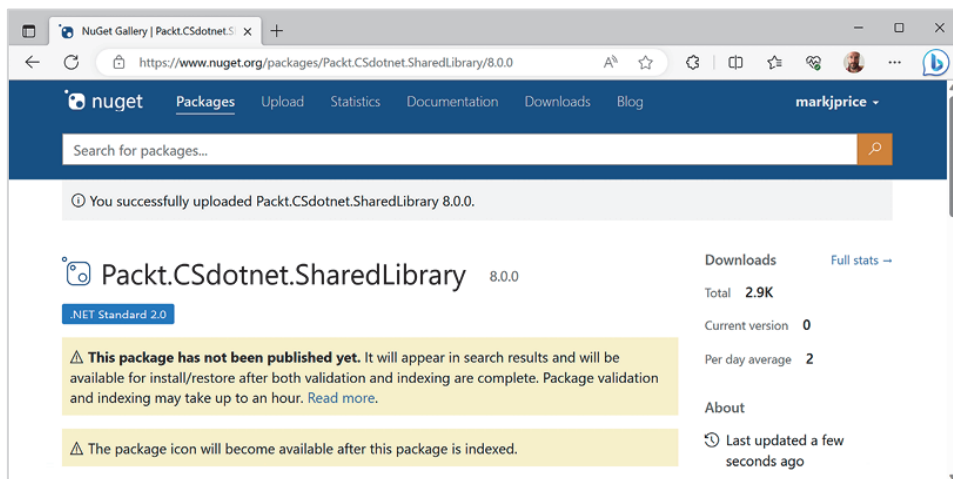
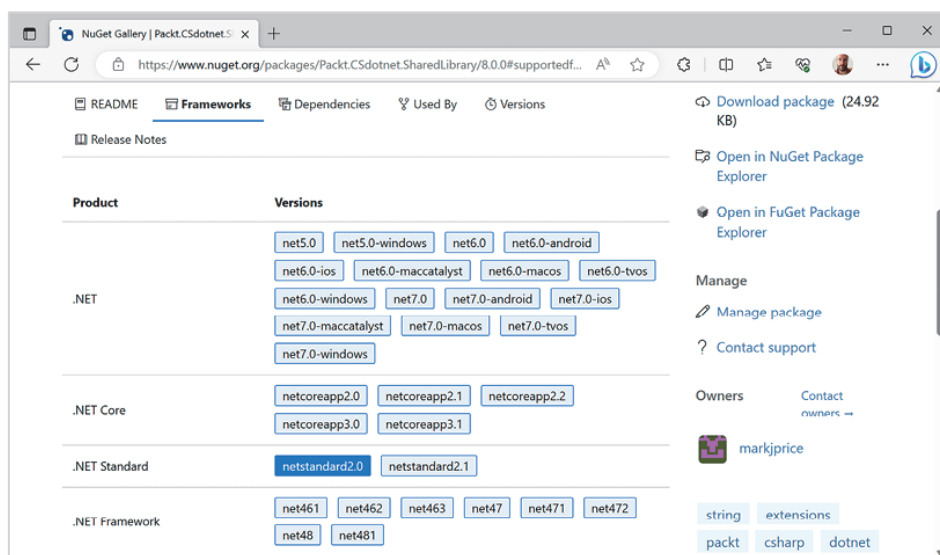
Rysunek 7.4. Zdekompileowana metoda <Main>\$ oraz informacja o tym, jak działają interpolowane ciągi znaków

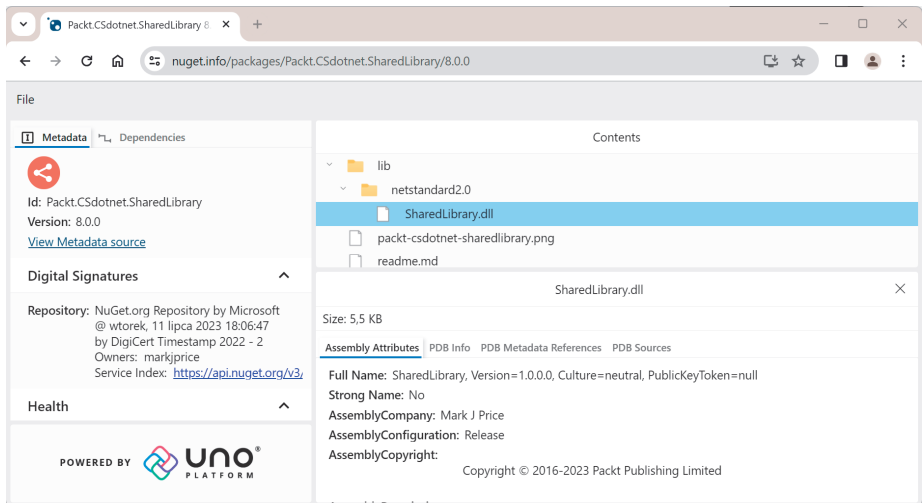


**Rysunek 7.5. Zdekompilowana metoda Count w klasie Enumerable dla systemu Linux**



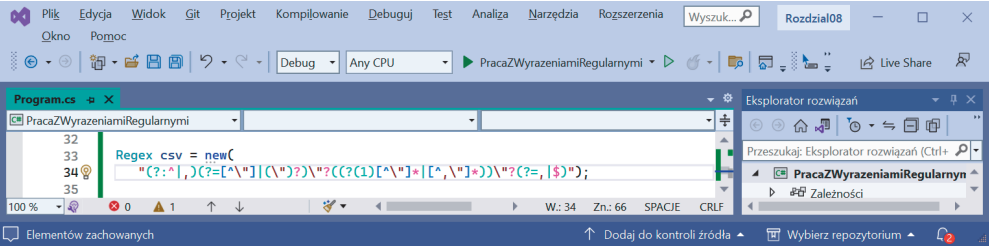
**Rysunek 7.6. Przeglądanie oryginalnego pliku z kodem źródłowym metody Count**

**Rysunek 7.7. Komunikat udanego przesłania pakietu NuGet****Rysunek 7.8. Biblioteka klas zgodna z .NET Standard 2.0 może być używana we wszystkich platformach .NET**

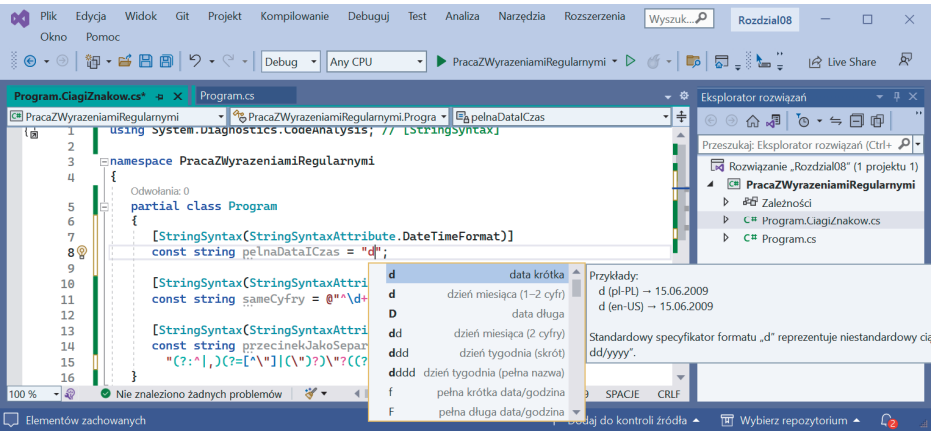


Rysunek 7.9. Przeglądanie danych mojego pakietu za pomocą narzędzia NuGet Package Explorer

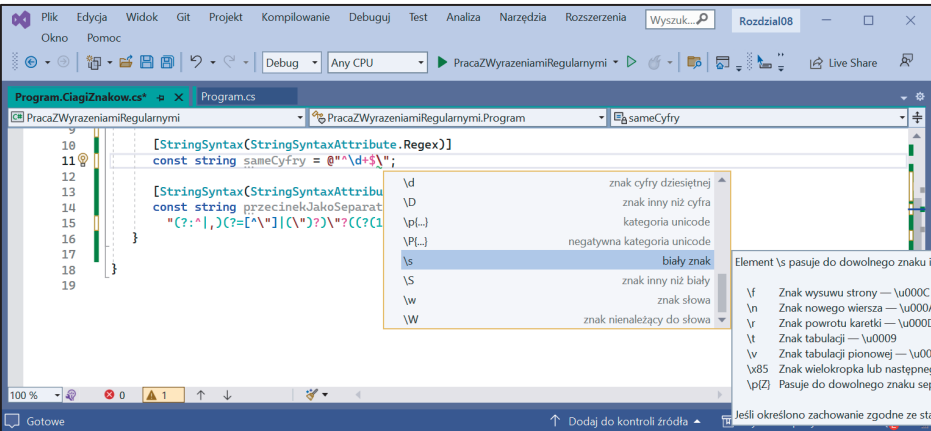
# Rozdział 8. Używanie typów .NET



Rysunek 8.1. Kolorowanie składni wyrażen regularnych w konstruktorze klasy Regex

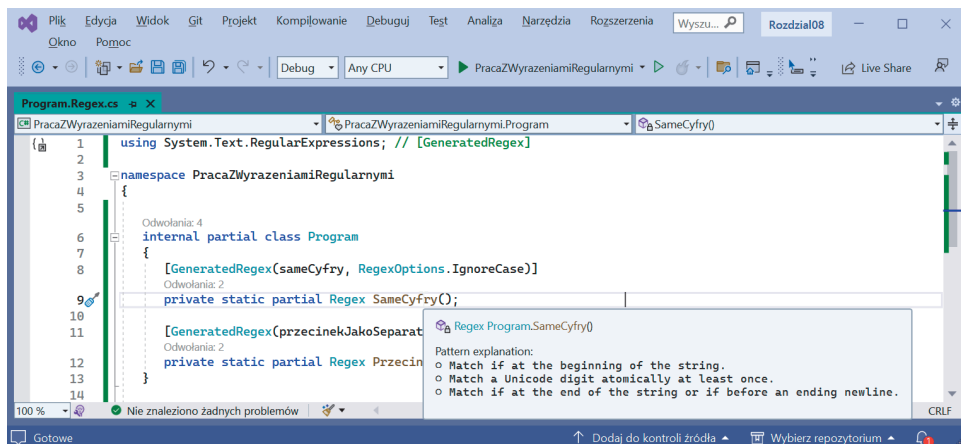


Rysunek 8.2. IntelliSense jest aktywowane przez atrybut StringSyntax

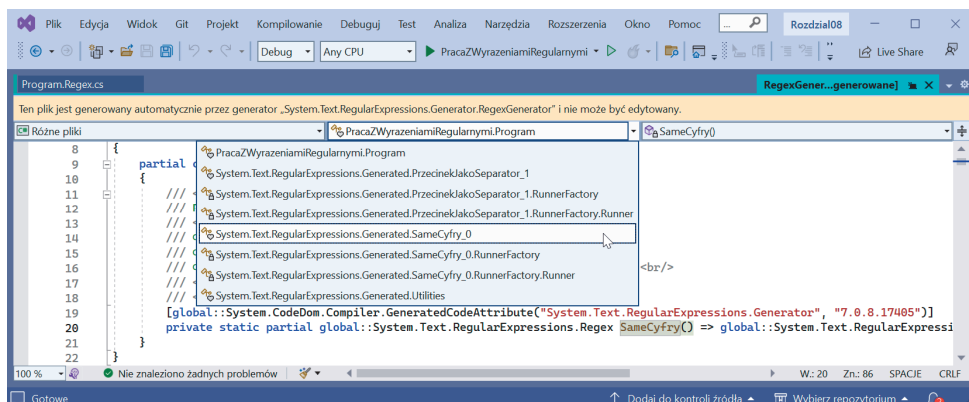


Rysunek 8.3. IntelliSense pomaga tworzyć wyrażenie regularne



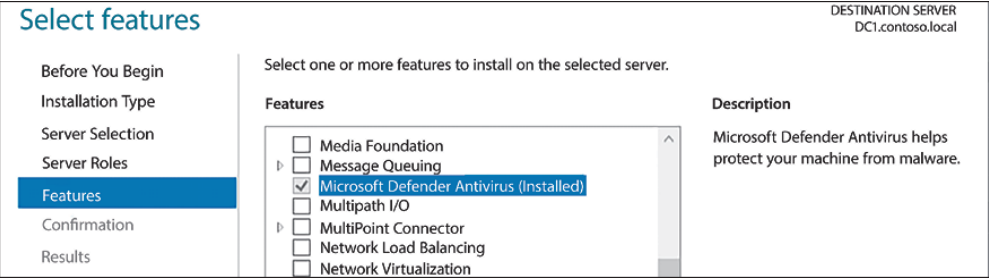


Rysunek 8.4. W podpowiedzi dla metody częściowej podawany jest opis wyrażenia regularnego



Rysunek 8.5. Automatycznie wygenerowany kod źródłowy dla wyrażenia regularnego

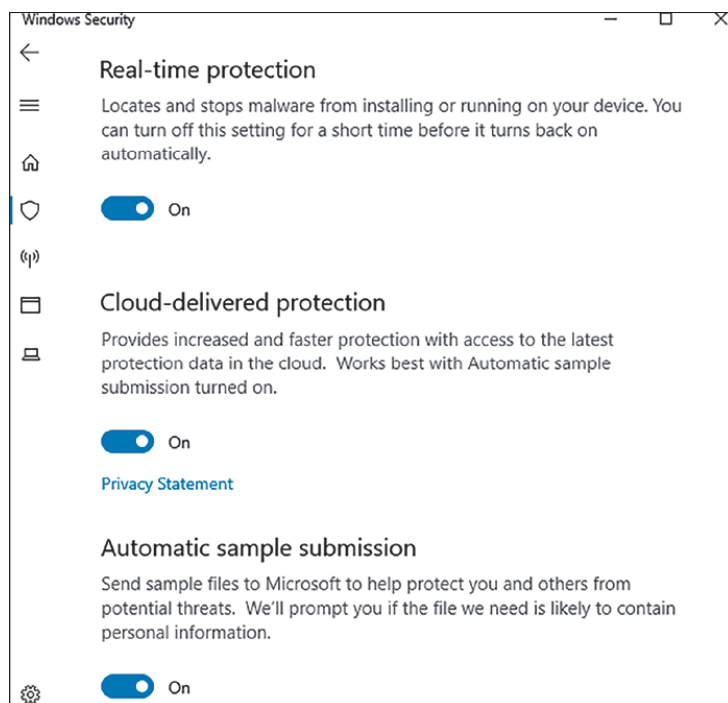
# Rozdział 9. Hardening i bezpieczeństwo



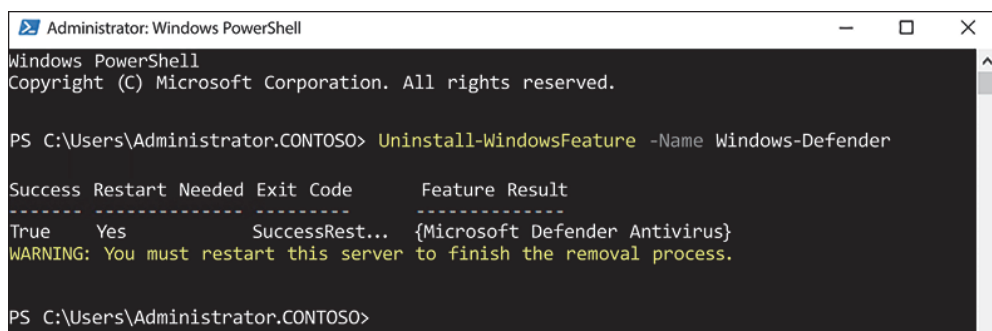
Rysunek 9.1. Program antywirusowy Microsoft Defender jest domyślnie zainstalowany



