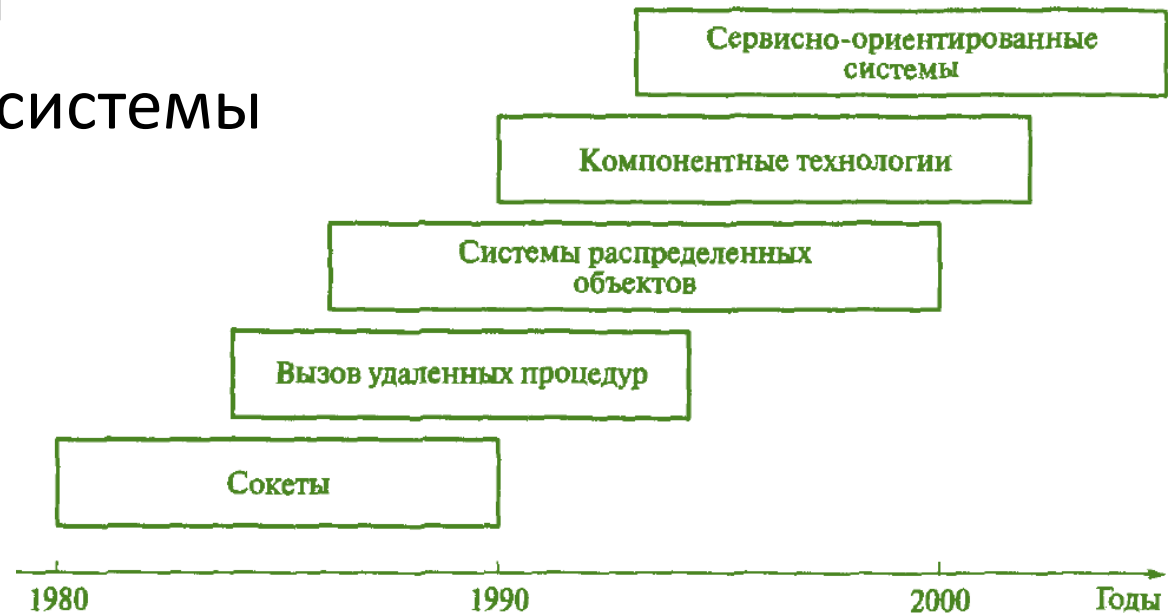


Веб-сервисы

SOAP (asmx) сервисы в .NET

Фазы развития технологий разработки распределенных систем

- Сокеты
- Вызов удаленных процедур
- Системы распределенных объектов
- Компонентные технологии
- Сервисно-ориентированные системы



Web-сервисы

Web-сервисы – отдельные независимые приложения многократного использования, представляют свои функции через Web-интерфейс. Для связи с внешним миром, вместо протокола удаленного вызова процедур (RPC), используют протокол HTTP.

Потребители могут использовать службы независимо от языка их реализации и устройства на котором они установлены.

Web-сервисы позволяют приложениям совместно использовать данные и функции таким способом, при котором не имеет значения, как именно эти приложения выполняются, какую платформу, операционную систему или устройство они используют.

В качестве «официального» определения можно рассматривать определение, которое дается консорциумом W3C:

«Web-сервис представляет собой приложение, которое идентифицируется строкой URI. Интерфейсы и привязки данного приложения описываются и обнаруживаются с использованием XML-средств. Приложения взаимодействуют посредством обмена сообщениями, которые пересылаются с использованием интернет-протоколов».

Историческая справка

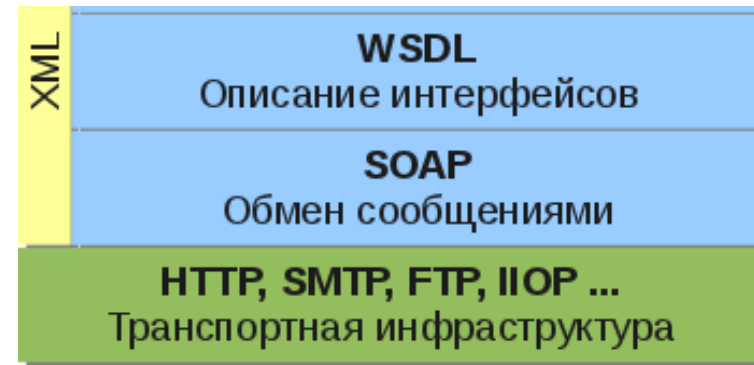
- С самого начала корпорация Microsoft была одним из основных разработчиков стандарта SOAP.
- В 2002 году в составе самой первой версии ASP.NET 1.0 она представила технологию ASMX (Active Server Method Extended), которая позволила разработчикам в новейшей Visual Studio 2002 легко создавать и потреблять SOAP веб-сервисы.
- Эта технология на MSDN имеет название «XML Web Services».
- Консорциум W3C одобрил SOAP 1.1 в 2000 году, SOAP 1.2 в 2003 году (дополнен в 2007 году).
- В 2007 году в составе .NET 3.0 была представлена технология Windows Communication Foundation (WCF), которая стала официальной заменой ASMX.
- Несмотря на то, что технология ASMX уже давно не развивается, она продолжает широко использоваться и поддерживается новейшими версиями .NET Framework.

Web-сервисы

- Web-сервисы исполняются на сервере (исполняющая среда - ASP.NET)
- Web-сервисы публикуют методы, которые могут быть вызваны внешними клиентами
- Web-сервисы ожидают поступления http-запросов для вызова методов
- Web-сервисы исполняют запросы и возвращают http ответы

Протоколы Web-сервисов:

- **WSDL** – язык описания Web-служб.
- **SOAP** – простой протокол доступа к объектам.

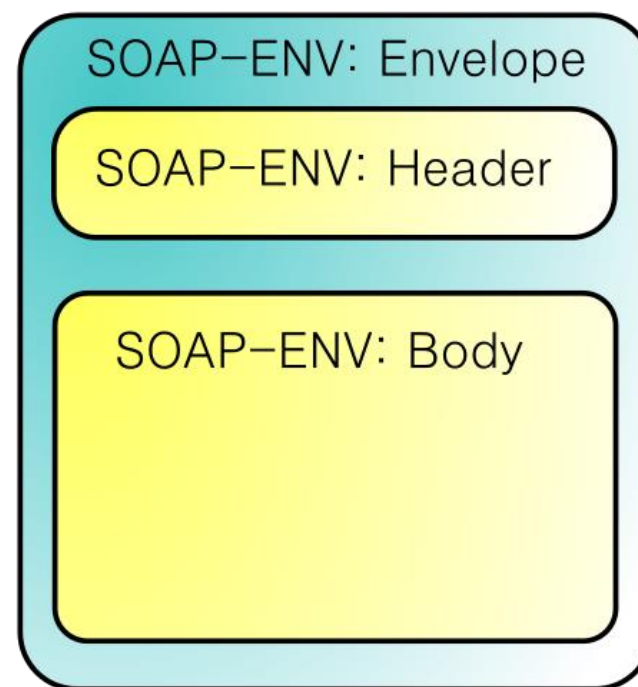


SOAP

- SOAP (первоначально *Simple Object Access Protocol*) - это спецификация протокола для обмена структурированной информацией при реализации веб-сервисов в компьютерных сетях .
 - Он использует XML как формат сообщения и опирается на протоколы уровня приложения (HTTP), для согласования и передачи сообщений.

