

RabbitMQSample

RabbitMQ teknolojisinin nasıl kullanılacağına dair örnek bilgi amaçlı hazırlanmış bir konsol uygulaması

Table of Contents


Symbol Reference	1
RabbitMQSample Namespace	1
Classes	1
Kisi Class	1
Kisi Properties	2
Classes	2
Program Class	3
Program Methods	3
Program.GirilenDegerUzerindenHataMesajiGoster Method	4
Program.IlkVeriyiAl Method	4
Program.KisiBilgisiOlustur Method	4
Program.KisiyiKuyrugaGonder Method	5
Program.KisiyiKuyruktanAl Method	5
Program.KuyrugaGonderilenKisiBilgisiniMesajOlarakGoster Method	6
Program.KuyrugaGondermeSonrasiEkrandallgiliMesajlariGoster Method	6
Program.Main Method	7
Program.RabbitMQKuyrugaGonder Method	7
Program.RabbitMQKuyruktanAl Method	8
Files	8
Kisi.cs	9
Program.cs	9
RabbitMQ.sln	13
RabbitMQSample.csproj	13
Index	a

1 Symbol Reference

1.1 RabbitMQSample Namespace

This is namespace RabbitMQSample.

Classes


	Name	Description
	Kisi (see page 1)	RabbitMQ teknolojisi ile gönderilecek ve alınacak verilerin içerisinde bulunduğu sınıftır

1.1.1 Classes

Summary

The following table lists classes in this documentation.

Classes

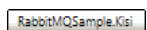
	Name	Description
	Kisi (see page 1)	RabbitMQ teknolojisi ile gönderilecek ve alınacak verilerin içerisinde bulunduğu sınıftır

1.1.1.1 Kisi Class

Summary

RabbitMQ teknolojisi ile gönderilecek ve alınacak verilerin içerisinde bulunduğu sınıftır

Class Hierarchy



C#

```
public class Kisi;
```

File

Kisi.cs ([see page 9](#))

Kisi Properties

	Name	Description
	Adi (see page 2)	Kisinin Adi Bilgisi
	DogumTarihi (see page 2)	Kisinin Dogum Tarihi Bilgisi
	DogumYeri (see page 2)	Kisinin Dogum Yeri Bilgisi
	ID (see page 2)	Kisinin ID Bilgisi (Veri Tabanından geliyor gibi düşünebiliriz)
	Soyadi (see page 2)	Kisinin Soyadi Bilgisi

1.1.1.1.1 Kisi Properties

1.1.1.1.1.1 Kisi.Adi Property

Summary

Kisinin Adi Bilgisi

C#

```
public string Adi;
```

1.1.1.1.1.2 Kisi.DogumTarihi Property

Summary

Kisinin Dogum Tarihi Bilgisi

C#

```
public DateTime DogumTarihi;
```

1.1.1.1.1.3 Kisi.DogumYeri Property

Summary

Kisinin Dogum Yeri Bilgisi

C#

```
public string DogumYeri;
```

1.1.1.1.1.4 Kisi.ID Property

Summary

Kisinin ID Bilgisi (Veri Tabanından geliyor gibi düşünebiliriz)

C#

```
public int ID;
```

1.1.1.1.1.5 Kisi.Soyadi Property

Summary

Kisinin Soyadi Bilgisi

C#


```
public string Soyadi;
```

1.2 Classes

Summary

The following table lists classes in this documentation.

Classes

	Name	Description
	Program (see page 3)	Rabbit MQ uygulamasinin konsolda çalishmasini saglayan metotlari içeren temel siniftir.

1.2.1 Program Class

Summary

Rabbit MQ uygulamasinin konsolda çalismasini saglayan metotlari içeren temel siniftir.