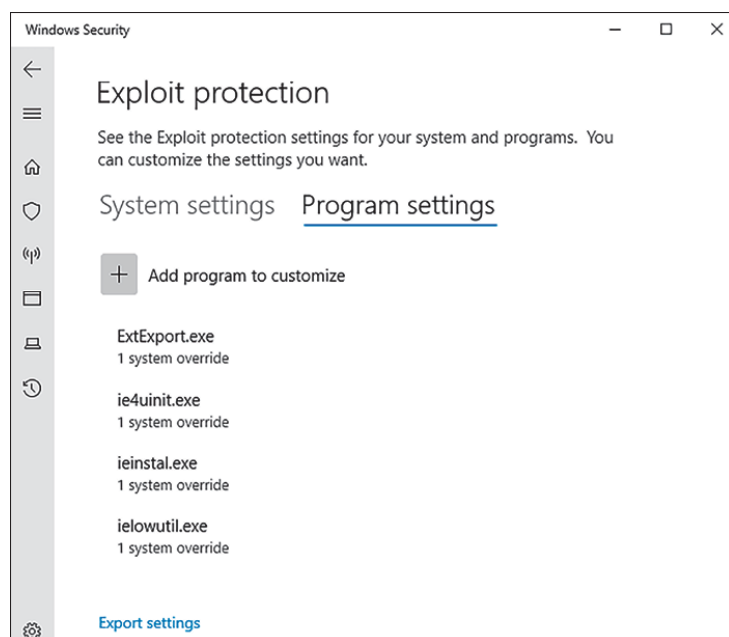
Rysunek 9.2. Funkcje programu Microsoft Defender



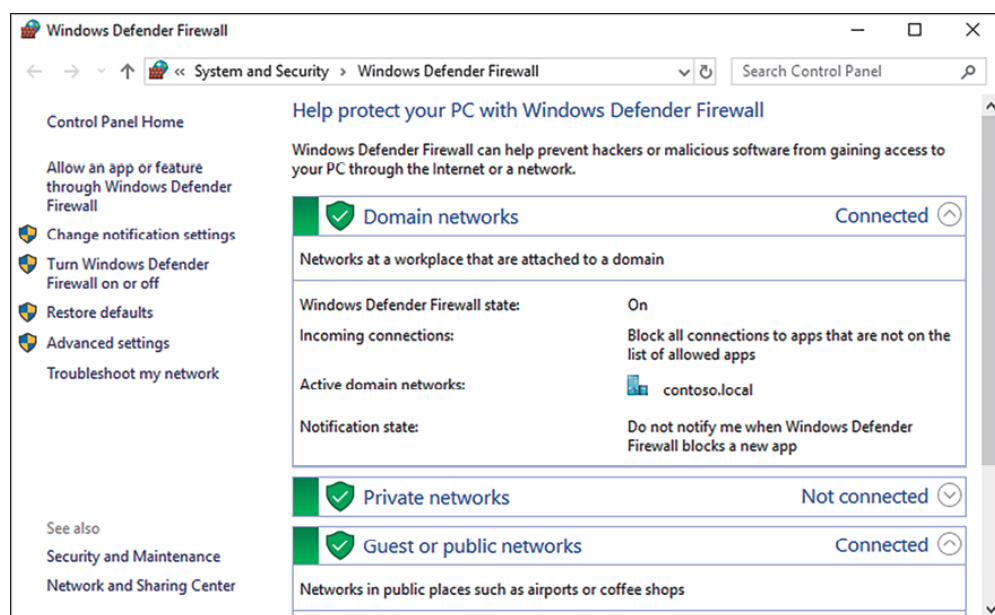
Rysunek 9.3. Opcje ochrony przed wirusami i zagrożeniami



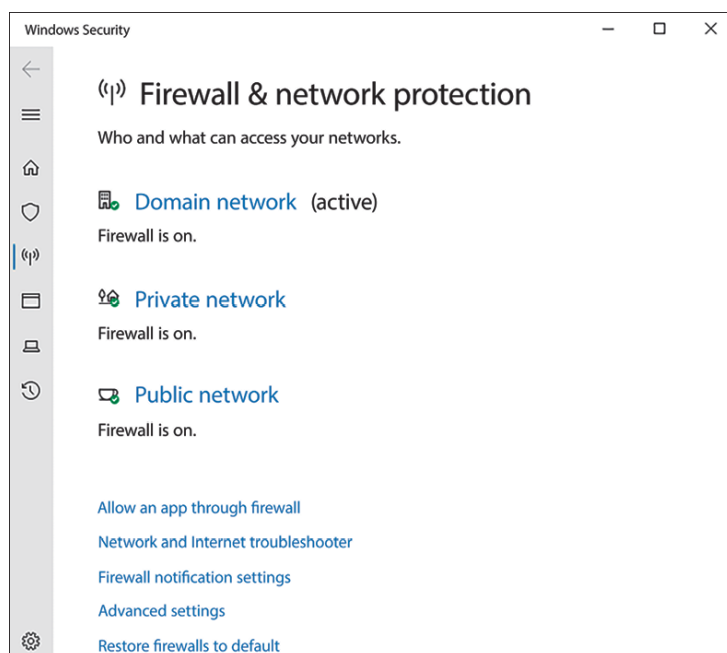
Rysunek 9.4. Dezinstalacja Defendera



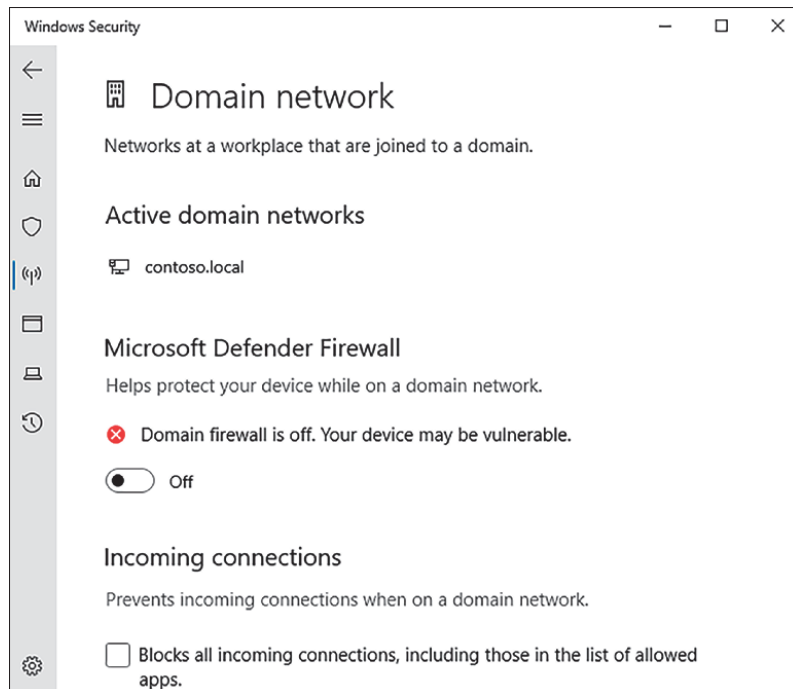
Rysunek 9.5. Ustawienia ochrony przed eksploitantami



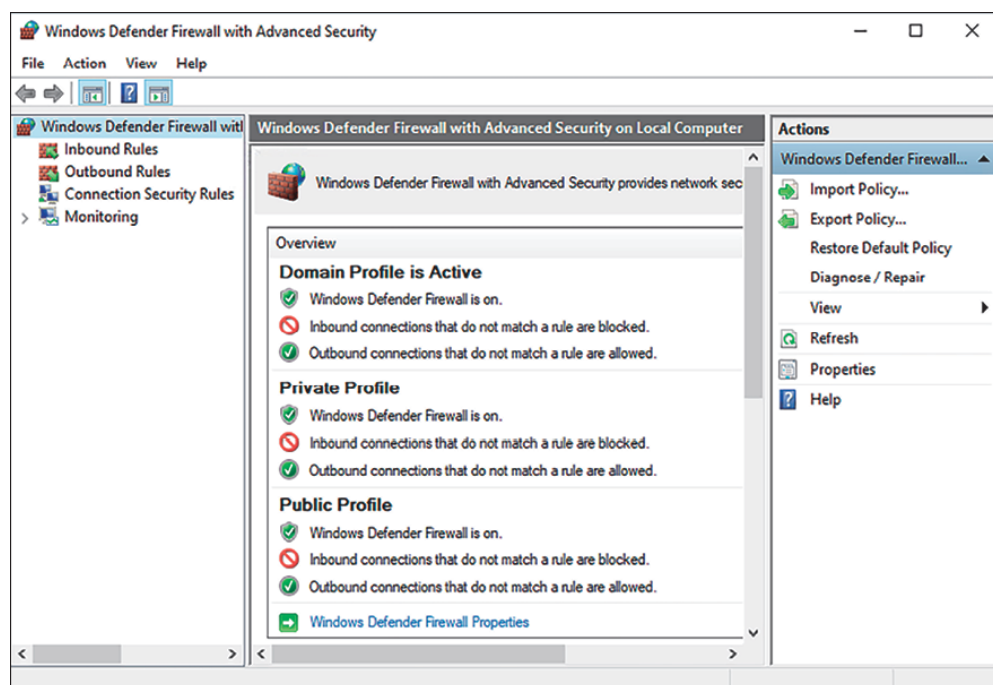
Rysunek 9.6. Ustawienia Zapory w Panelu sterowania



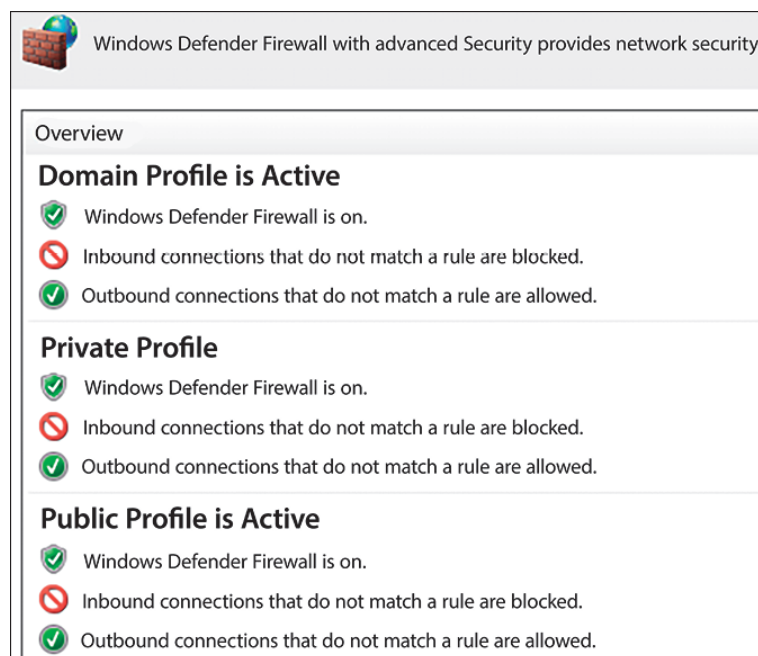
Rysunek 9.7. Ustawienia Zapory w ustawieniach systemu Windows



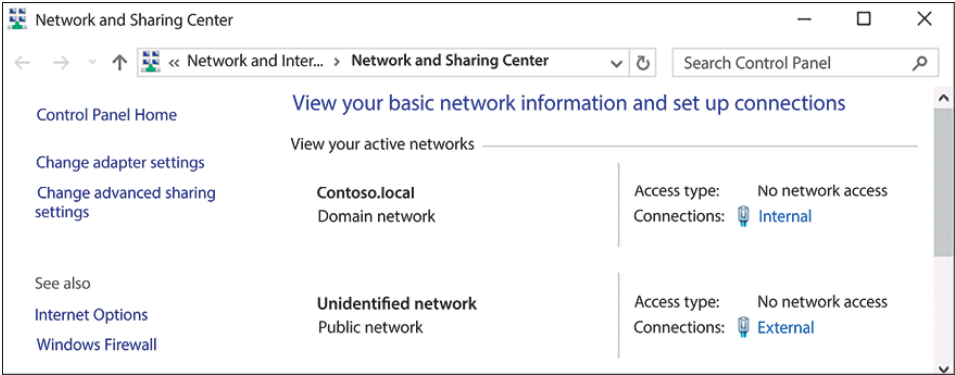
Rysunek 9.8. Wyłączanie Zapory systemu Windows w sieci z domeną



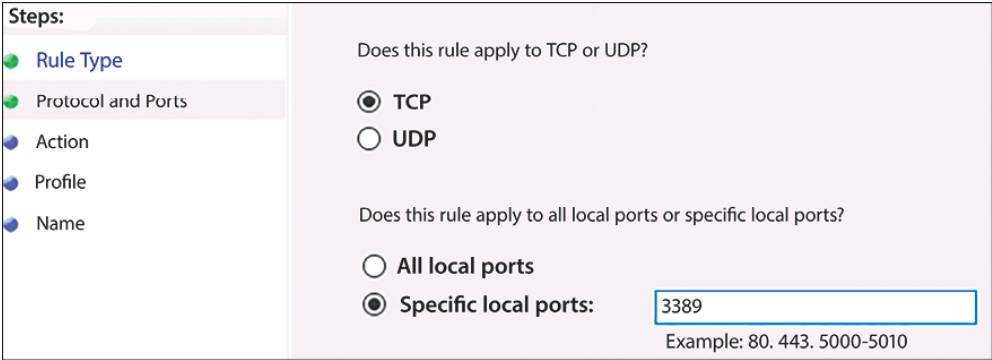
Rysunek 9.9. Zapora systemu Windows z zaawansowanymi ustawieniami zabezpieczeń



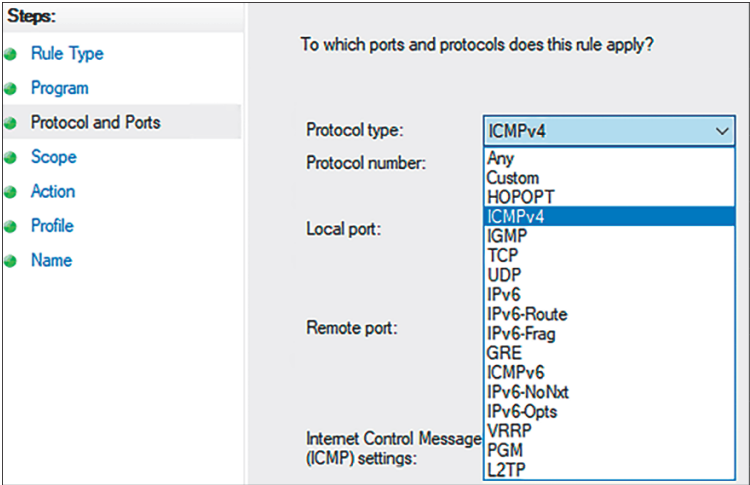
Rysunek 9.10. Wiele kart sieciowych może oznaczać wiele aktywnych profili zapory