Структура SOAP

- **Сообщение SOAP** представляет информацию, которой обмениваются два узла.
- Сообщение содержит две части данных:
 - **SOAP header**
 - Заголовочный блок SOAP содержит информацию, которая может быть полезной для посредников SOAP и для конечного адресата (например, приоритет).
 - **SOAP body**
 - Тело сообщения содержит фактическую полезную нагрузку. Интерпретация и обработка тела SOAP определяются блоками *header*.



WSDL (*Web Services Description Language*)

язык описания веб-сервисов и доступа к ним, основанный на языке XML.

Структура WSDL

- **определение типов данных (types)** — определение вида отправляемых и получаемых сервисом XML-сообщений
- **элементы данных (message)** — сообщения, используемые web-сервисом
- **абстрактные операции (portType)** — список операций, которые могут быть выполнены с сообщениями
- **связывание сервисов (binding)** — способ, которым сообщение будет доставлено

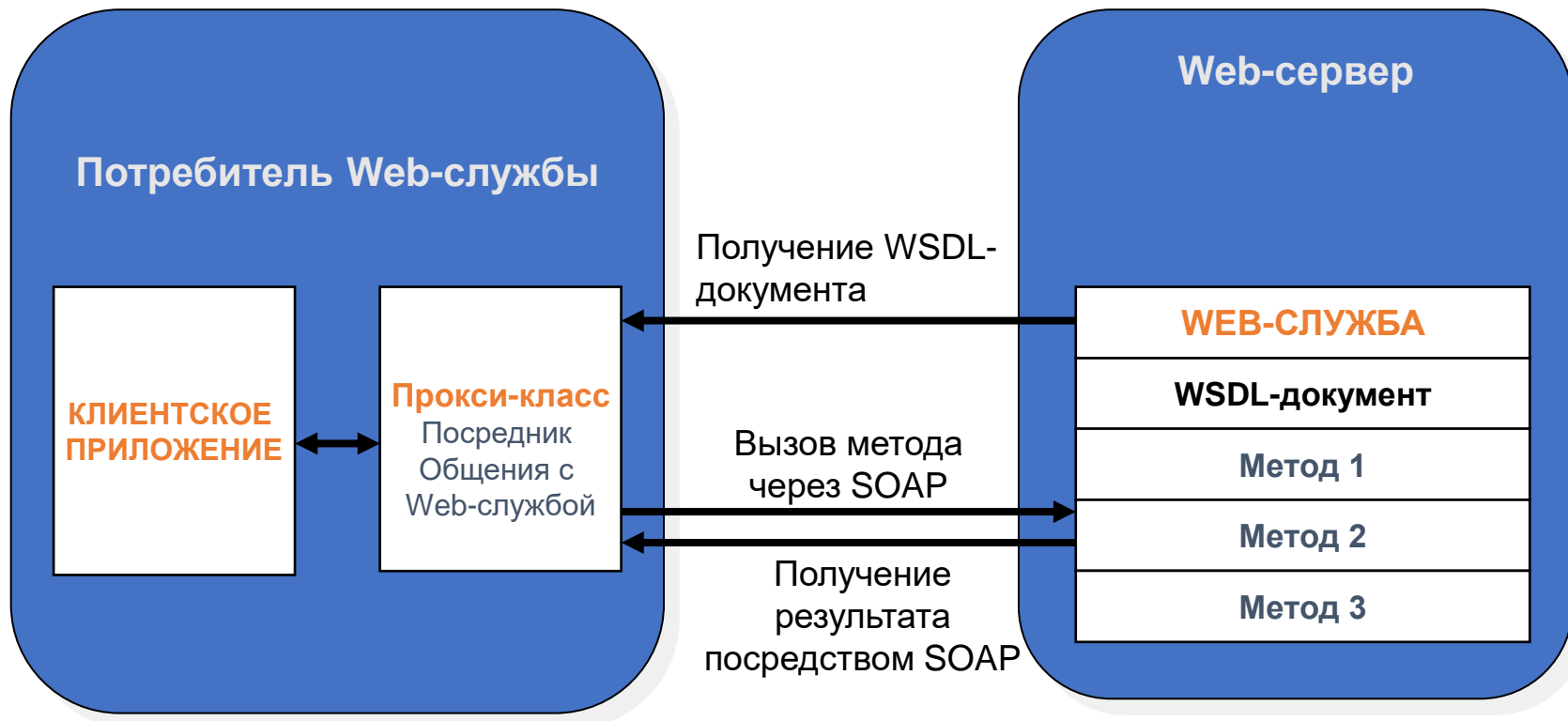
Пример WSDL

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```


Архитектура приложений



Реализация Web-служб для .NET приложений

1. Вы разрабатываете web-службу как **.NET**-класс с атрибутами, которые идентифицируют его как web-службу с некоторыми функциями.
2. В среде **.NET** автоматически создается документ WSDL, где описывается, как клиент должен взаимодействовать с web-службой.
3. Потребитель находит вашу web-службу и, решив воспользоваться ею, добавляет соответствующую web-ссылку в проект **Visual Studio** (или запускает утилиту `wsdl.exe`).
4. В среде **.NET** осуществляется автоматическая проверка документа WSDL и генерируется прокси-класс, который позволяет потребителю взаимодействовать с web-службой.
5. Потребитель вызывает один из методов вашего класса web-службы. С его точки зрения этот вызов не отличается от вызова метода любого другого класса, но в действительности потребитель взаимодействует с прокси-классом, а не с web-службой.
6. Прокси-класс преобразует переданные параметры в сообщение SOAP и отправляет его web-службе.
7. Вскоре прокси-класс получает SOAP-ответ, преобразует таковой в соответствующий тип данных и возвращает его как обычный тип данных **.NET**.
8. Потребитель использует возвращенную ему информацию.