Class Hierarchy

Program

C#

```
public class Program;
```

File

Program.cs (see page 9)

Program Methods

	Name	Description
	GirilenDegerUzerindenHataMesajiGoster (see page 4)	Ekrandan girilen deger bir sayi veya e harfi degilse Geçerli bir deger olmadigi için Hata mesajı gösterilmesi ile ilgili kod blogudur
	IlkVeriyiAl (see page 4)	Ekrandan bir sayi veya çıkis istegi olan e harfini alma ile ilgili metottur
	KisiBilgisiOlustur (see page 4)	Bogus kütüphanesi kullanılarak Rastgele Kisi sinifina ait Kisi bilgisi olusturulmasi için Gerekli islemlerin gerçekleştirildiği metottur.
	KisiyiKuyrugayaGonder (see page 5)	RabbitMQ tarafinda kuyruguyu olusturup ilgili kuyrugaya kisi bilgisinin gönderilmesi için Gerekli islemlerin gerçekleştirildiği metottur.
	KisiyiKuyruktanAl (see page 5)	RabbitMQ kuyrugundaki Kisi bilgilerini alip Bu bilgilerin gösterilmesi için gerekli islemlerin gerçekleştirildiği metottur.
	KuyrugayaGonderilenKisiBilgisiniMesajOlarakGoster (see page 6)	RabbitMQ kuyruguna gönderilen kisi bilgisinin Konsol tarafinda gösterilmesi için Gerekli islemlerin gerçekleştirildiği metottur.
	KuyrugayaGondermeSonrasiEkrandalligiliMesajlariGoster (see page 6)	RabbitMQ kuyruguna veriler gönderildikten sonra Sistem tarafından ekranda gösterilen mesajların vb. Bulundugu kod blogudur
	Main (see page 7)	Konsol uygulamasinin çalıştigi ana metottur
	RabbitMQKuyrugayaGonder (see page 7)	Ekrandan alınan sayi kadar Kisi olusturup Rabbit MQ kuyruguna göndermeyi ve gönderilen kisiler ile ilgili bilgileri Konsol ekranında göstermeye yarayan metottur
	RabbitMQKuyruktanAl (see page 8)	Rabbit MQ kuyrugunda biriken bilgilerin FIFO mantigina göre Kuyruktan alinarak Kisi bilgilerinin konsol ekranında görüntülenmesini saglayan metottur

1.2.1.1 Program Methods

1.2.1.1.1 Program.GirilenDegerUzerindenHataMesajiGoster Method

Summary

Ekrandan girilen deger bir sayi veya e harfi degilse Geçerli bir deger olmadigi için Hata mesaji gösterilmesi ile ilgili kod blogudur

C#

```
private static void GirilenDegerUzerindenHataMesajiGoster(string? metin);
```

Parameters

Parameters	Description
string? metin	Ekrandan alınan deger

Body Source

```
1: private static void GirilenDegerUzerindenHataMesajiGoster(string? metin)
2: {
3:     Console.WriteLine($"{metin} geçerli bir sayi veya uygulama kapatma istegi degildir");
4:     Console.WriteLine("-----");
5: }
```

1.2.1.1.2 Program.IlkJeriYiAl Method

Summary

Ekrandan bir sayi veya çıkis istegi olan e harfini alma ile ilgili metottur

C#

```
private static string? IlkJeriYiAl();
```

Returns

Sayi veya Çıkis Istegi

Body Source

```
1: private static string? IlkJeriYiAl()
2: {
3:     Console.WriteLine("RabbitMQ kuyruguna kaç tane kayıt göndermek istiyorsunuz?
(Programdan çıkmak için E veya e yazınız)");
4:     var sayi = Console.ReadLine();
5:     return sayi;
6: }
```

1.2.1.1.3 Program.KisiBilgisiOlustur Method

Summary

Bogus kütüphanesi kullanılarak Rastgele Kisi sinifina ait Kisi bilgisi olusturulmasi için Gerekli islemlerin gerçekleştirildiği metottur.