Rysunek 9.11. Profile zapory przypisane do każdej karty sieciowej



Rysunek 9.12. Ta reguła zapory będzie manipulowała ruchem na porcie 3389 (RDP)

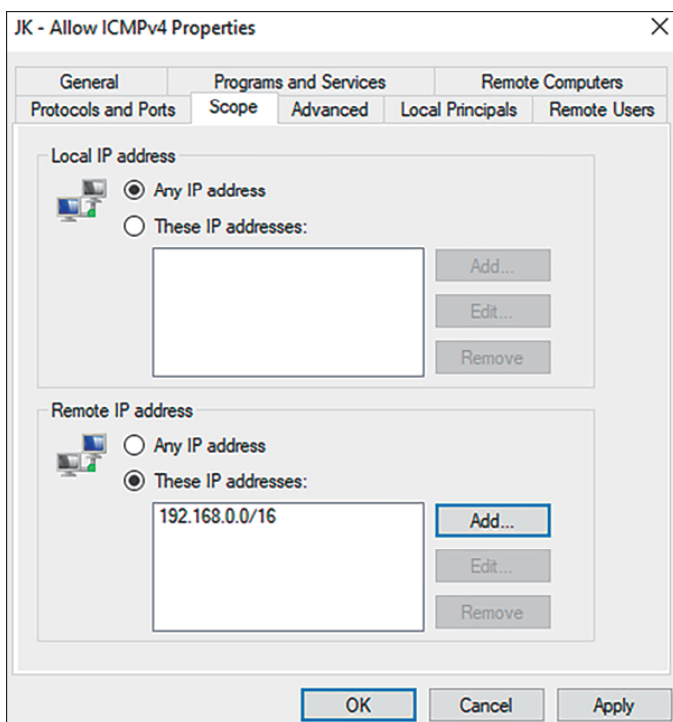


Rysunek 9.15. Stosowanie reguł do określonych protokołów

```
PS C:\Users\administrator.CONTOSO> ping ra1

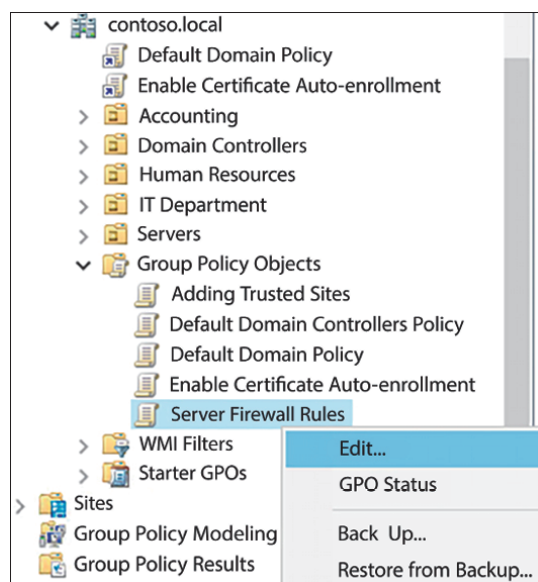
Pinging ra1.contoso.local [10.10.10.13] with 32 bytes of data:
Reply from 10.10.10.13: bytes=32 time<1ms TTL=128
Reply from 10.10.10.13: bytes=32 time<1ms TTL=128
Reply from 10.10.10.13: bytes=32 time<1ms TTL=128
Reply from 10.10.10.13: bytes=32 time<1ms TTL=128
```

Rysunek 9.16. Wysyłanie sygnału ping do serwera i odbiór odpowiedzi ICMP

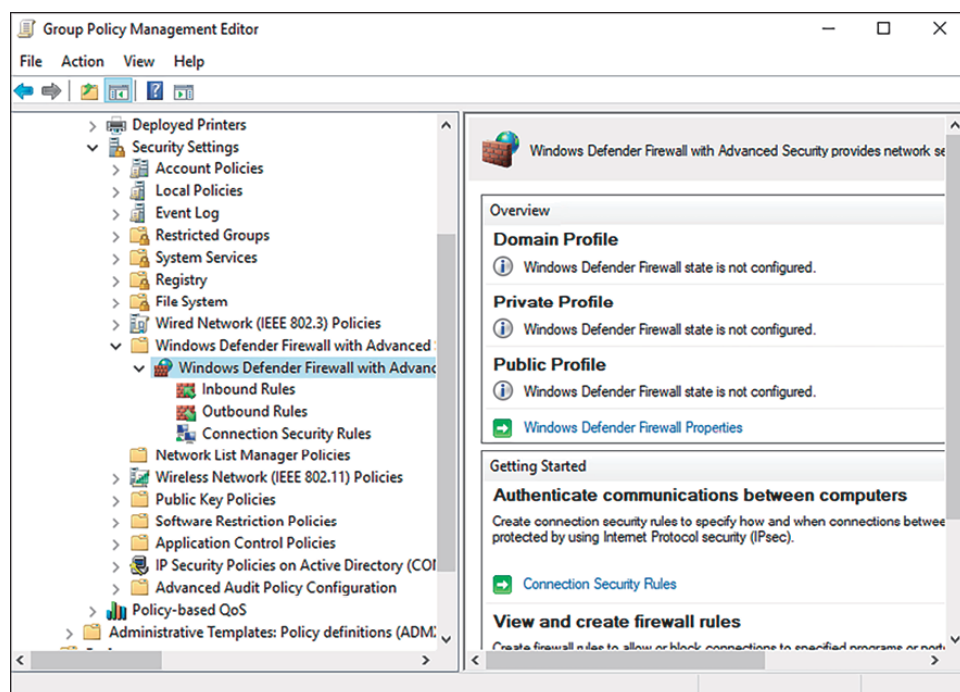


Rysunek 9.17. Ograniczanie zakresu reguły zapory do wybranych adresów IP lub podsieci





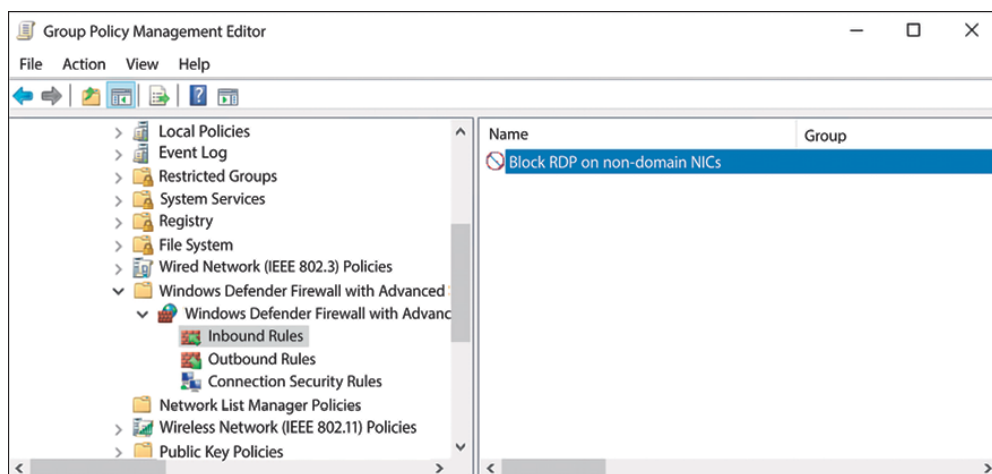
**Rysunek 9.18. Centralne zarządzanie regułami zapory systemu Windows za pomocą zasad grupy**



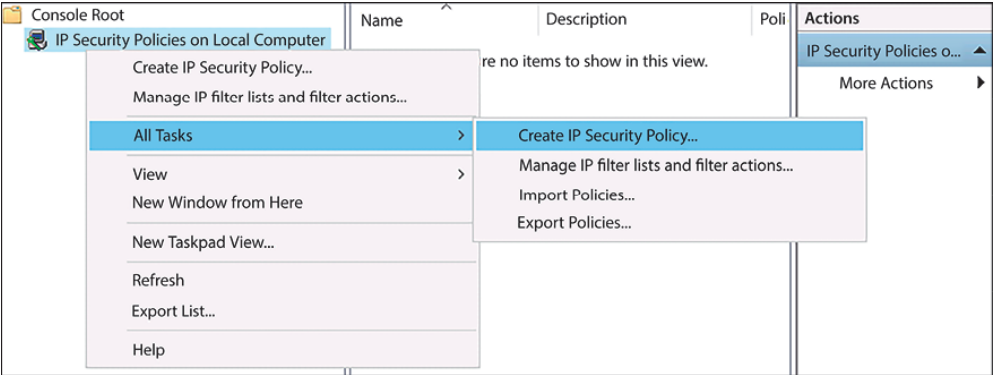
**Rysunek 9.19. Lokalizacja ustawień reguł zapory w GPO**



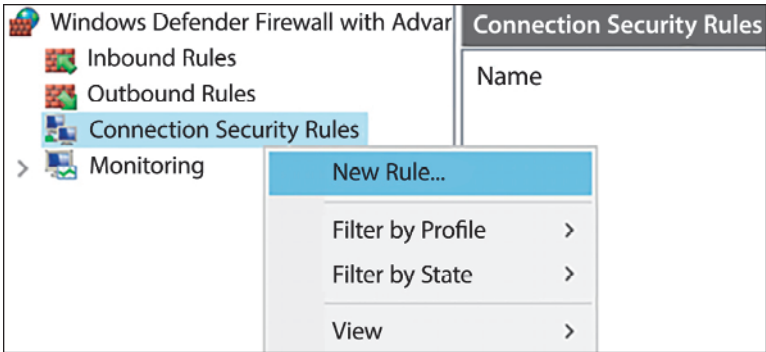
Rysunek 9.20. Osobne karty dla ustawień każdego z profili zapory



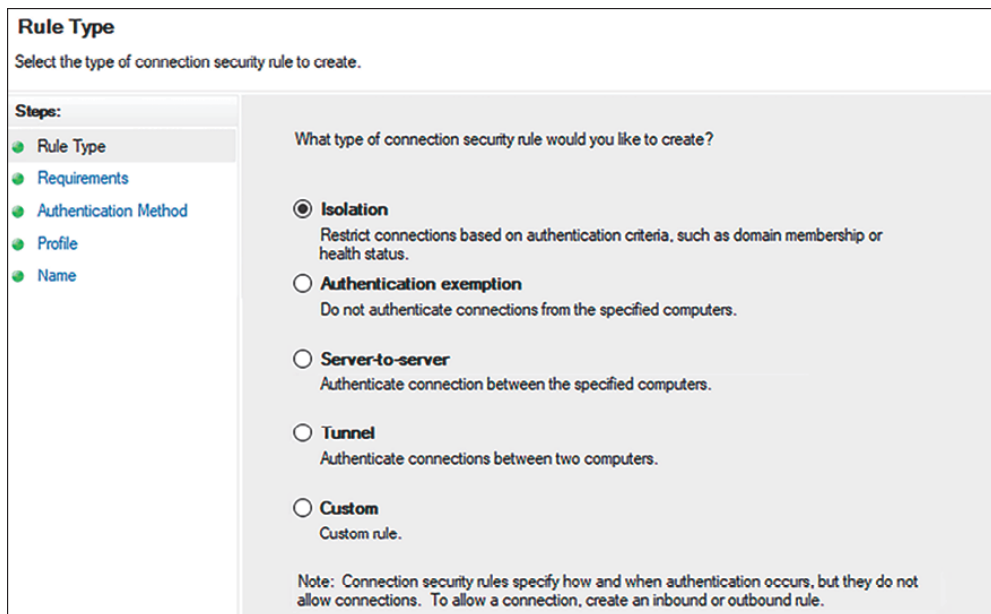
Rysunek 9.21. Pomyślnie zaimplementowana reguła zapory w GPO



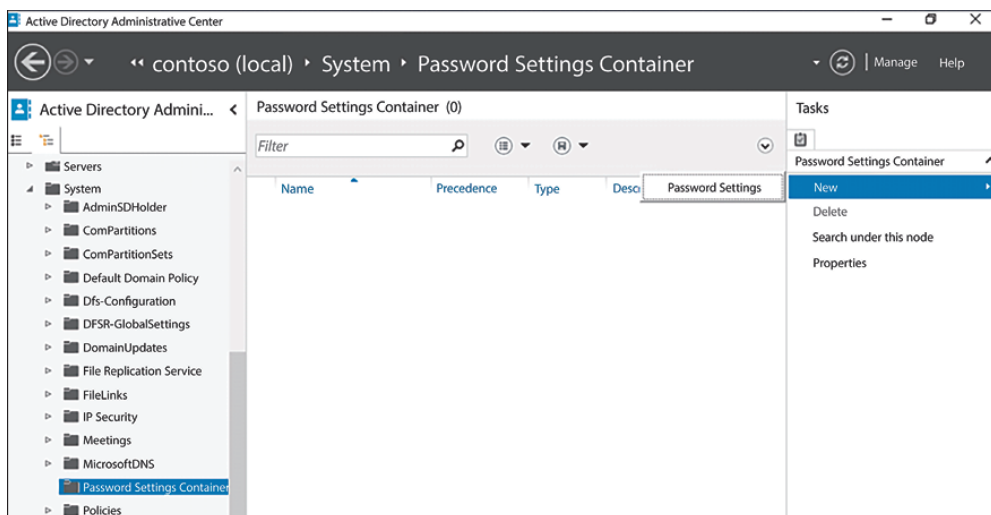
Rysunek 9.23. Tworzenie zasady zabezpieczeń IP