Общие характеристики Web-сервисов

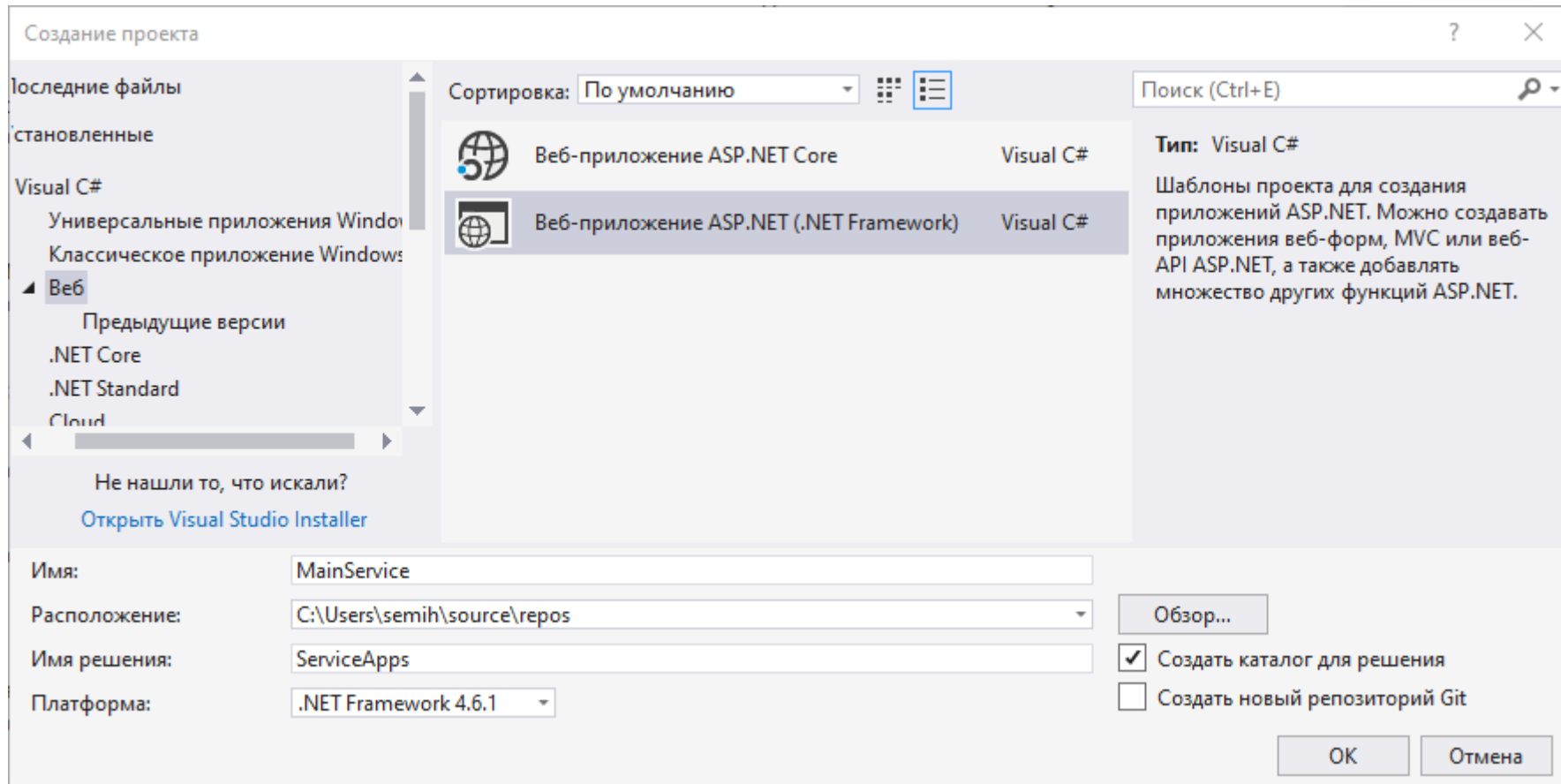
Web-сервисы реализуются как asmx-файлы. Исполняемый код Web-сервиса может находиться как в самом файле asmx, так и в специальном файле с кодом.

Asmx-файлы начинаются директивой `@ WebService`, эта директива содержит, по крайней мере, атрибут `class`, задающий класс, из которого состоит Web-сервис.

Классы Web-сервисов могут иметь необязательные атрибуты `WebService`.

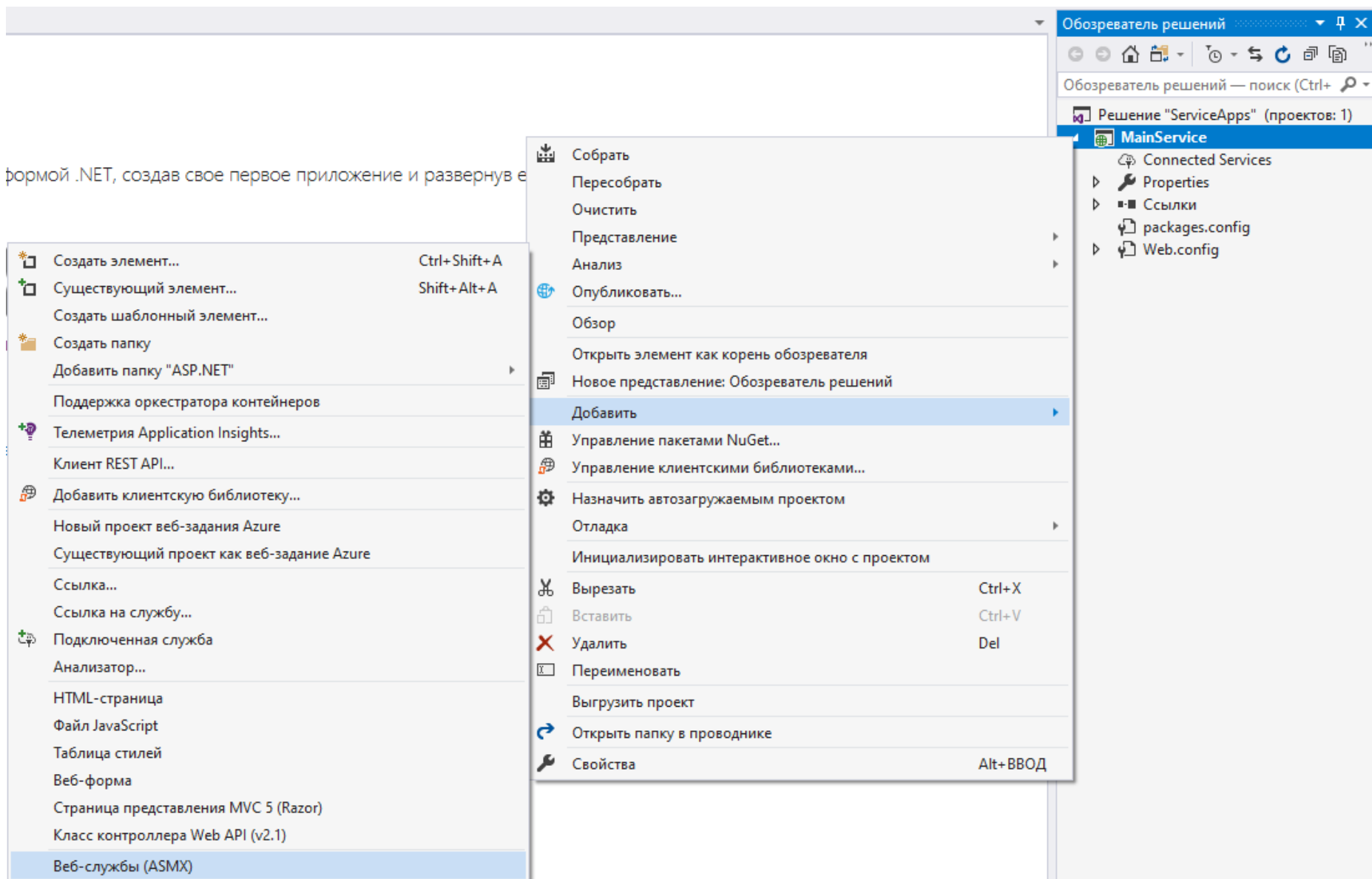
Web-методы объявляются путем назначения открытым методам класса сервиса атрибута `WebMethod`.