C#

```
private static Kisi KisiBilgisiOlustur();
```

Returns

Kisi Bilgisi

Body Source

```
1: private static Kisi KisiBilgisiOlustur()
2: {
3:     // Bogus kütüphanesi kullanılarak sahte veri içeren kisi kayıtlarının
4:     // otomatik olarak olusturulmasını sağlayan yapidir
```

```

5:     var kisiOlusturucu = new Faker<Kisi>()
6:     .CustomInstantiator(f => new Kisi())
7:     .RuleFor(u => u.Adı, f => f.Name.FirstName())
8:     .RuleFor(u => u.Soyadı, f => f.Name.LastName())
9:     .RuleFor(u => u.DogumYeri, (f, u) => f.Address.City())
10:    .RuleFor(u => u.DogumTarihi, (f, u) => f.Person.DateOfBirth)
11:    .RuleFor(u => u.ID, f => new Random().Next());
12:
13:    var kisi = kisiOlusturucu.Generate();
14:    return kisi;
15: }

```

1.2.1.1.4 Program.KisiyiKuyruğaGonder Method

Summary

RabbitMQ tarafında kuyruğu oluşturup ilgili kuyruğa kisi bilgisinin gönderilmesi için Gerekli işlemlerin gerçekleştirildiği metottur.

C#

```
private static void KisiyiKuyruğaGonder(Kisi kisi, IModel channel);
```

Parameters

Parameters	Description
Kisi kisi	Kisi Bilgisi
IModel channel	RabbitMQ Kanal Bilgisi

Body Source

```

1: private static void KisiyiKuyruğaGonder(Kisi kisi, IModel channel)
2: {
3:     // burada coderseardar adında bir kanal açarak, onun içerisine ilgili kisi
    kayıtlarının gönderilmesi için hazırlamaktadır
4:     // diğer parametreler de kullanılarak kanalın kullanımı özelleştirilebilir
5:     channel.QueueDeclare(queue: "coderseardar", durable: false, exclusive: false,
    autoDelete: false, arguments: null);
6:
7:     string message = JsonConvert.SerializeObject(kisi);
8:     // sınıf serialize edilerek byte dizisi haline dönüştürülür
9:     var body = Encoding.UTF8.GetBytes(message);
10:
11:    // burada dönüştürülen byte dizisinin kanala gönderilmesi sağlanır
12:    channel.BasicPublish(exchange: "", routingKey: "coderseardar", basicProperties:
    null, body: body);
13: }

```

1.2.1.1.5 Program.KisiyiKuyruktanAl Method

Summary

RabbitMQ kuyruğundaki Kisi bilgilerini alıp Bu bilgilerin gösterilmesi için gerekli işlemlerin gerçekleştirildiği metottur.

C#

```
private static void KisiyiKuyruktanAl(IModel channel);
```

Parameters

Parameters	Description
IModel channel	RabbitMQ Kanal Bilgisi

Body Source

```

1: private static void KisiyiKuyruktanAl(IModel channel)
2: {
3:     // burada coderseardar adında bir kanal açarak, onun içerisine ilgili kisi

```

```

kayitlarinin gönderilmesi için hazırlamaktadır
4:    // diger parametreler de kullanilarak kanalın kullanımı özelleştirilebilir
5:    channel.QueueDeclare(queue: "coderserdar", durable: false, exclusive: false,
autoDelete: false, arguments: null);
6:
7:    // burada kuyruğu tüketecek bir consumer nesnesi oluşturulur ve tüketmeye başlar
8:    var consumer = new EventingBasicConsumer(channel);
9:    consumer.Received += (model, ea) =>
10:   {
11:       var body = ea.Body;
12:       var message = Encoding.UTF8.GetString(body.ToArray());
13:       Kisi kisi = JsonConvert.DeserializeObject<Kisi>(message);
14:       Console.WriteLine($"Adi Soyadi: {kisi.Adi} {kisi.Soyadi} [{kisi.DogumYeri}]");
15:       Console.WriteLine("RabbitMQ ile tanistiniz. İyi günler.");
16:   };
17:
18:   // kanalın kuyruğu tüketerek ilgili bilgilerin konsolda gösterilmesi sağlanır
19:   channel.BasicConsume(queue: "coderserdar", autoAck: true, consumer: consumer);
20:
21:   Console.ReadLine();
22: }

```

1.2.1.1.6 Program.KuyruğaGonderilenKisiBilgisiniMesajOlarakGoster Method

Summary

RabbitMQ kuyruğuna gönderilen kisi bilgisinin Konsol tarafında gösterilmesi için Gerekli işlemlerin gerçekleştirildiği metottur.