Rysunek 9.24. Za pomocą WFAS można także tworzyć zasady zabezpieczeń IP

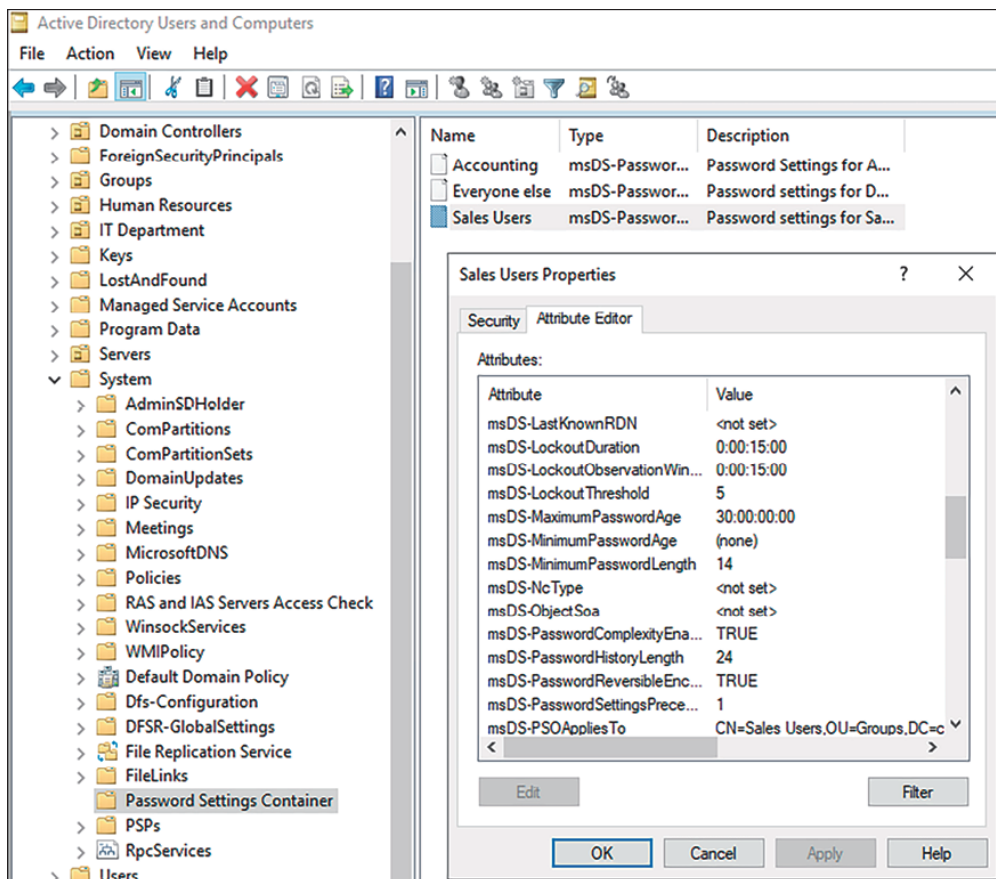


Rysunek 9.25. Typy zasad zabezpieczeń połączeń

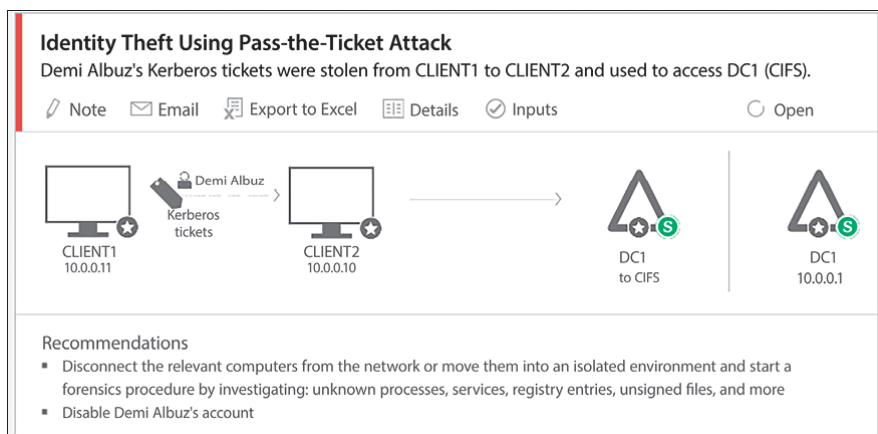


Rysunek 9.26. Szczegółowe zasady haseł konfiguruje się w ADAC





Rysunek 9.29. Przeglądanie kryteriów haseł poza ADAC



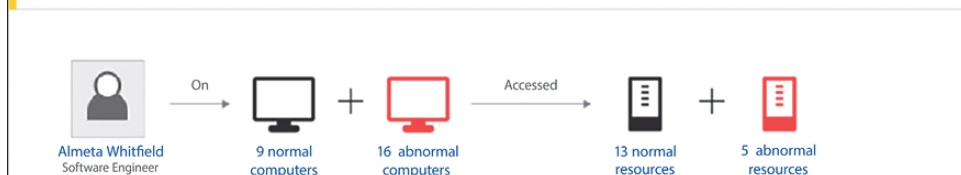
Rysunek 9.30. Usługa ATA wykryła atak typu pass-the-ticket

## Suspicion of identity theft based on abnormal behavior

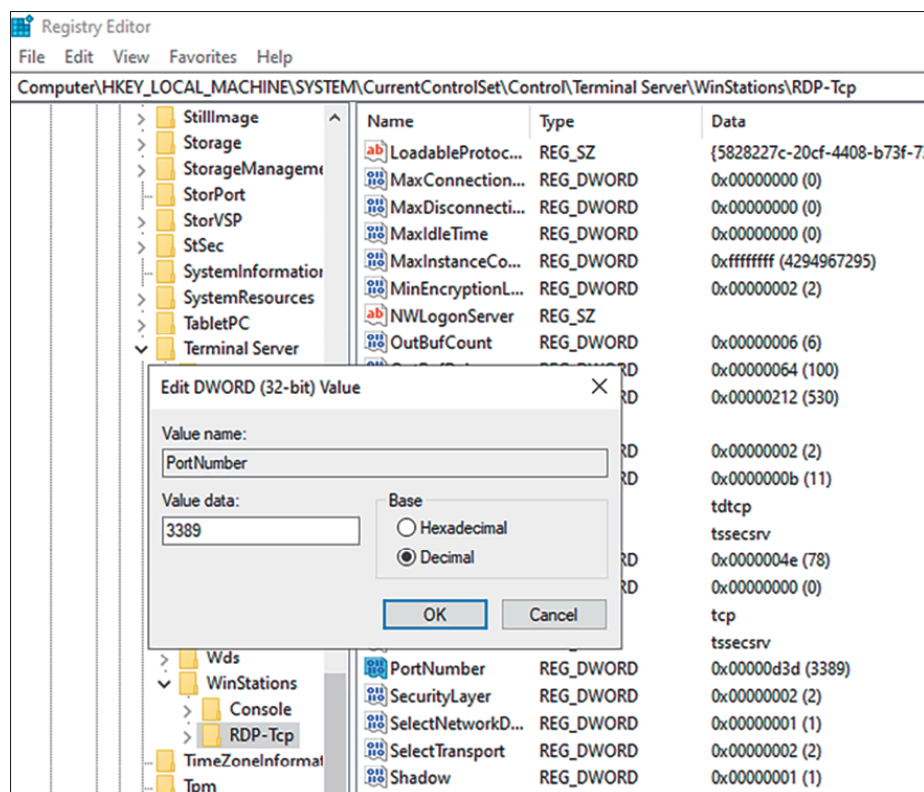
OPEN

Almeta Whitfield exhibited abnormal behavior when performing activities that were not seen over the last month and are also not in accordance with the activities of other accounts in the organization. The abnormal behavior is based on the following activities:

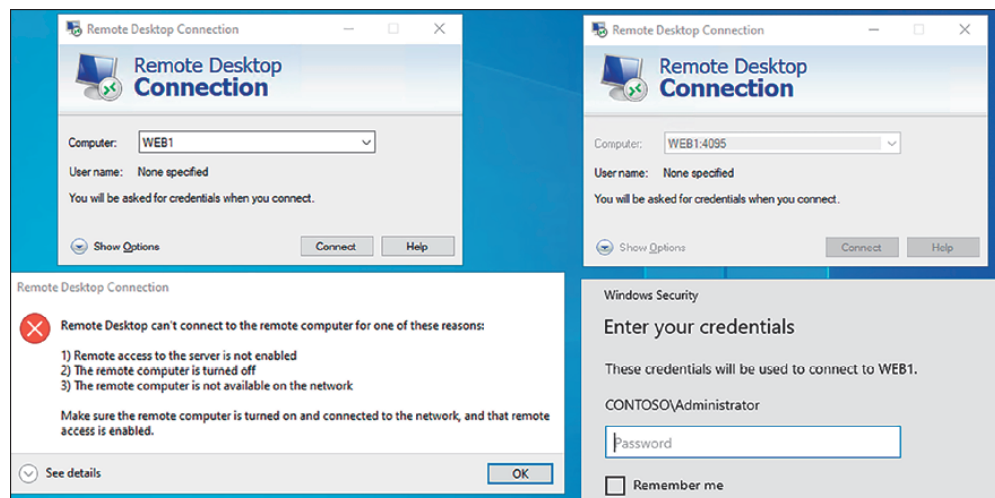
- Performed interactive login from 16 abnormal workstations.
- Requested access to 5 abnormal resources.



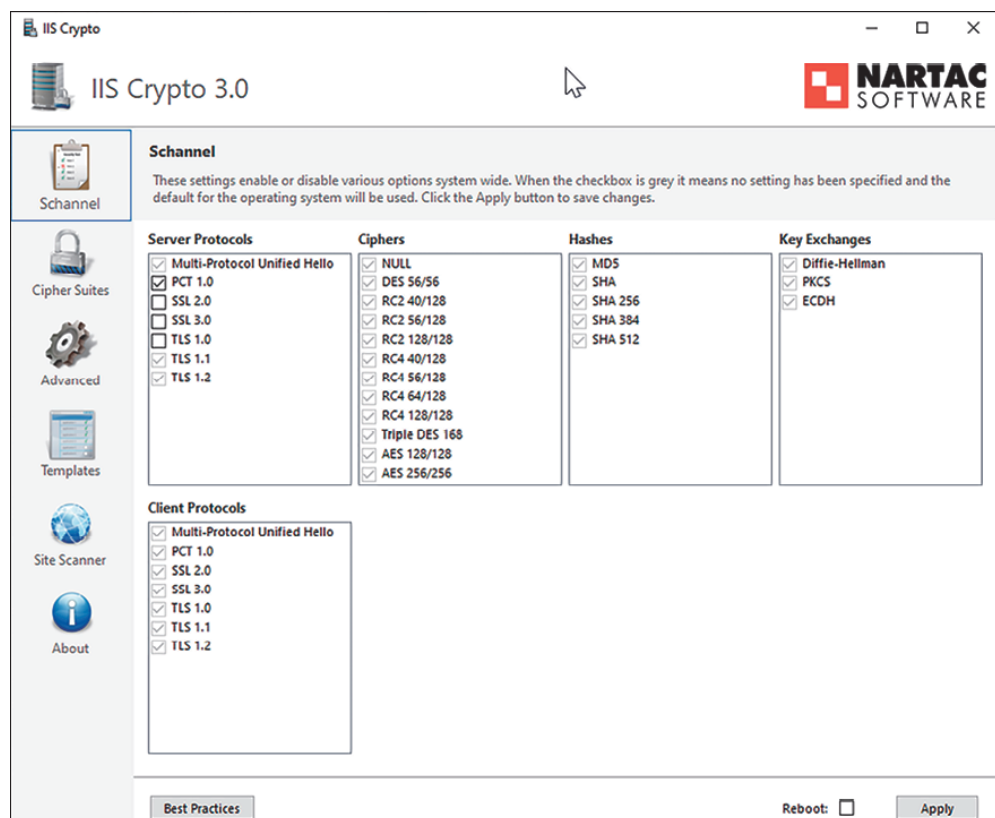
Rysunek 9.31. ATA ostrzeżenia o nietypowym zachowaniu



Rysunek 9.32. Aktualizacja numeru portu RDP w rejestrze



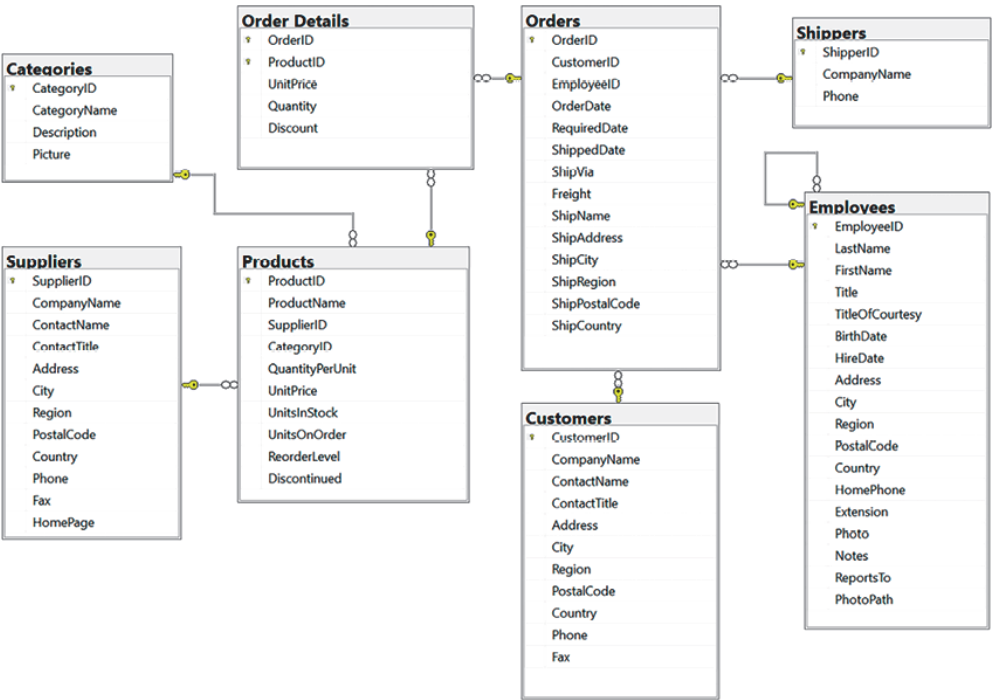
Rysunek 9.33. Teraz serwer WEB1 nasłuchuje na porcie 4095



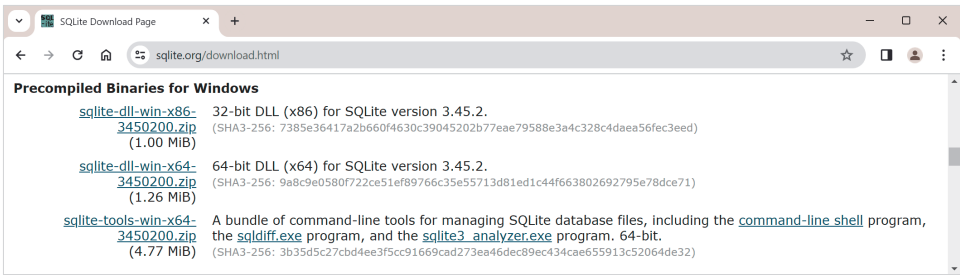
Rysunek 9.34. Program IIS Crypto ułatwia zwiększanie poziomu bezpieczeństwa protokołów



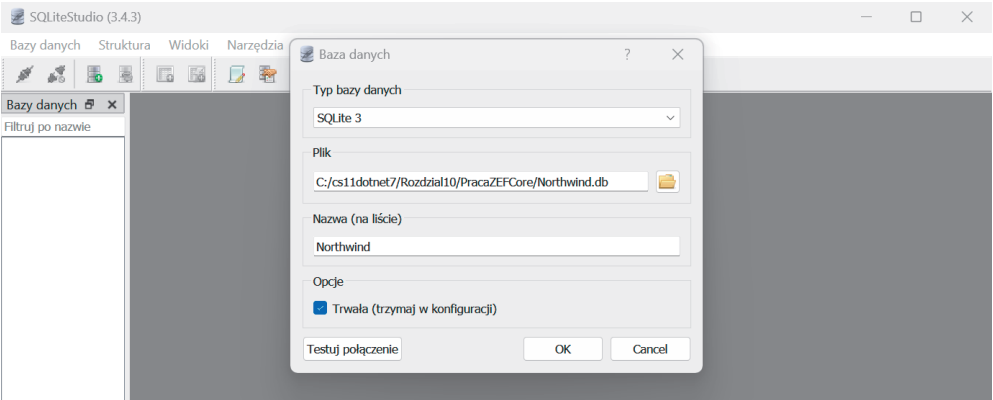
# Rozdział 10. Praca z bazami danych przy użyciu Entity Framework Core