Создание проекта для Web-службы

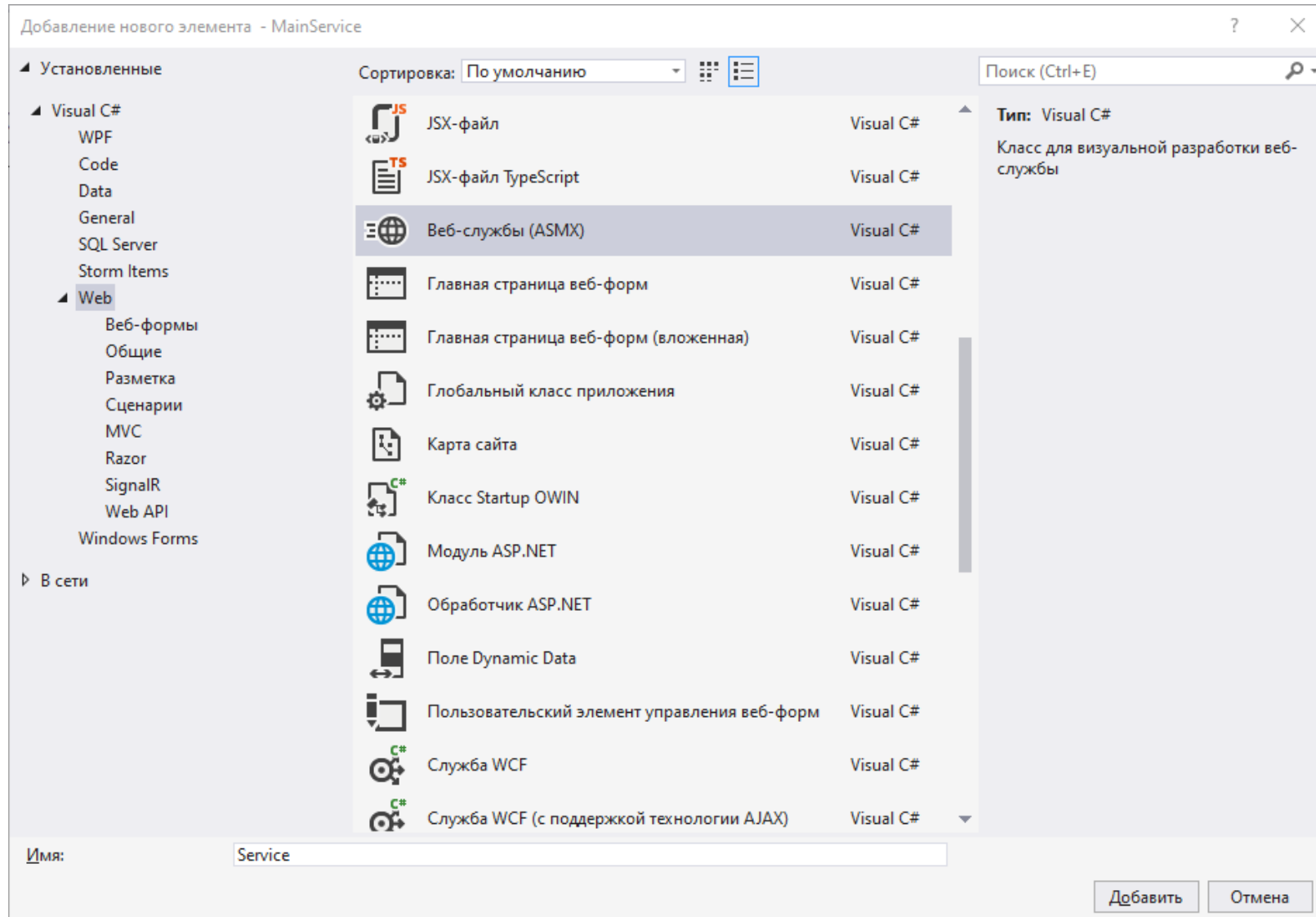


Создание службы

формой .NET, создав свое первое приложение и развернув е

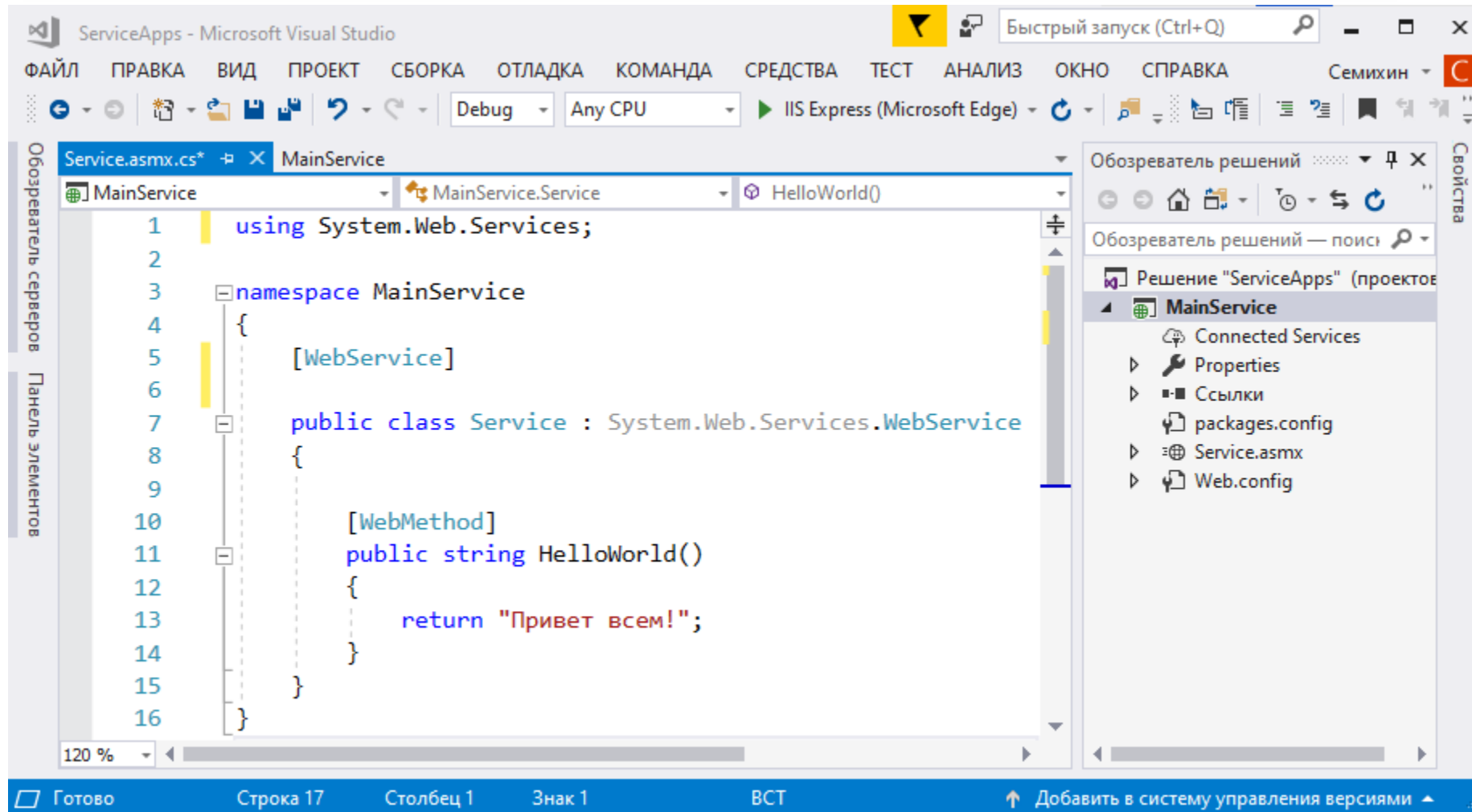


Создание службы