C#

```
private static void KuyruğaGonderilenKisiBilgisiniMesajOlarakGoster(int sayac, Kisi kisi);
```

Parameters

Parameters	Description
int sayac	Döngüdeki sayaç bilgisi
Kisi kisi	Kisi Bilgisi

Body Source

```

1: private static void KuyruğaGonderilenKisiBilgisiniMesajOlarakGoster(int sayac, Kisi kisi)
2: {
3:     Console.WriteLine($"Gönderilen kisi: Adi Soyadi: {kisi.Adi} {kisi.Soyadi} Dogum
Tarihi: {kisi.DogumTarihi.ToShortDateString()}");
4:     Console.WriteLine((sayac + 1) + ". kisi gönderildi...");
5: }

```

1.2.1.1.7 Program.KuyruğaGondermeSonrasiEkranıİlgiliMesajlarıGoster Method

Summary

RabbitMQ kuyruğuna veriler gönderildikten sonra Sistem tarafından ekranda gösterilen mesajların vb. Bulunduğu kod blogudur

C#

```
private static void KuyruğaGondermeSonrasiEkranıİlgiliMesajlarıGoster();
```

Body Source

```

1: private static void KuyruğaGondermeSonrasiEkranıİlgiliMesajlarıGoster()
2: {
3:     Console.WriteLine("-----");
4:     Console.WriteLine("RabbitMQ kuyruğundan alma işlemleri için ekrana bir değer
giriniz");

```



```
5: Console.ReadLine();
6: }
```

1.2.1.1.8 Program.Main Method

Summary

Konsol uygulamasinin çalıştığı ana metottur

C#

```
public static void Main(string[] args);
```

Parameters

Parameters	Description
string[] args	Gönderilen Argüman Dizisi

Body Source

```
1: public static void Main(string[] args)
2: {
3:     var programKapansinMi = false;
4:     // Kullanici Programdan çıkmak için e veya E yazmadigi sürece çalışmasını sağlayan
döngüdür
5:     while (!programKapansinMi)
6:     {
7:         string? sayi = İlkVeriyiAl();
8:         // eger geçerli bir sayi girerse o sayi kadar Kisi kaydi olusturup
9:         // RabbitMQ kuyruguna yollayacak, alacak
10:        // Ve bu bilgileri konsol ekranında yazdıracak kod blogu burasıdır
11:        if (int.TryParse(sayi, out int kisiSayisi))
12:        {
13:            RabbitMQKuyrugayaGonder(kisiSayisi);
14:            KuyrugayaGondermeSonrasiEkrandaIlgiliMesajlariGoster();
15:            RabbitMQKuyruktanAl();
16:        }
17:        else
18:        {
19:            // Eger e veya E harfi girerse kullanıcı konsoldan
20:            // programdan çıkis yapılmasını sağlayacak kod blogudur
21:            if (sayi == "e" || sayi == "E")
22:                programKapansinMi = true;
23:            // eger geçersiz bir sayi veya metin girerse gösterilecek hata mesajıdır
24:            else
25:                GirilenDegerUzerindenHataMesajiGoster(sayi);
26:        }
27:    }
28:    Console.WriteLine("Programi kullandiginiz için teşekkürler. İyi günler");
29:    Environment.Exit(0);
30: }
```

1.2.1.1.9 Program.RabbitMQKuyrugayaGonder Method

Summary

Ekrandan alınan sayi kadar Kisi olusturup Rabbit MQ kuyruguna göndermeyi ve gönderilen kisiler ile ilgili bilgileri Konsol ekranında göstermeye yarayan metottur

C#

```
private static void RabbitMQKuyrugayaGonder(int kisiSayisi);
```

Parameters

Parameters	Description
int kisiSayisi	Kisi Sayisi Bilgisi

Body Source

```

1: private static void RabbitMQKuyrugagonder(int kisiSayisi)
2: {
3:     for (int i = 0; i < kisiSayisi; i++)
4:     {
5:         Kisi kisi = KisiBilgisiOlustur();
6:
7:         // Bilgisayara kurulan localhost RabbitMQ Sunucusuna baglanti için gerekli
bilgileri içeren nesnedir
8:         var factory = new ConnectionFactory() { HostName = "localhost" };
9:         using (IConnection connection = factory.CreateConnection())
10:        using (IModel channel = connection.CreateModel())
11:        {
12:            KisiyiKuyrugagonder(kisi, channel);
13:
14:            KuyrugagonderilenKisiBilgisiniMesajOlarakGoster(i, kisi);
15:        }
16:    }
17: }