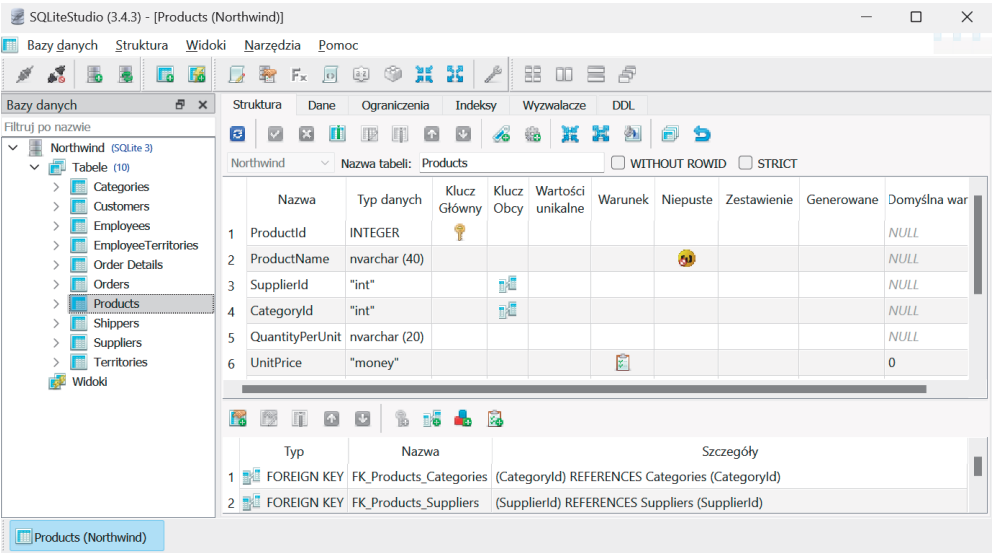
Rysunek 10.1. Tabele i relacje bazy danych Northwind



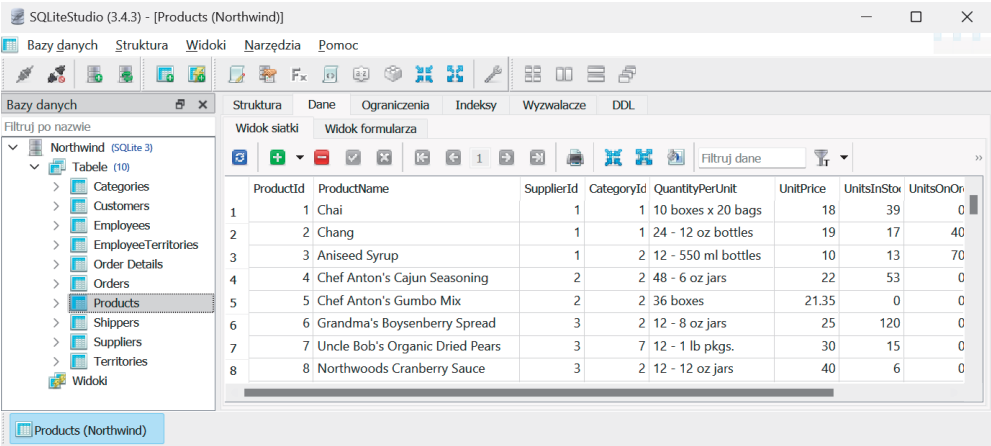
Rysunek 10.2. Pobieranie SQLite dla systemu Windows



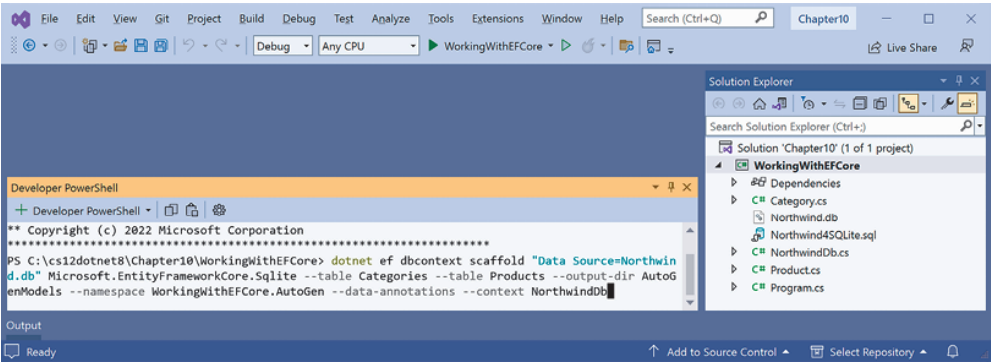
Rysunek 10.3. Dodawanie bazy danych Northwin w SQLiteStudio



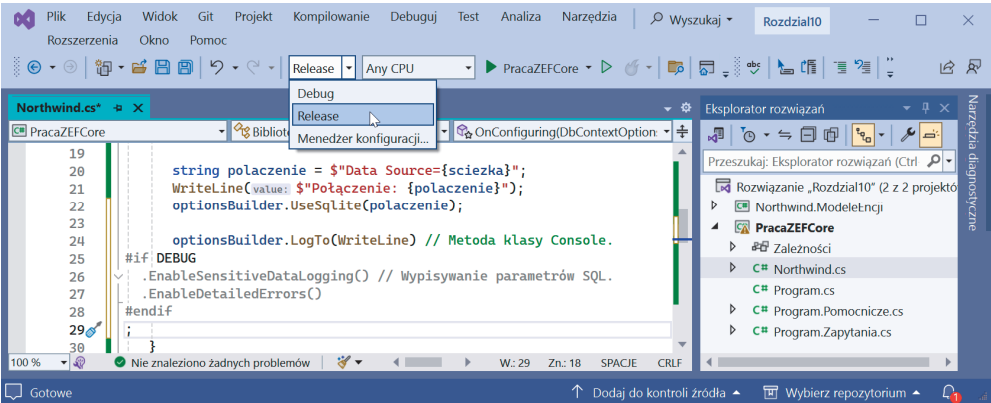
Rysunek 10.4. Edytor tabeli w SQLiteStudio wyświetlający strukturę tabeli Products



Rysunek 10.5. Karta Dane wyświetlająca wiersze danych z tabeli Products

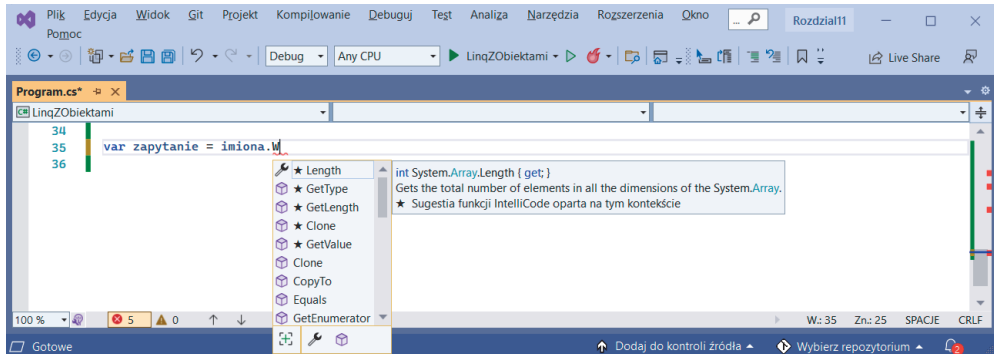


Rysunek 10.6. Wprowadzanie polecenia dotnet-ef w terminalu, w Visual Studio 2022

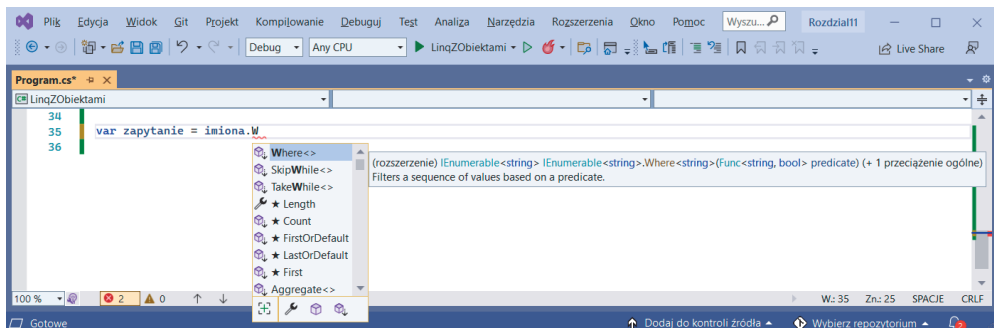


Rysunek 10.7. Dołączanie parametrów SQL do protokołu jedynie w konfiguracji Debug

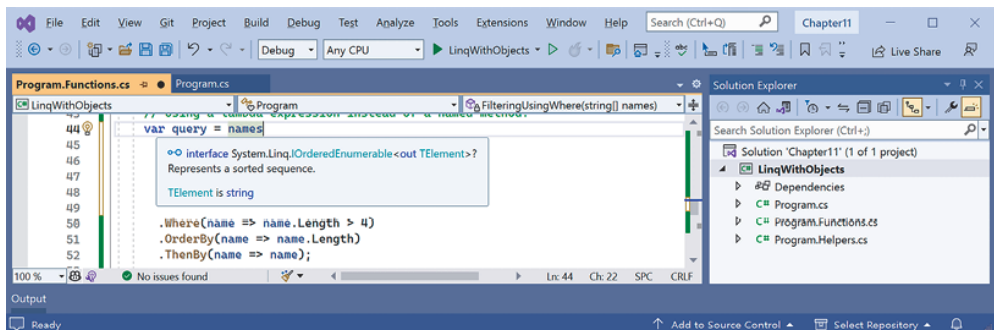
# Rozdział 11. Odczytywanie danych i manipulowanie nimi za pomocą LINQ



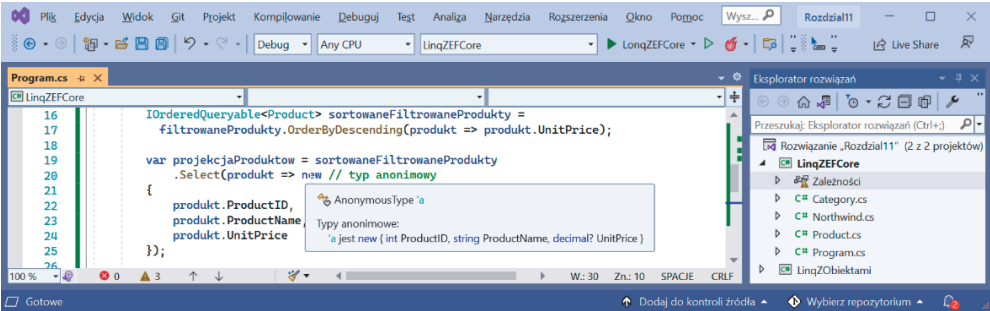
Rysunek 11.1. Okienko IntelliSense, w którym brakuje metody Where



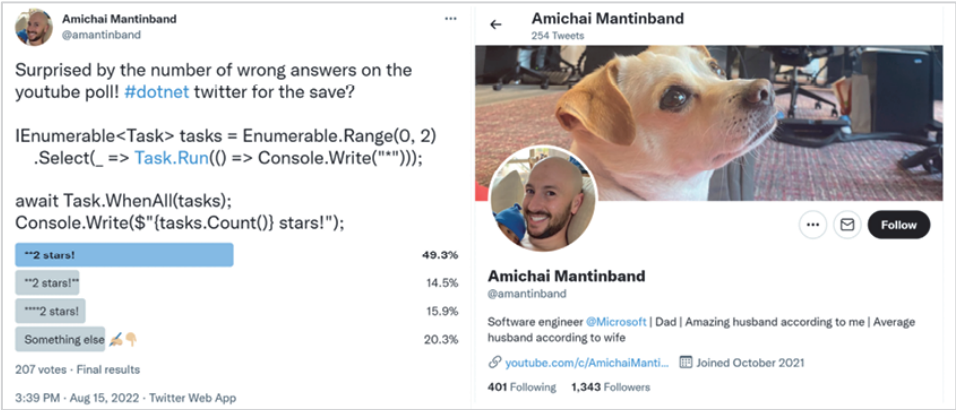
Rysunek 11.2. Okienko IntelliSense wyświetlające też metody rozszerzające z klasy Enumerable



Rysunek 11.3. Umieszczenie wskaźnika myszy na słowie var ujawnia wywiedziony typ wyrażenia

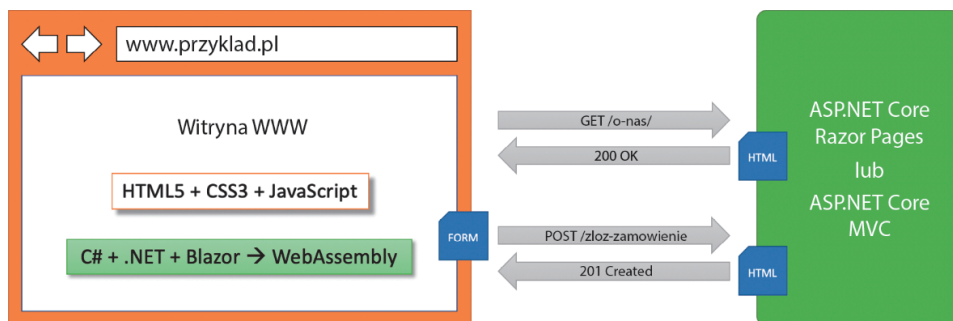


Rysunek 11.5. Typ anonimowy używany w projekcji LINQ

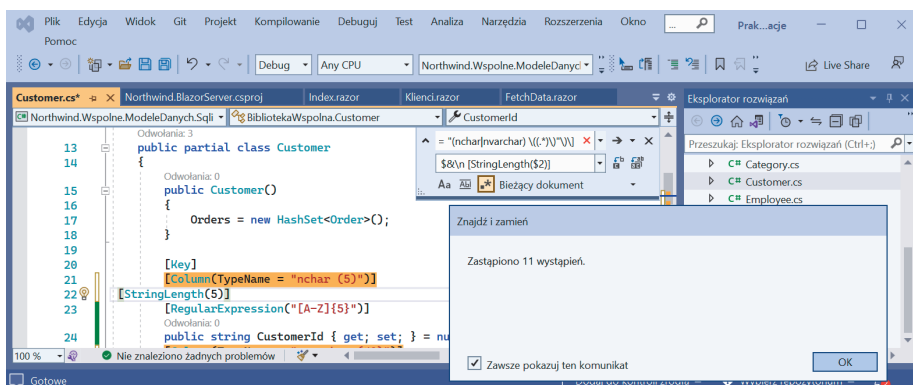


Rysunek 11.6. Ciekawy kod z LINQ i zadaniami zaprezentowany przez Amichaia Mantinbanda