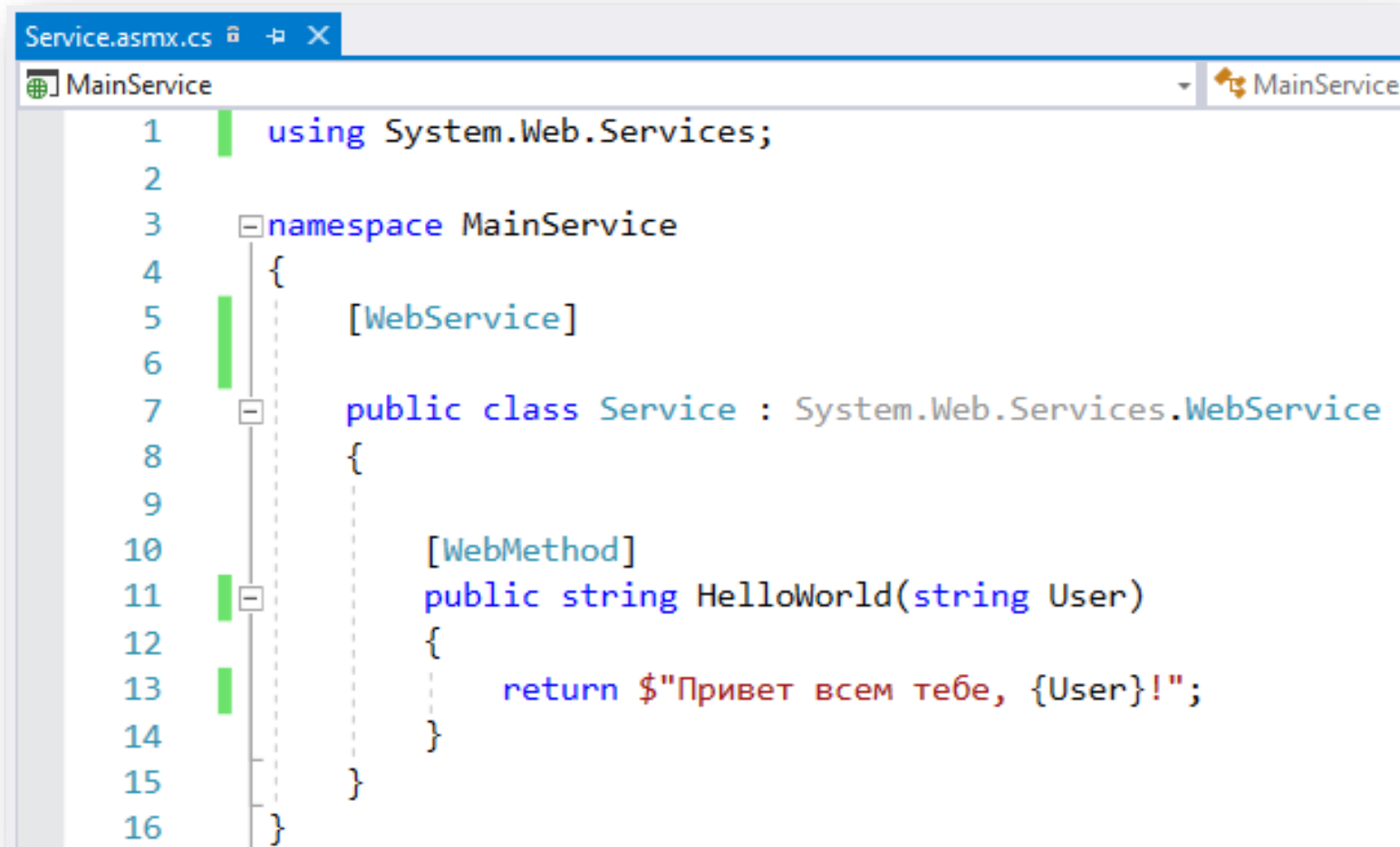


Hello, world...

```
Service.asmx  X Service.asmx.cs
1  <%@ WebService Language="C#" CodeBehind="Service.asmx.cs" Class="MainService.Service" %>
```