```

1.2.1.1.10 Program.RabbitMQKuyruktanAl Method**Summary**

Rabbit MQ kuyrugunda biriken bilgilerin FIFO mantigina göre Kuyruktan alınarak Kisi bilgilerinin konsol ekranında görüntülenmesini sağlayan metottur

C#

```
private static void RabbitMQKuyruktanAl();
```

Body Source

```

1: private static void RabbitMQKuyruktanAl()
2: {
3:     // Bilgisayara kurulan localhost RabbitMQ Sunucusuna baglanti için gerekli
bilgileri içeren nesnedir
4:     var factory = new ConnectionFactory() { HostName = "localhost" };
5:     using (IConnection connection = factory.CreateConnection())
6:     using (IModel channel = connection.CreateModel())
7:     {
8:         KisiyiKuyruktanAl(channel);
9:     }
10:    Console.WriteLine("Kuyrugun tamamı tüketildi ve gösterildi. Devam etmek için bir
tusa basınız");
11:    Console.ReadLine();
12:    Console.WriteLine("-----");
13: }

```

1.3 Files

Summary

The following table lists files in this documentation.

Files

Name	Description
Kisi.cs (🔗 see page 9)	This is file Kisi.cs.
Program.cs (🔗 see page 9)	This is file Program.cs.
RabbitMQ.sln (🔗 see page 13)	This is file RabbitMQ.sln.

RabbitMQSample.csproj (see page 13)

This is file RabbitMQSample.csproj.

1.3.1 Kisi.cs

This is file Kisi.cs.

Body Source

```

1: ?namespace RabbitMQSample
2: {
3:     /// <summary>
4:     /// RabbitMQ teknolojisi ile gönderilecek ve alınacak verilerin içerisinde
    bulunduđu siniftir
5:     /// </summary>
6:     public class Kisi
7:     {
8:         /// <summary>
9:         /// Kisinin ID Bilgisi (Veri Tabanından geliyor gibi düşünebiliriz)
10:        /// </summary>
11:        public int ID { get; set; }
12:        /// <summary>
13:        /// Kisinin Adi Bilgisi
14:        /// </summary>
15:        public string Adi { get; set; }
16:        /// <summary>
17:        /// Kisinin Soyadi Bilgisi
18:        /// </summary>
19:        public string Soyadi { get; set; }
20:        /// <summary>
21:        /// Kisinin Dogum Tarihi Bilgisi
22:        /// </summary>
23:        public DateTime DogumTarihi { get; set; }
24:        /// <summary>
25:        /// Kisinin Dogum Yeri Bilgisi
26:        /// </summary>
27:        public string DogumYeri { get; set; }
28:    }
29: }
```

Namespaces

Name	Description
RabbitMQSample (see page 1)	This is namespace RabbitMQSample.

1.3.2 Program.cs

This is file Program.cs.

