

```

0 references
internal class Program
{
    0 references
    private static void Main(string[] args)
    {
        Console.WriteLine("-----");
        BaseCustomerManager _starbucksCustomerManager = new StarbucksCustomerManager(new MernisServiceAdapter());
        _starbucksCustomerManager.Save(
            new Customer
            {
                DateOfBirth = new DateTime(1990, 1, 15),
                FirstName = "Atif Ertuğrul",
                LastName = "Kan",
                NationalityId = 40399173916
            });
        Console.WriteLine("-----");
        BaseCustomerManager _nerocustomerManager = new NeroCustomerManager();
        _nerocustomerManager.Save
        (
            new Customer()
            {
                DateOfBirth = new DateTime(1991, 02, 19),
                FirstName = "Gözde",
                LastName = "Kan",
                NationalityId = 20225242368
            });
        Console.WriteLine("-----");
    }
}

```

```

namespace ConsoleUI.Business.Abstract
{
    1 reference
    public interface ICustomerService
    {
        4 references
        void Save(Customer customer);
    }
}

```

```

namespace ConsoleUI.Business.Abstract
{
    3 references
    public abstract class BaseCustomerManager : ICustomerService
    {
        4 references
        public virtual void Save(Customer customer)
        {
            Console.WriteLine("Yeni Müşterimizi : [" + customer.FirstName + " " + customer.LastName + "] VeriTabanına kayıt edildi.");
        }
    }
}

```

```

namespace ConsoleUI.Business.Concrete
{
    0 references
    public class NeroCustomerManager : BaseCustomerManager
    {
        // firmadan doğrulama talebi gelirse buraya da eklenecek
    }
}

```

```

namespace ConsoleUI.Business.Concrete
{
    2 references
    public class StarbucksCustomerManager : BaseCustomerManager
    {
        IPersonCheckService _personCheckService;

        1 reference
        public StarbucksCustomerManager(IPersonCheckService personCheckService)
        {
            _personCheckService = personCheckService;
        }

        4 references
        public override void Save(Customer customer)
        {
            if (_personCheckService.ChackIfRealPerson(customer))
            {
                base.Save(customer);
            }
            else
            {
                throw new Exception("Geçerli bir kişi değil");
            }
        }
    }
}

```

```

namespace ConsoleUI.Adapters
{
    4 references
    public interface IPersonCheckService
    {
        3 references
        public bool ChackIfRealPerson(Customer customer)
        {
            return true;
        }
    }
}

```

```

namespace ConsoleUI.Adapters
{
    1 reference
    public class MernisServiceAdapter : IPersonCheckService
    {
        2 references
        public bool ChackIfRealPerson(Customer customer)
        {
            KPSPublicSoapClient mernisClient = new KPSPublicSoapClient(EndpointConfiguration.KPSPublicSoap);
            TCKimlikNoDogrulaResponse tcKimlikDogrulamaServisResponse = mernisClient.TCKimlikNoDogrulaAsync
            (
                customer.NationalityId,
                customer.FirstName,
                customer.LastName,
                customer.DateOfBirth.Year
            ).GetAwaiter().GetResult();
            return tcKimlikDogrulamaServisResponse.Body.TCKimlikNoDogrulaResult;
        }
    }
}

```