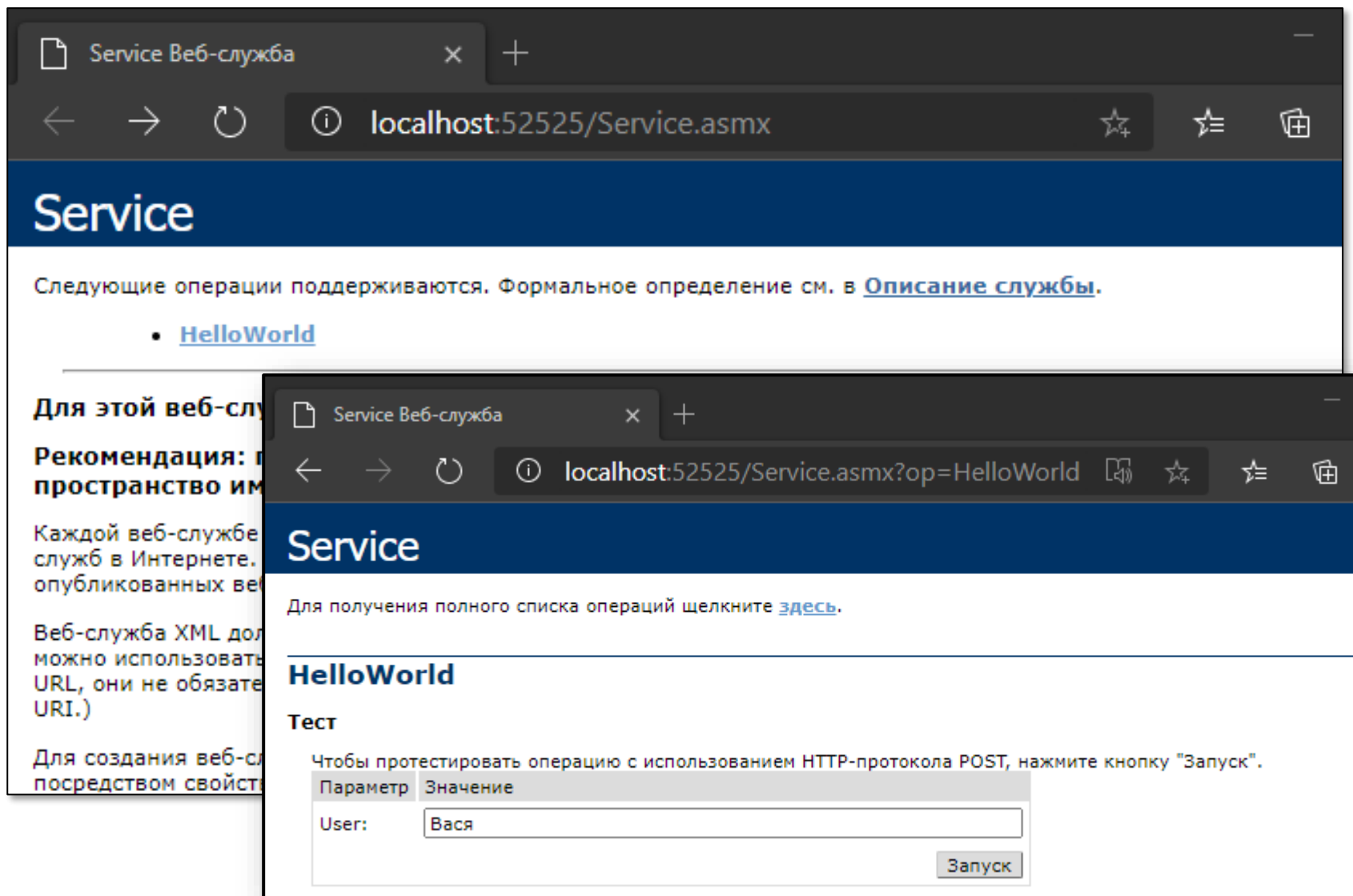
Изменение методов



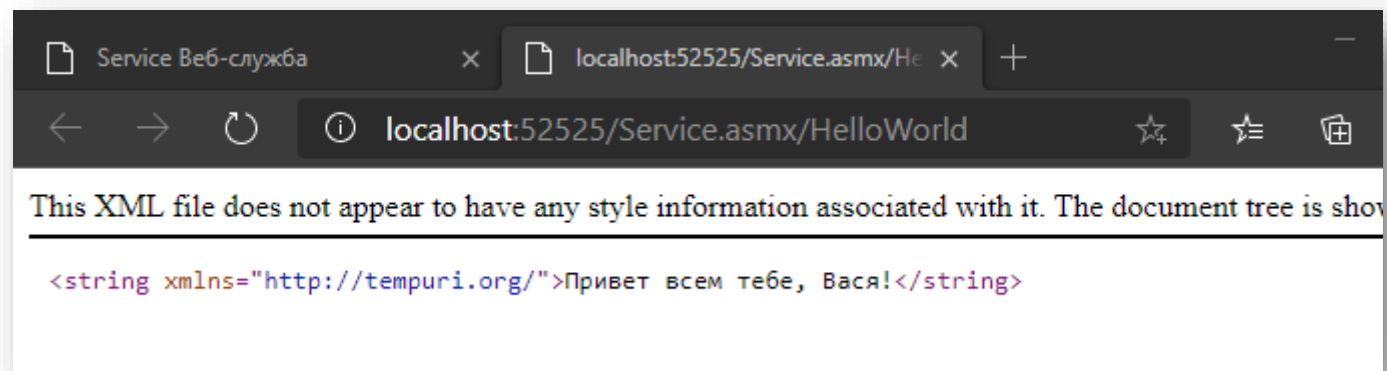
```
Service.asmx.cs
MainService

1  using System.Web.Services;
2
3  namespace MainService
4  {
5      [WebService]
6
7      public class Service : System.Web.Services.WebService
8      {
9
10         [WebMethod]
11         public string HelloWorld(string User)
12         {
13             return $"Привет всем тебе, {User}!";
14         }
15     }
16 }
```