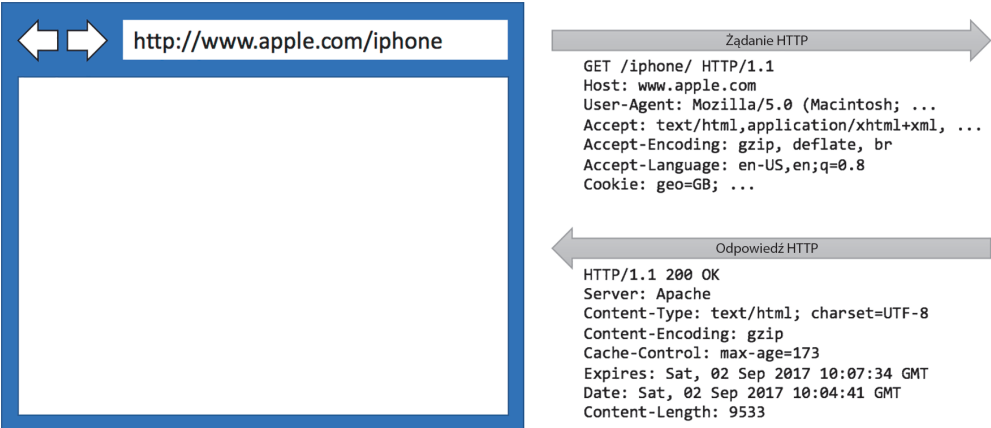
## Rozdział 12. Wprowadzenie do aplikacji sieciowych w ASP.NET Core



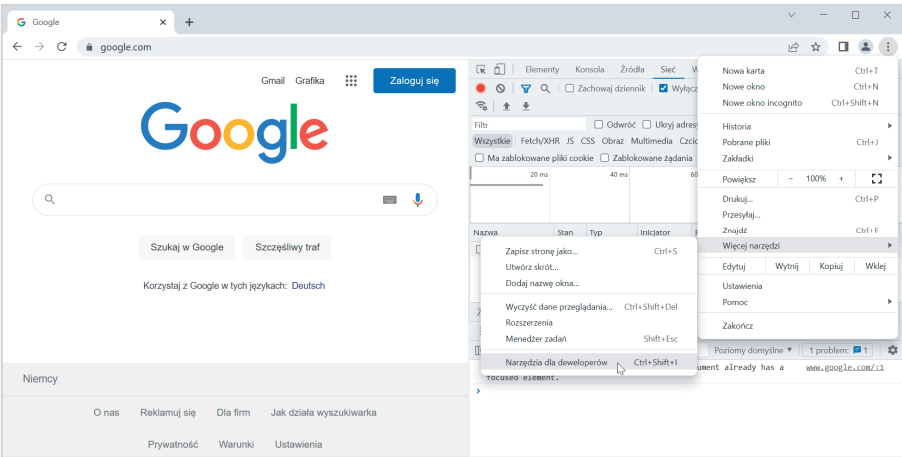
Rysunek 12.1. Sposób użycia języka C# i środowiska .NET do tworzenia serwisów sieciowych po stronie serwera i po stronie klienta



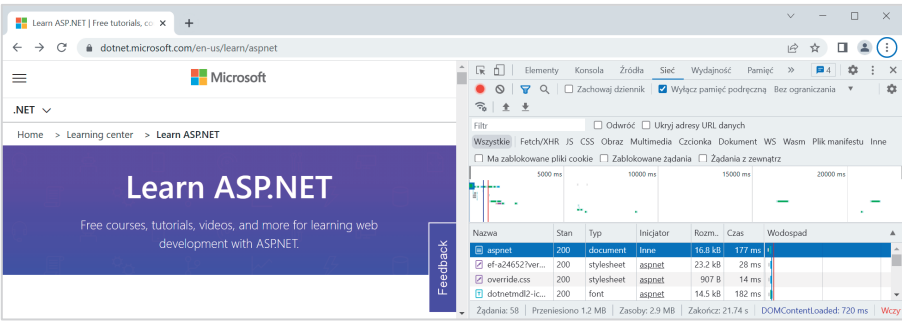
Rysunek 12.2. Wyszukaj i zamień wszystkie dopasowania wyrażenia regularnego w Visual Studio 2022



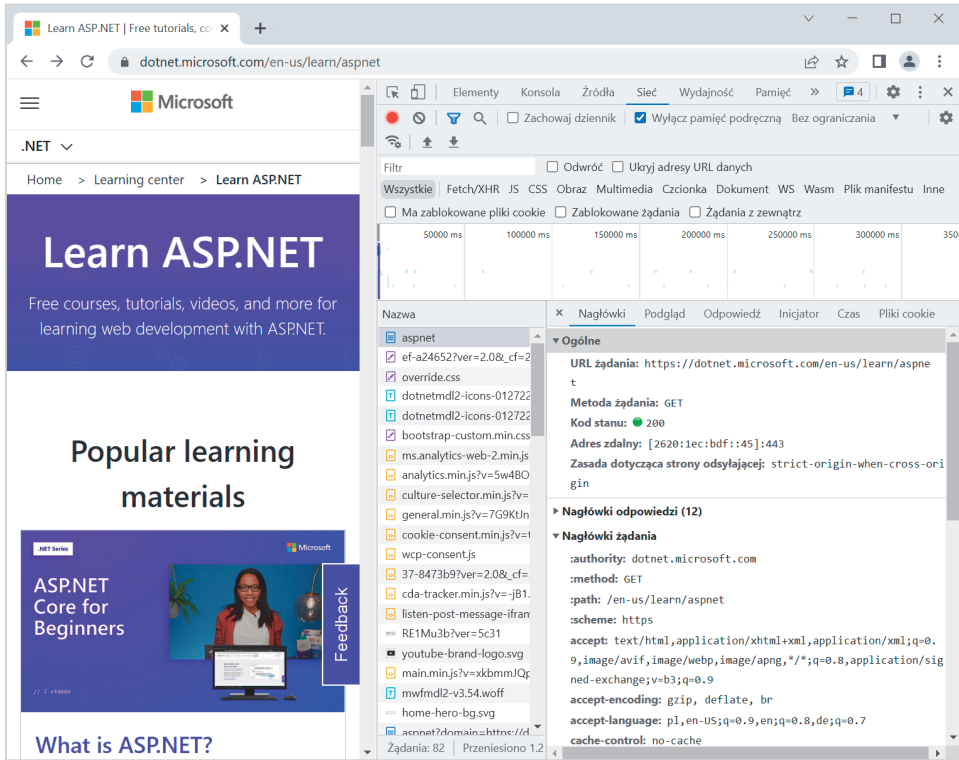
Rysunek 12.3. Żądanie i odpowiedź HTTP



Rysunek 12.4. Narzędzia dla developerów w Google Chrome zapisują dane ruchu sieciowego



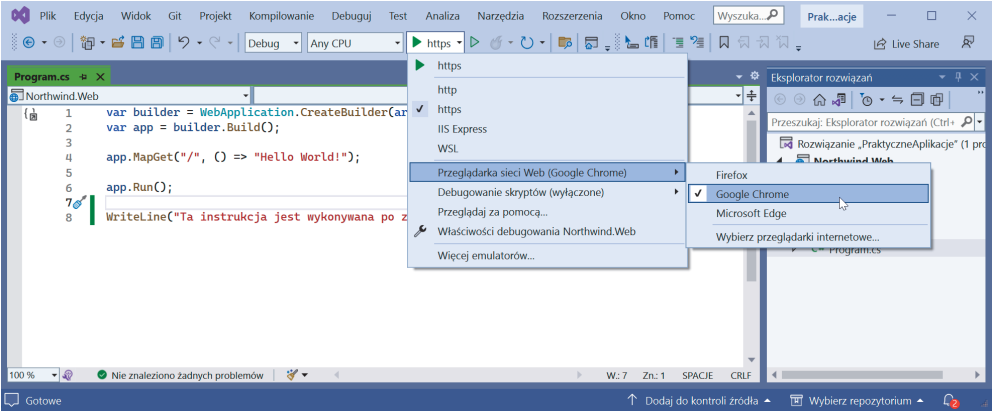
Rysunek 12.5. Zapisane żądanie wyświetlone w oknie narzędzi dla developerów



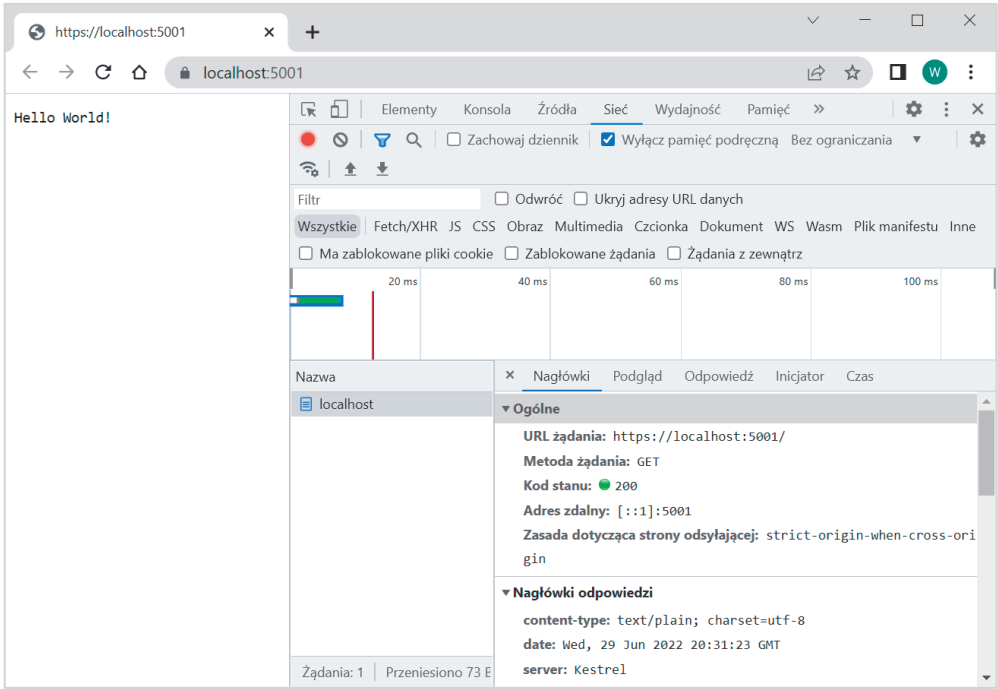
Rysunek 12.6. Nagłówki żądania i odpowiedzi



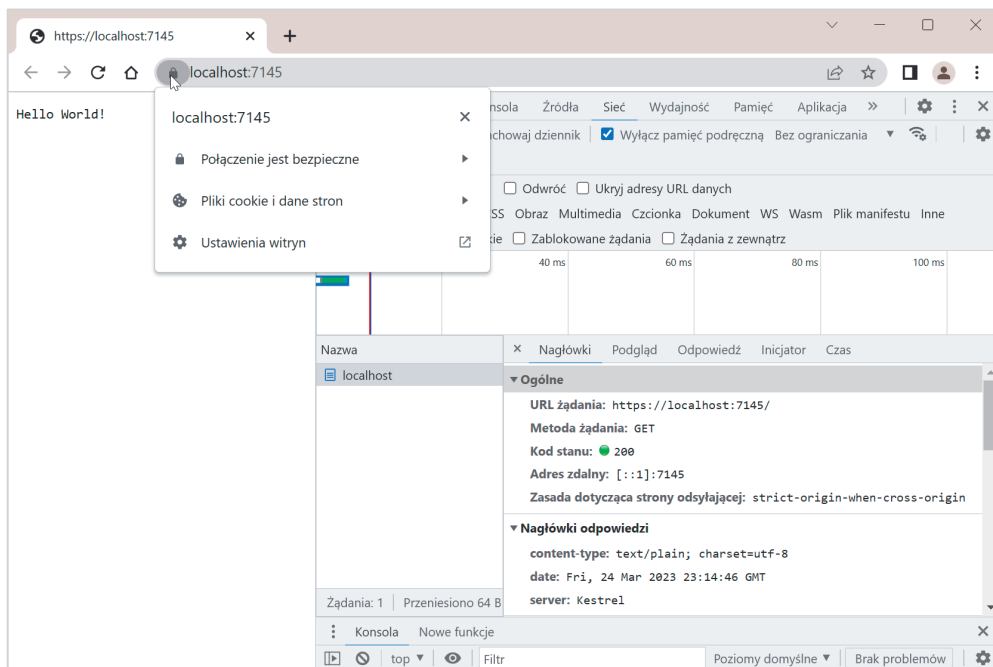
# Rozdział 13. Tworzenie witryn WWW przy użyciu ASP.NET Core Razor Pages



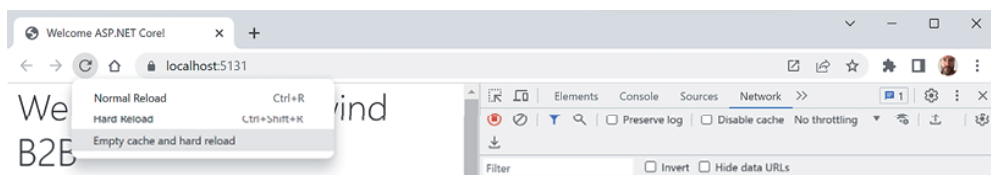
Rysunek 13.1. Wybieranie profilu Northwind.Web z serwerem Kestrel w Visual Studio



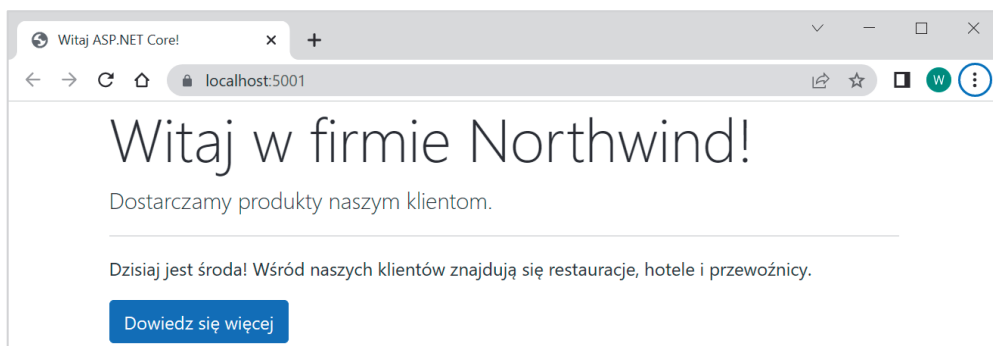
Rysunek 13.2. Odpowiedź jawnym tekstem z adresu *http://localhost:5001/*



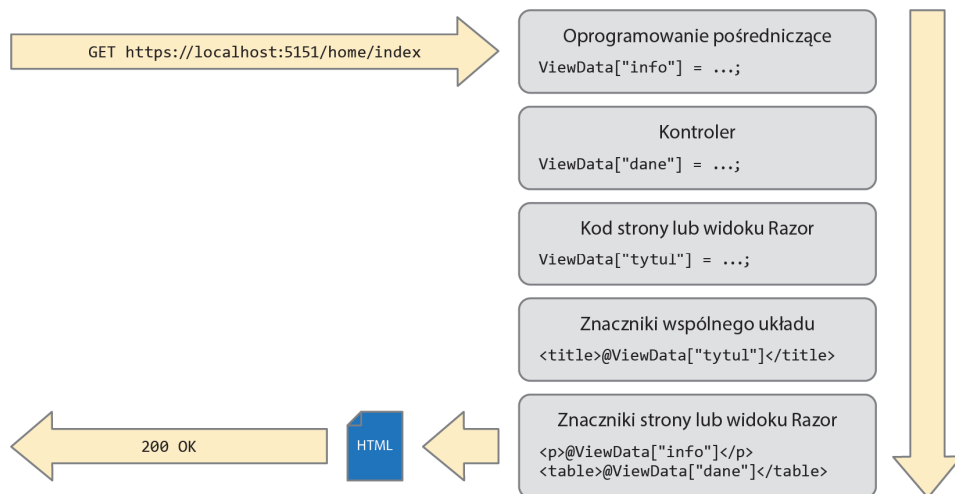
**Rysunek 13.3. Połączenie jest teraz zabezpieczone certyfikatem i przekierowaniem http 307**