Body Source

```

1: ?using Bogus;
2: using Newtonsoft.Json;
3: using RabbitMQ.Client;
4: using RabbitMQ.Client.Events;
5: using RabbitMQSample;
6: using System.Text;
7:
8: /// <summary>
9: /// Rabbit MQ uygulamasinin konsolda çalishmasini saglayan metotlari içeren temel
    siniftir.
10: /// </summary>
```

```

11: public class Program
12: {
13:     /// <summary>
14:     /// Konsol uygulamasinin çalistigi ana metottur
15:     /// </summary>
16:     /// <param name="args">Gönderilen Argüman Dizizi</param>
17:     public static void Main(string[] args)
18:     {
19:         var programKapansinMi = false;
20:         // Kullanici Programdan çikmak için e veya E yazmadigi sürece çalismasini
saglayan döngüdür
21:         while (!programKapansinMi)
22:         {
23:             string? sayi = IlkVeriyiAl();
24:             // eger geçerli bir sayi girerse o sayi kadar Kisi kaydi olusturup
25:             // RabbitMQ kuyruguna yollayacak, alacak
26:             // Ve bu bilgileri konsol ekraninda yazdiracak kod blogu burasidir
27:             if (int.TryParse(sayi, out int kisiSayisi))
28:             {
29:                 RabbitMQKuyrugayaGonder(kisiSayisi);
30:                 KuyrugayaGondermeSonrasiEkrandaIlgiliMesajlariGoster();
31:                 RabbitMQKuyruktanAl();
32:             }
33:             else
34:             {
35:                 // Eger e veya E harfi girerse kullanıcı konsoldan
36:                 // programdan çikis yapilmasini saglayacak kod blogudur
37:                 if (sayi == "e" || sayi == "E")
38:                     programKapansinMi = true;
39:                 // eger geçersiz bir sayi veya metin girerse gösterilecek hata
mesajidir
40:                 else
41:                     GirilenDegerUzerindenHataMesajiGoster(sayi);
42:             }
43:         }
44:         Console.WriteLine("Programi kullandiginiz için tesekkürler. İyi günler");
45:         Environment.Exit(0);
46:     }
47:
48:     /// <summary>
49:     /// Ekrandan alinan sayi kadar Kisi olusturup
50:     /// Rabbit MQ kuyruguna göndermeyi ve gönderilen kisiler ile ilgili bilgileri
51:     /// Konsol ekraninda göstermeye yarayan metottur
52:     /// </summary>
53:     /// <param name="kisiSayisi">Kisi Sayisi Bilgisi</param>
54:     private static void RabbitMQKuyrugayaGonder(int kisiSayisi)
55:     {
56:         for (int i = 0; i < kisiSayisi; i++)
57:         {
58:             Kisi kisi = KisiBilgisiOlustur();
59:
60:             // Bilgisayara kurulan localhost RabbitMQ Sunucusuna baglanti için gerekli
bilgileri içeren nesnedir
61:             var factory = new ConnectionFactory() { HostName = "localhost" };
62:             using (IConnection connection = factory.CreateConnection())
63:             using (IModel channel = connection.CreateModel())
64:             {
65:                 KisiyiKuyrugayaGonder(kisi, channel);
66:
67:                 KuyrugayaGonderilenKisiBilgisiniMesajOlarakGoster(i, kisi);
68:             }
69:         }
70:     }
71:
72:     /// <summary>
73:     /// Rabbit MQ kuyrugunda biriken bilgilerin FIFO mantigina göre
74:     /// Kuyruktan alinarak
75:     /// Kisi bilgilerinin konsol ekraninda görüntülenmesini saglayan metottur
76:     /// </summary>