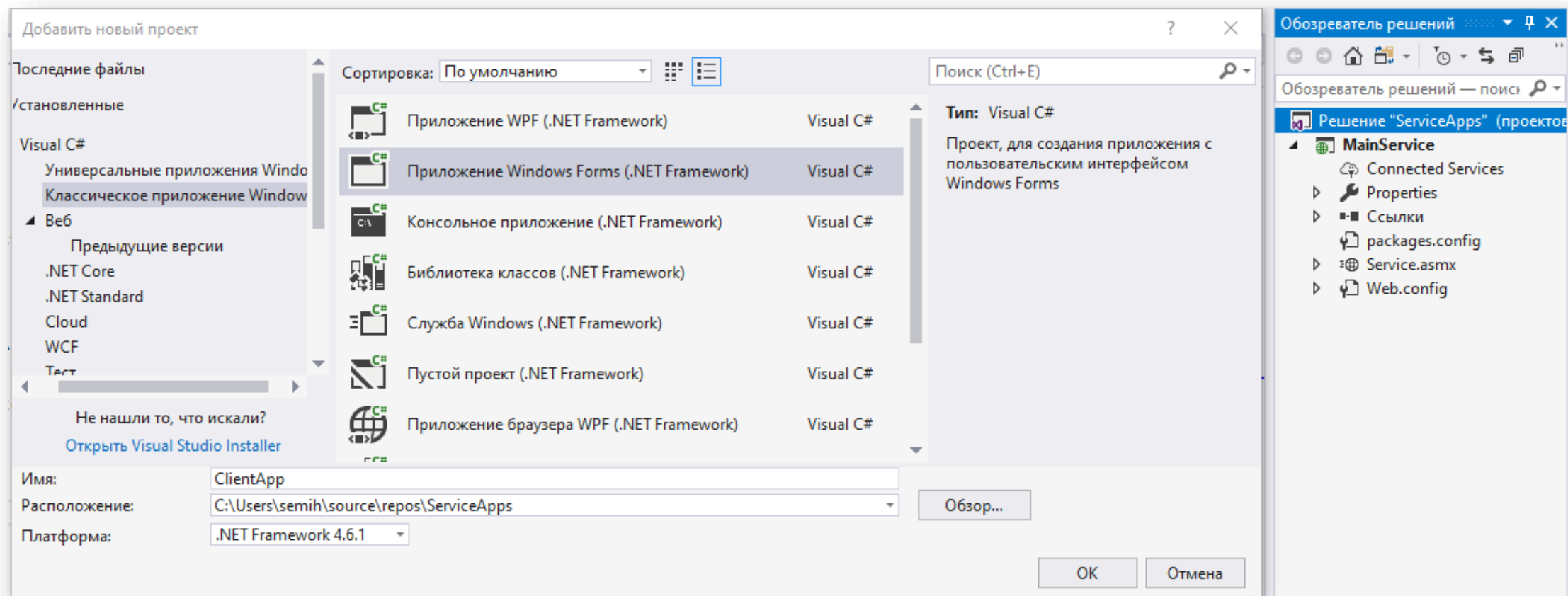

Запуск службы



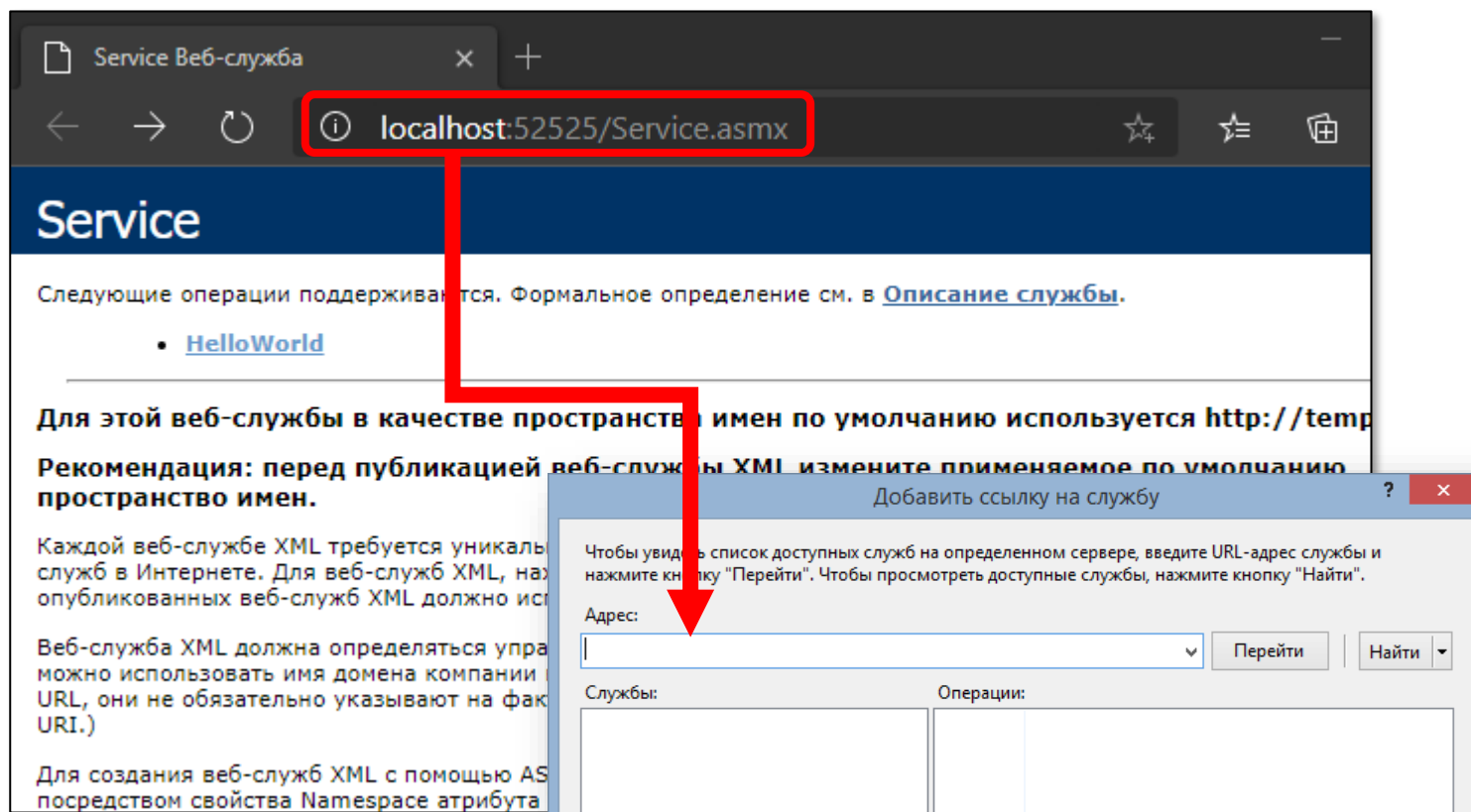
Возвращаемый результат



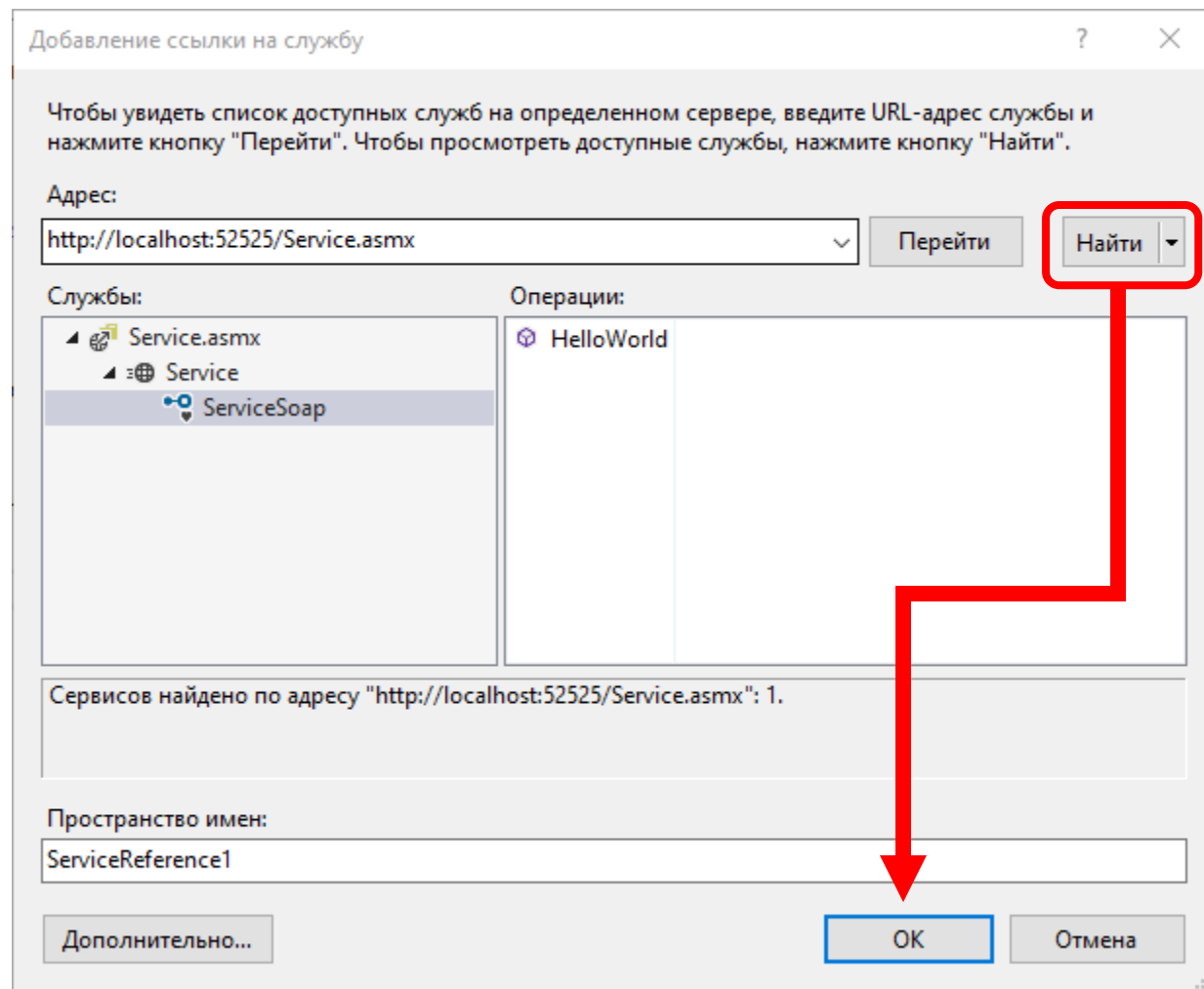
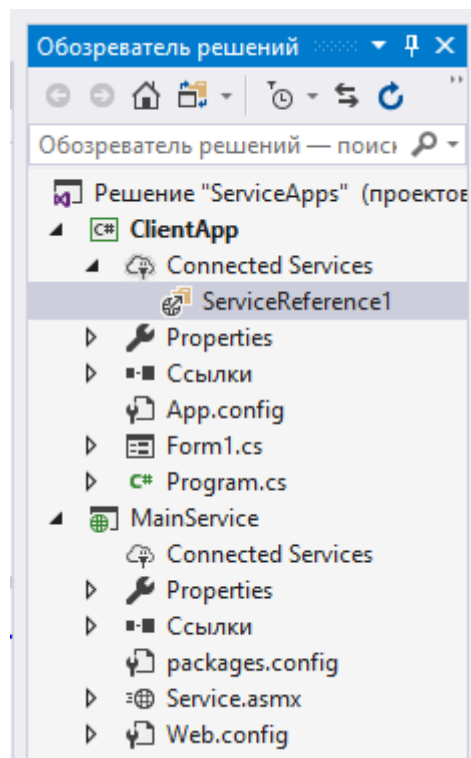