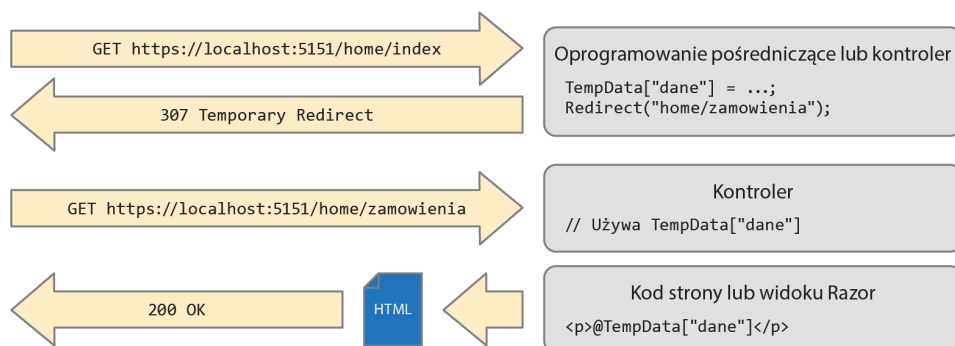
**Rysunek 13.4. Wywołaj narzędzia dla deweloperów, a potem kliknij i przytrzymaj przycisk Załaduj tę stronę ponownie, aby zobaczyć dodatkowe polecenia**



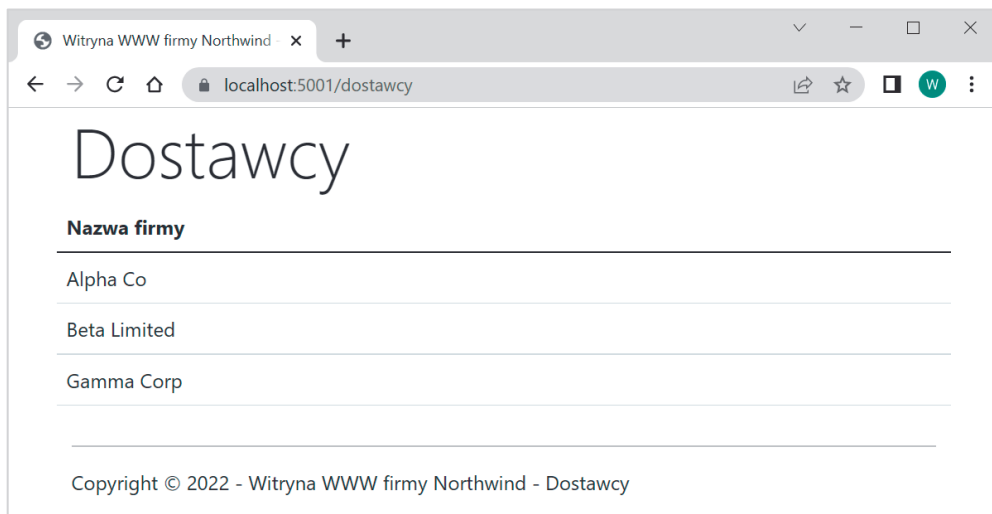
**Rysunek 13.5. Strona Witaj w firmie Northwind wyświetla nazwę dnia**



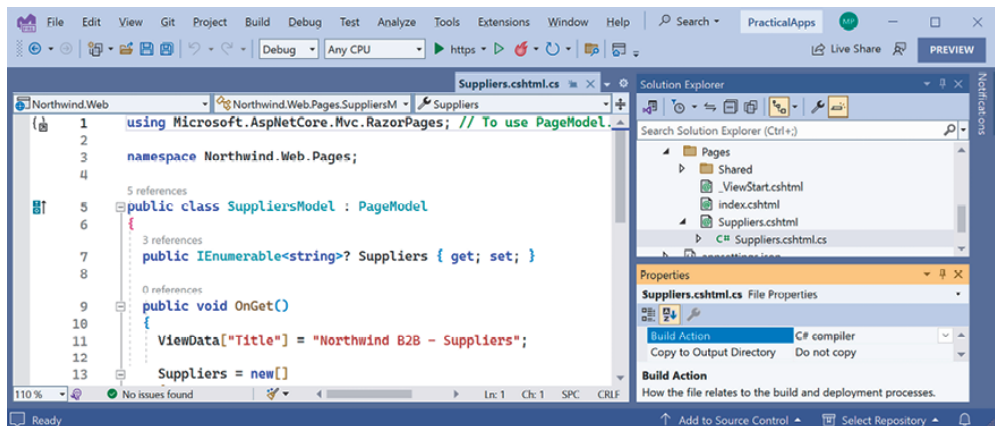
**Rysunek 13.6. Użycie słownika ViewData do przekazywania danych w ramach jednego żądania**



**Rysunek 13.7. Użycie słownika TempData do przekazywania informacji między dwoma żądaniami**



Rysunek 13.8. Tabela dostawców z danymi załadowanymi z tablicy ciągów znaków



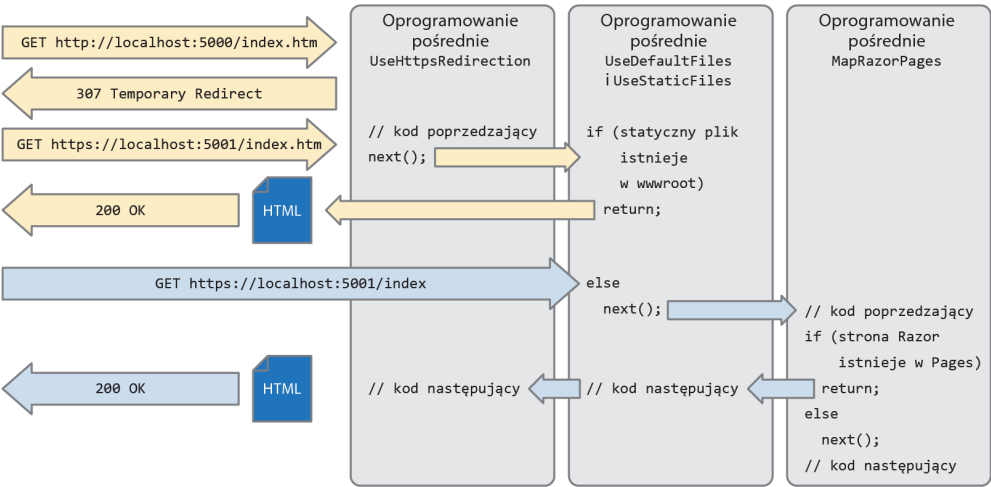
Rysunek 13.9. Okienko Właściwości pliku Dostawcy.cshtml wskazuje, że jego domyślną akcją kompilacji jest Kompilator C#

Witryna WWW firmy Northwind

localhost:5001/dostawcy

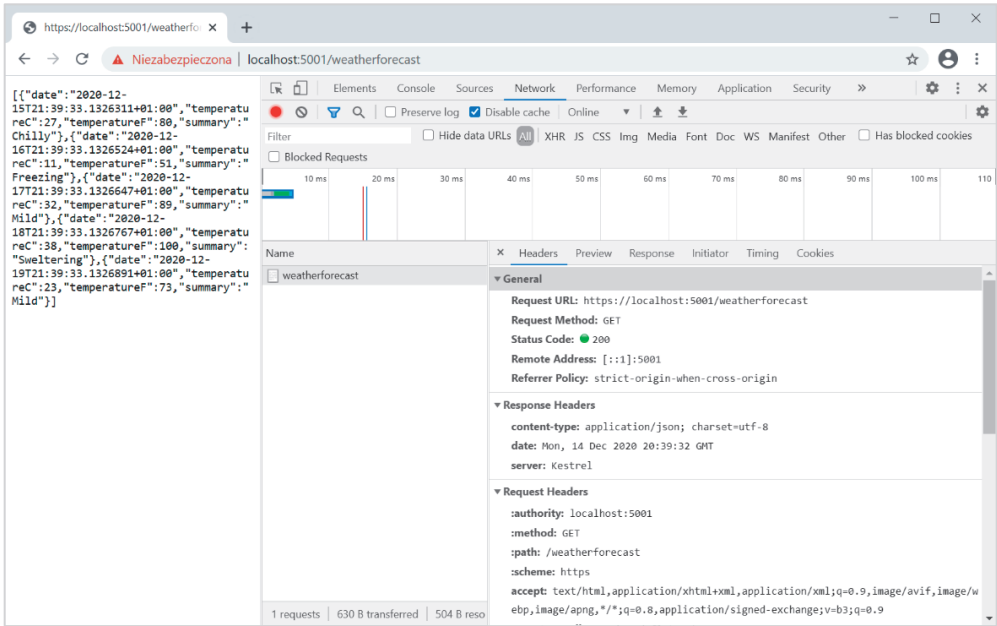
Dostawcy		
Nazwa firmy	Kraj	Telefon
G'day, Mate	Australia	(02) 555-5914
Pavlova, Ltd.	Australia	(03) 444-2343
Refrescos Americanas LTDA	Brazil	(11) 555 4640
Forêts d'érables	Canada	(514) 555-2955

Rysunek 13.10. Tabela dostawców z danymi pobranymi z bazy danych Northwind

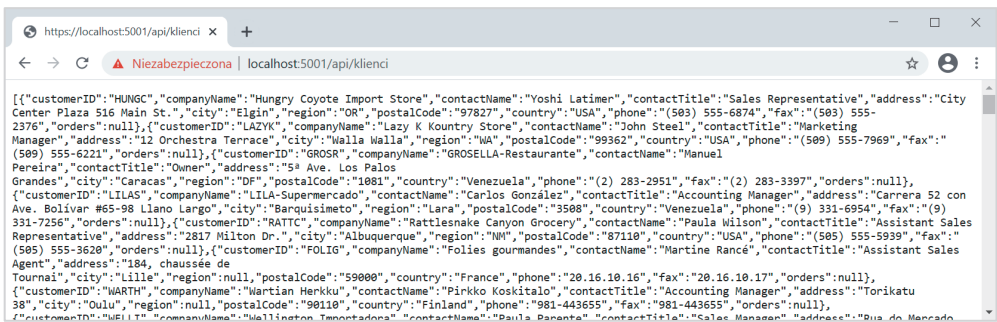


Rysunek 13.11. Potok żądań i odpowiedzi HTTP

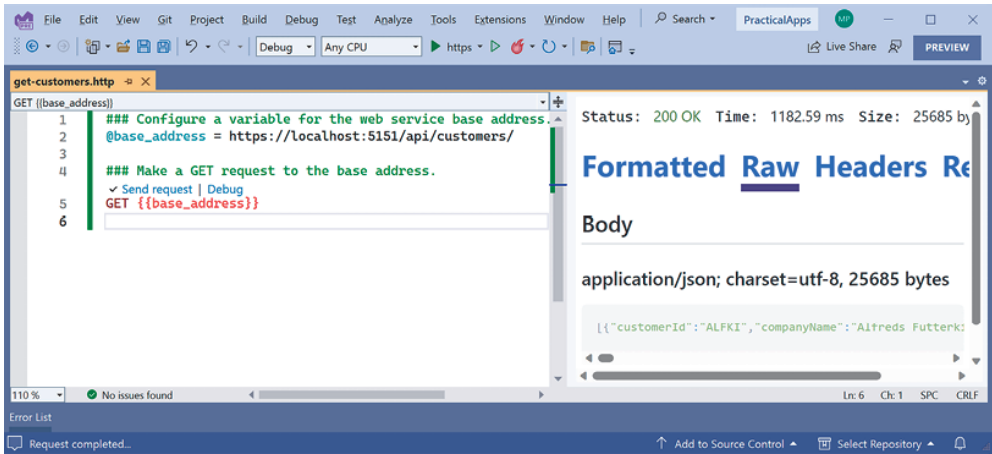
# Rozdział 14. Tworzenie i używanie serwisów sieciowych



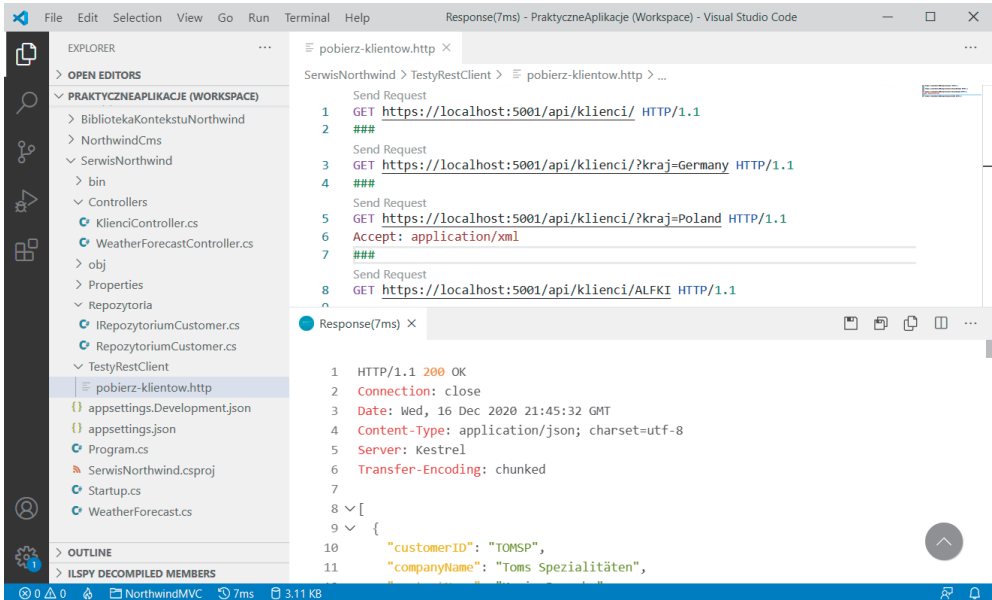
Rysunek 14.1. Żądanie i odpowiedź z sieciowego serwisu pogodowego