```

```

77:     private static void RabbitMQKuyruktanAl()
78:     {
79:         // Bilgisayara kurulan localhost RabbitMQ Sunucusuna baglanti için gerekli
bilgileri içeren nesnedir
80:         var factory = new ConnectionFactory() { HostName = "localhost" };
81:         using (IConnection connection = factory.CreateConnection())
82:         using (IModel channel = connection.CreateModel())
83:         {
84:             KisiyiKuyruktanAl(channel);
85:         }
86:         Console.WriteLine("Kuyrugun tamamı tüketildi ve gösterildi. Devam etmek için
bir tusa basınız");
87:         Console.ReadLine();
88:         Console.WriteLine("-----");
89:     }
90:
91:     /// <summary>
92:     /// Ekrandan bir sayı veya çıkış isteği olan e harfini alma ile ilgili metottur
93:     /// </summary>
94:     /// <returns>Sayı veya Çıkış İsteği</returns>
95:     private static string? İlkVeriyiAl()
96:     {
97:         Console.WriteLine("RabbitMQ kuyruguna kaç tane kayıt göndermek istiyorsunuz?
(Programdan çıkmak için E veya e yazınız)");
98:         var sayı = Console.ReadLine();
99:         return sayı;
100:     }
101:
102:     /// <summary>
103:     /// RabbitMQ kuyruguna veriler gönderildikten sonra
104:     /// Sistem tarafından ekranda gösterilen mesajların vb.
105:     /// Bulunduğu kod blogudur
106:     /// </summary>
107:     private static void KuyruğaGondermeSonrasıEkrandaİlgiliMesajlarıGoster()
108:     {
109:         Console.WriteLine("-----");
110:         Console.WriteLine("RabbitMQ kuyrugundan alma işlemleri için ekrana bir değer
giriniz");
111:         Console.ReadLine();
112:     }
113:
114:     /// <summary>
115:     /// Ekrandan girilen değer bir sayı veya e harfi değilse
116:     /// Geçerli bir değer olmadığı için
117:     /// Hata mesajı gösterilmesi ile ilgili kod blogudur
118:     /// </summary>
119:     /// <param name="metin">Ekrandan alınan değer</param>
120:     private static void GirilenDeğerÜzerindenHataMesajıGoster(string? metin)
121:     {
122:         Console.WriteLine($"{metin} geçerli bir sayı veya uygulama kapatma isteği
degildir");
123:         Console.WriteLine("-----");
124:     }
125:
126:     /// <summary>
127:     /// Bogus kütüphanesi kullanılarak
128:     /// Rastgele Kisi sınıfına ait Kisi bilgisi oluşturulması için
129:     /// Gerekli işlemlerin gerçekleştirildiği metottur.
130:     /// </summary>
131:     /// <returns>Kisi Bilgisi</returns>
132:     private static Kisi KisiBilgisiOluştur()
133:     {
134:         // Bogus kütüphanesi kullanılarak sahte veri içeren kisi kayıtlarının
135:         // otomatik olarak oluşturulmasını sağlayan yapıdır
136:         var kisiOluşturucu = new Faker<Kisi>()
137:             .CustomInstantiator(f => new Kisi())
138:             .RuleFor(u => u.Adı, f => f.Name.FirstName())
139:             .RuleFor(u => u.Soyadı, f => f.Name.LastName())
140:             .RuleFor(u => u.DogumYeri, (f, u) => f.Address.City())

```


```

141:         .RuleFor(u => u.DogumTarihi, (f, u) => f.Person.DateOfBirth)
142:         .RuleFor(u => u.ID, f => new Random().Next());
143:
144:         var kisi = kisiOlusturucu.Generate();
145:         return kisi;
146:     }
147:
148:     /// <summary>
149:     /// RabbitMQ kuyruguna gönderilen kisi bilgisinin
150:     /// Konsol tarafında gösterilmesi için
151:     /// Gerekli işlemlerin gerçekleştirildiği metottur.
152:     /// </summary>
153:     /// <param name="sayac">Döngüdeki sayaç bilgisi</param>
154:     /// <param name="kisi">Kisi Bilgisi</param>
155:     private static void KuyruGonderilenKisiBilgisiniMesajOlarakGoster(int sayac,
Kisi kisi)
156:     {
157:         Console.WriteLine($"Gönderilen kisi: Adi Soyadi: {kisi.Adi} {kisi.Soyadi}
Dogum Tarihi: {kisi.DogumTarihi.ToShortDateString()}");
158:         Console.WriteLine((sayac + 1) + ". kisi gönderildi...");
159:     }
160:
161:     /// <summary>
162:     /// RabbitMQ tarafında kuyruğu olusturup
163:     /// İlgili kuyruğa kisi bilgisinin gönderilmesi için
164:     /// Gerekli işlemlerin gerçekleştirildiği metottur.
165:     /// </summary>
166:     /// <param name="kisi">Kisi Bilgisi</param>
167:     /// <param name="channel">RabbitMQ Kanal Bilgisi</param>
168:     private static void KisiyiKuyruGonder(Kisi kisi, IModel channel)
169:     {
170:         // burada coderserdar adında bir kanal açarak, onun içerisine ilgili kisi
kayitlarinin gönderilmesi için hazırlamaktadır
171:         // diğer parametreler de kullanılarak kanalın kullanımı özelleştirilebilir
172:         channel.QueueDeclare(queue: "coderserdar", durable: false, exclusive: false,
autoDelete: false, arguments: null);
173:
174:         string message = JsonConvert.SerializeObject(kisi);
175:         // sınıf serialize edilerek byte dizisi haline dönüştürülür
176:         var body = Encoding.UTF8.GetBytes(message);
177:
178:         // burada dönüştürülen byte dizisinin kanala gönderilmesi sağlanır
179:         channel.BasicPublish(exchange: "", routingKey: "coderserdar", basicProperties:
null, body: body);
180:     }
181:
182:     /// <summary>
183:     /// RabbitMQ kuyrugundaki
184:     /// Kisi bilgilerini alıp
185:     /// Bu bilgilerin gösterilmesi için gerekli işlemlerin gerçekleştirildiği metottur.
186:     /// </summary>
187:     /// <param name="channel">RabbitMQ Kanal Bilgisi</param>
188:     private static void KisiyiKuyruktanAl(IModel channel)
189:     {
190:         // burada coderserdar adında bir kanal açarak, onun içerisine ilgili kisi
kayitlarinin gönderilmesi için hazırlamaktadır
191:         // diğer parametreler de kullanılarak kanalın kullanımı özelleştirilebilir
192:         channel.QueueDeclare(queue: "coderserdar", durable: false, exclusive: false,
autoDelete: false, arguments: null);
193:
194:         // burada kuyruğu tüketecek bir consumer nesnesi oluşturulur ve tüketmeye
baslar
195:         var consumer = new EventingBasicConsumer(channel);
196:         consumer.Received += (model, ea) =>
197:         {
198:             var body = ea.Body;
199:             var message = Encoding.UTF8.GetString(body.ToArray());
200:             Kisi kisi = JsonConvert.DeserializeObject<Kisi>(message);
201:             Console.WriteLine($"Adi Soyadi: {kisi.Adi} {kisi.Soyadi}