Приложение – клиент веб-службы



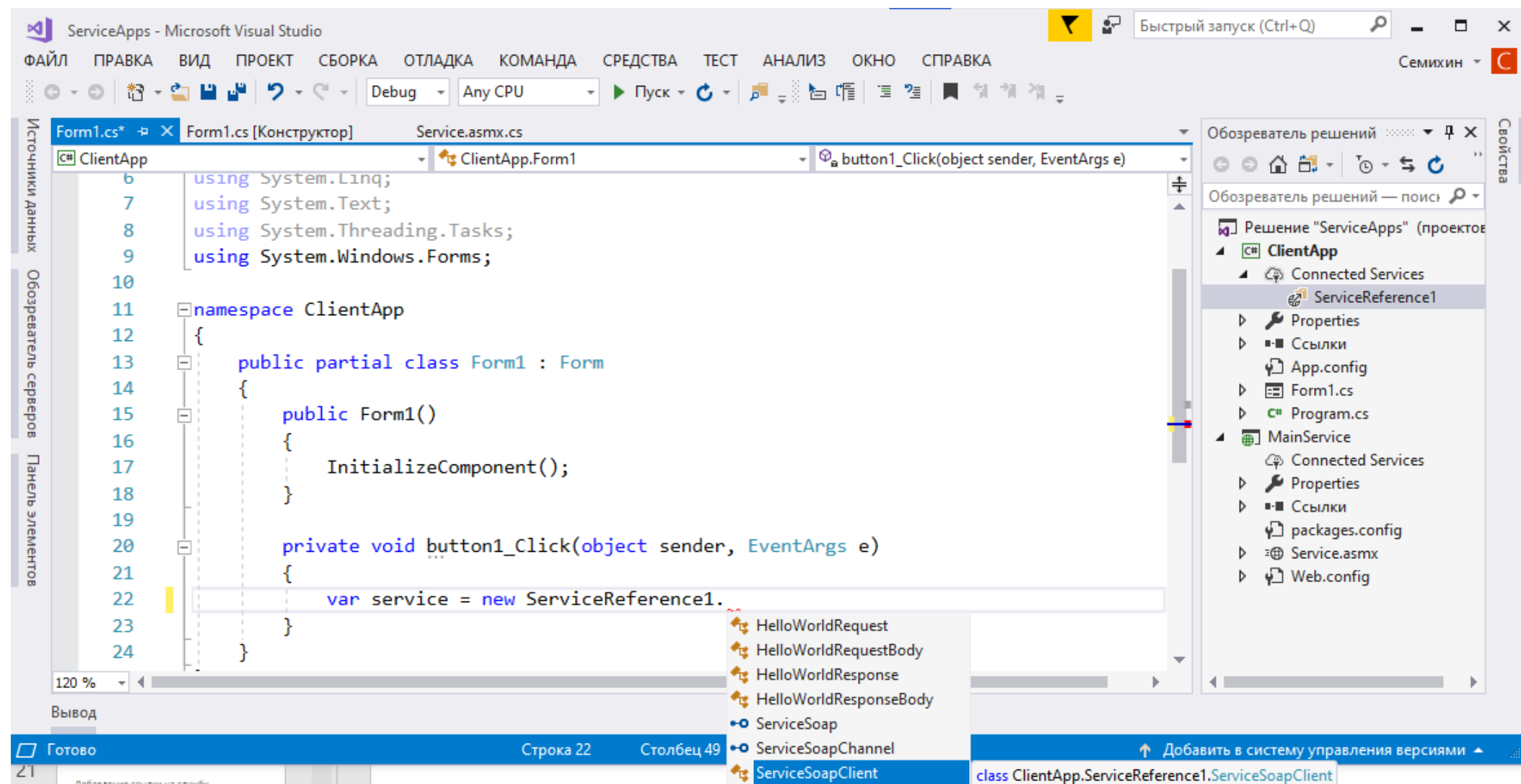
Добавление ссылки на службу...



Добавление ссылки на службу...



Код клиента



Код клиента