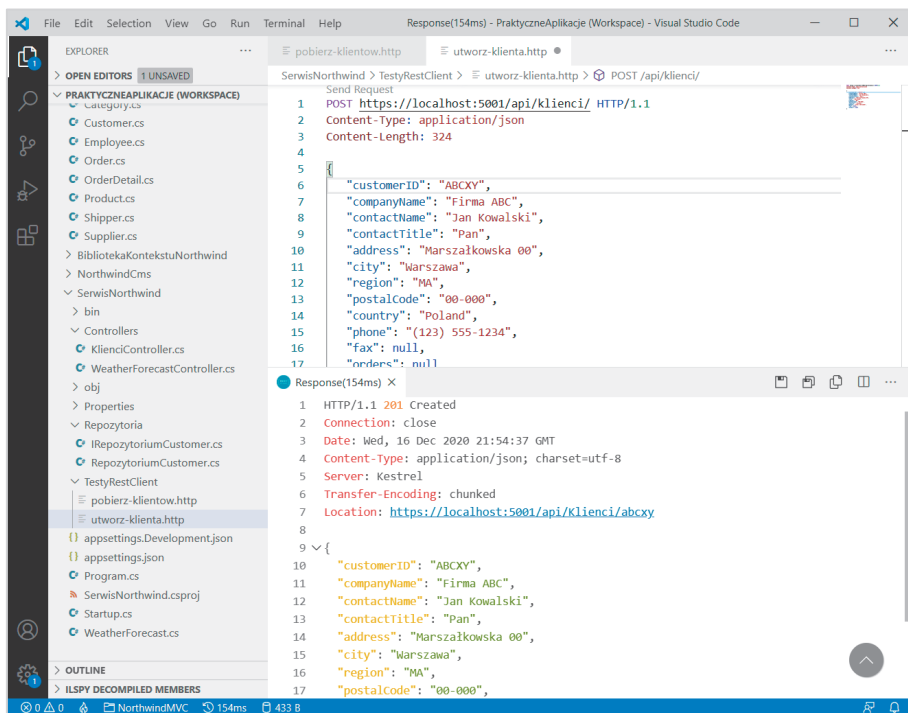
Rysunek 14.2. Klienci z bazy danych Northwind zapisani w dokumencie JSON



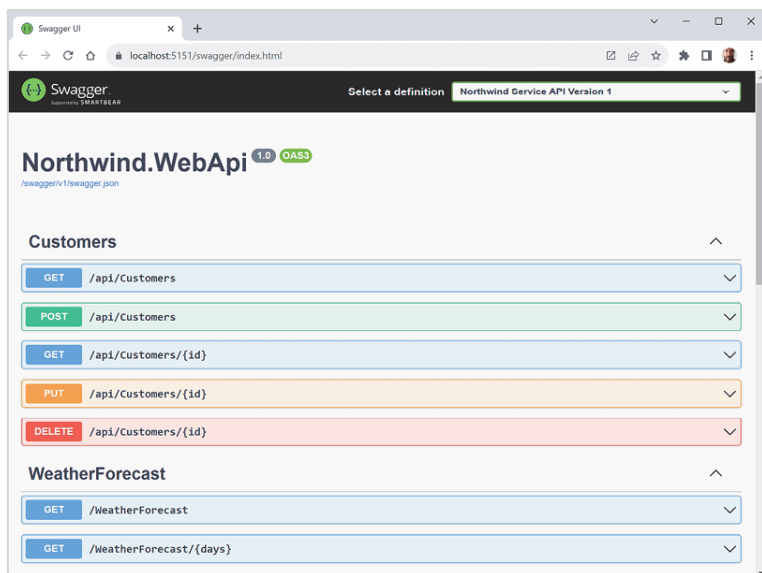
Rysunek 14.3. Wysyłanie żądania HTTP GET za pomocą narzędzia REST Client



Rysunek 14.4. Wysyłanie żądania dokumentu XML i pobieranie odpowiedzi w narzędziu REST Client



Rysunek 14.5. Dodawanie nowego klienta



Rysunek 14.6. Dokumentacja serwisu Northwind.Api przygotowana przez narzędzie Swagger



GET

/api/Klienci/{id}

Parameters

Cancel

Name	Description
id * required string (path)	<div>ALFKI</div>

Execute

Rysunek 14.7. Wprowadzanie identyfikatora klienta przed naciśnięciem przycisku Execute

Request URL

https://localhost:5001/api/Klienci/ALFKI

Server response

CodeDetails

200

Response body

```
{
  "customerID": "ALFKI",
  "companyName": "Alfreds Futterkiste",
  "contactName": "Maria Anders",
  "contactTitle": "Sales Representative",
  "address": "Obere Str. 57",
  "city": "Berlin",
  "region": null,
  "postalCode": "12209",
  "country": "Germany",
  "phone": "030-0074321",
  "fax": "030-0076545",
  "orders": null
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 17 Dec 2020 20:10:20 GMT
server: Kestrel
```

Rysunek 14.8. Informacje o kliencie ALFKI otrzymane w odpowiedzi na ządanie

Request URL

https://localhost:5001/api/Klienci/super

Server response

CodeDetails

404

Error:

Response body

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.4",
  "title": "Not Found",
  "status": 404,
  "traceId": "|7d1c2c8f-4de2623fe7d04687."
}
```

Download

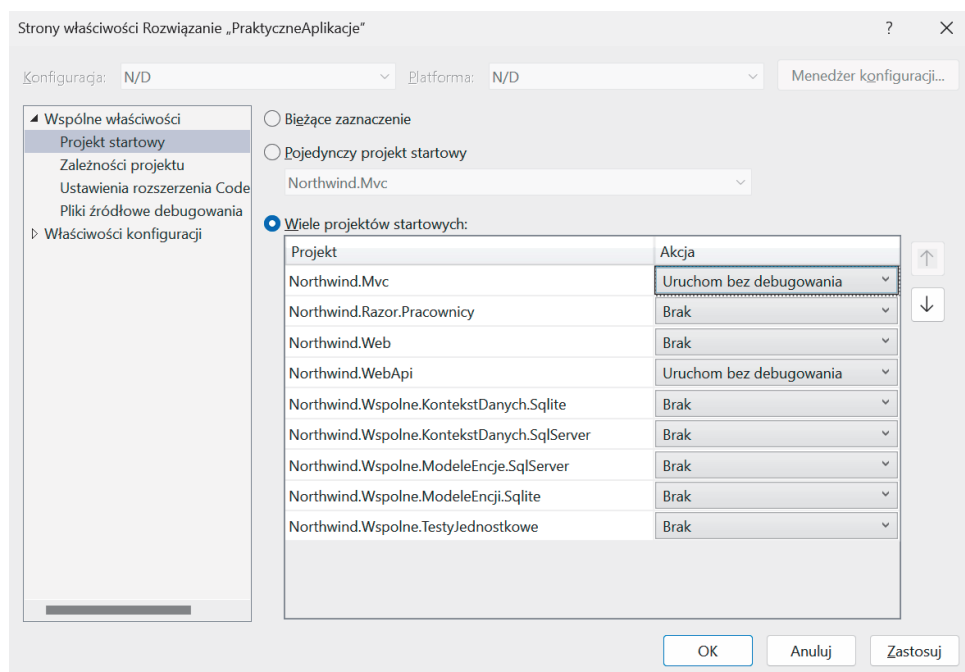
Response headers

```
content-type: application/problem+json; charset=utf-8
date: Thu, 17 Dec 2020 21:27:12 GMT
server: Kestrel
```

Rysunek 14.9. W bazie danych nie ma już usuniętego wcześniej klienta

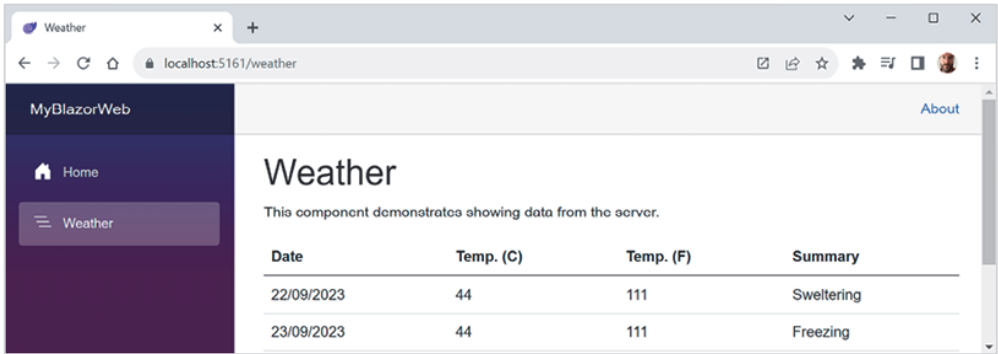


Rysunek 14.10. Klient istnieje w bazie, ale nie udało się go usunąć

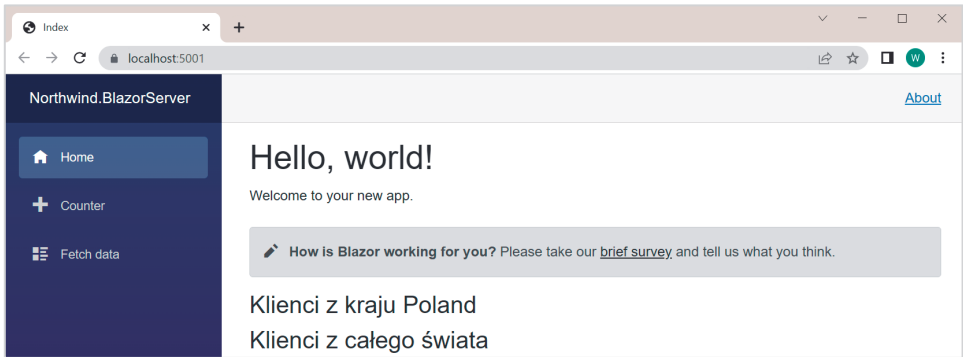


Rysunek 14.11. Zaznaczanie wielu projektów do uruchomienia w Visual Studio 2022

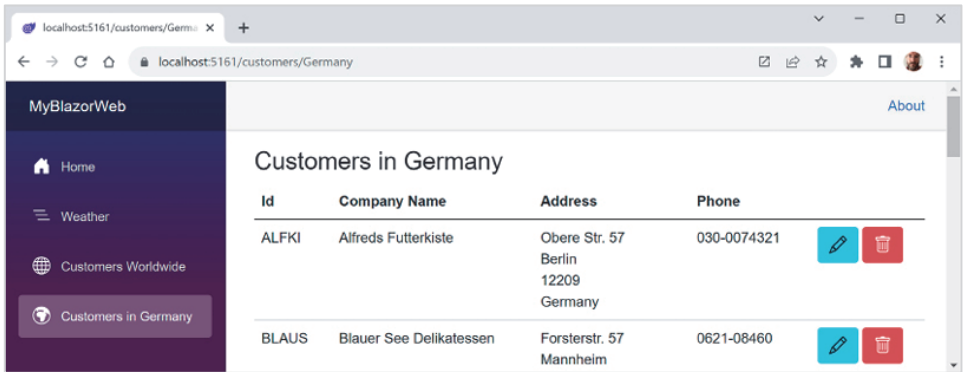
# Rozdział 15. Tworzenie interfejsów użytkownika w technologii Blazor



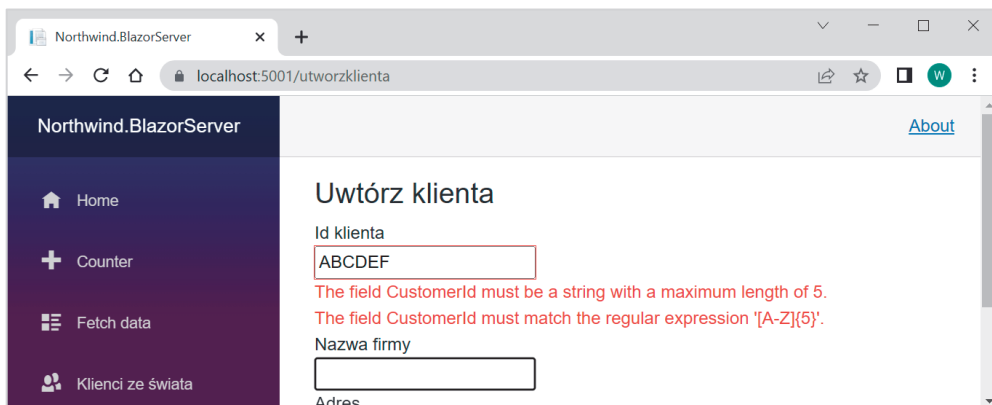
Rysunek 15.1. Pobieranie danych o pogodzie w aplikacji Blazor Server



Rysunek 15.2. Komponent Klienci z parametrem Kraj o wartości Poland oraz bez tego parametru



Rysunek 15.3. Lista klientów z Polski



**Rysunek 15.4. Wprowadzenie niepoprawnego identyfikatora przy tworzeniu nowego klienta**

# Rozdział 16. Epilog

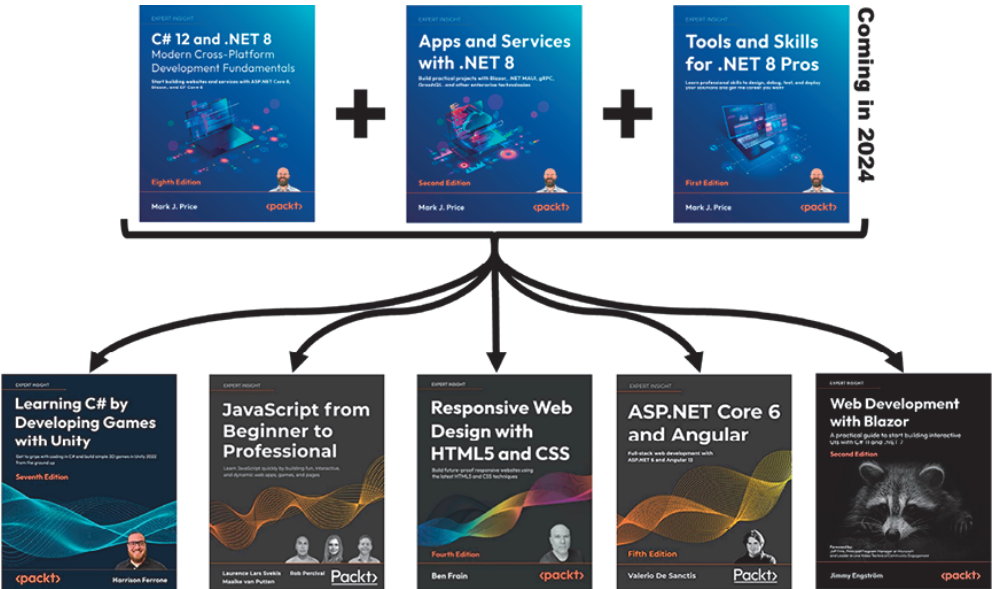
- 1. Język C#, w tym nowe funkcje języka C# 11, programowanie obiektowe oraz debugowanie i testy jednostkowe.
- 2. Biblioteki .NET, w tym obsługa liczb, tekstu i kolekcji, plików oraz danych za pomocą EF Core 7
- 3. Witryny i serwisy sieciowe tworzone za pomocą ASP.NET Core 7 i Blazor.



- 1. Więcej bibliotek .NET do obsługi internacjonalizacji, wielozadaniowości i zapewnienia bezpieczeństwa.
- 2. Więcej na temat obsługi danych za pomocą SQL Server i Azure Cosmos DB.
- 3. Więcej serwisów używających Minimal Web API, Odata, GraphQL, gRPC, SignalR i Azure Functions.
- 4. Więcej graficznych interfejsów użytkownika w ASP.NET Core MVC, Razor, Blazor i .NET MAUI.



Rysunek 16.1. Uzupełniająca książka do nauki C# i .NET



Rysunek 16.2. Książki, z którymi można kontynuować naukę C# i .NET