```

```
[{kisi.DogumYeri}]]");
202:         Console.WriteLine("RabbitMQ ile tanistiniz. İyi günler.");
203:     };
204:
205:     // kanalın kuyruğu tüketerek ilgili bilgilerin konsolda gösterilmesi sağlanır
206:     channel.BasicConsume(queue: "coderserdar", autoAck: true, consumer: consumer);
207:
208:     Console.ReadLine();
209: }
210: }
```

Classes

	Name	Description
	Program (see page 3)	Rabbit MQ uygulamasının konsolda çalışmasını sağlayan metotları içeren temel sınıftır.

1.3.3 RabbitMQ.sln

This is file RabbitMQ.sln.

1.3.4 RabbitMQSample.csproj

This is file RabbitMQSample.csproj.

Index

C

Classes 2

F

Files 8

K

Kisi class 1

about Kisi class 1

Adi 2

DogumTarihi 2

DogumYeri 2

ID 2

Soyadi 2

Kisi.cs 9

P

Program 3

Program class 3

about Program class 3

GirilenDegerUzerindenHataMesajiGoster 4

IlkVeriyiAl 4

KisiBilgisiOlustur 4

KisiyiKuyrugaGonder 5

KisiyiKuyruktanAl 5

KuyrugaGonderilenKisiBilgisiniMesajOlarakGoster 6

KuyrugaGondermeSonrasiEkranDallgiliMesajlariGoster 6

Main 7

RabbitMQKuyrugaGonder 7

RabbitMQKuyruktanAl 8

Program.cs 9

Program.GirilenDegerUzerindenHataMesajiGoster 4

Program.IlVeriyiAl 4

Program.KisiBilgisiOlustur 4

Program.KisiyiKuyrugaGonder 5

Program.KisiyiKuyruktanAl 5

Program.KuyrugaGonderilenKisiBilgisiniMesajOlarakGoster 6

Program.KuyrugaGondermeSonrasiEkranDallgiliMesajlariGoster 6

Program.Main 7

Program.RabbitMQKuyrugaGonder 7

Program.RabbitMQKuyruktanAl 8

R

RabbitMQ.sln 13

RabbitMQSample 1

RabbitMQSample namespace 1

Classes 1

RabbitMQSample.csproj 13

RabbitMQSample.Kisi 1

RabbitMQSample.Kisi.Adi 2

RabbitMQSample.Kisi.DogumTarihi 2

RabbitMQSample.Kisi.DogumYeri 2

RabbitMQSample.Kisi.ID 2

RabbitMQSample.Kisi.Soyadi 2