1)add  
2)после добавления пользователя, список должен быть увеличен на один  
3)список не увеличился  
4)мы никуда не добавили  
case "add":

Console.Write("Name: ");

string name = Console.ReadLine();

Console.Write("Phone: ");

string phone = Console.ReadLine();

Console.WriteLine($"{name} added to employees");

break;  
5)исправление  
// Создаем новый объект EmployeesDTO

EmployeesDTO newEmployee = new EmployeesDTO(name, phone);

// Добавляем его в список сотрудников

employeesList.Add(newEmployee);

Console.WriteLine($"{name} added to employees");

break;  
1)запуск программы  
2)должны добавиться номера  
3)номера не добавились  
4) метод AddPhone принимает строку phone в качестве параметра, но не используется внутри метода. Вместо этого, добавляется в словарь \_phones значение свойства Phone из объекта employee, которое всегда равно последнему элементу словаря или «-« . Это приводит к тому, что не сохраняется реальный номер телефона, а только дублируется существующий или добавляется «-«.Чтобы исправить это, нужно использовать параметр phone в методе AddPhone и добавлять его в словарь \_phones. Также, вам нужно изменить свойство Phone так, чтобы оно возвращало последний номер телефона из словаря \_phones или null, если словарь пустой.

public partial class EmployeesDTO : DataTransferObject

{

public string Name { get; set; }

public string Phone {

get { return \_phones.Any() ? \_phones.Last().Value : null; }

}

public void AddPhone(string phone)

{

if(String.IsNullOrEmpty(phone))

{

throw new Exception("Phone must be provided!");

}

\_phones.Add(DateTime.Now, phone);

}

private List<KeyValuePair<DateTime, string>> \_phones = new List<KeyValuePair<DateTime, string>>();

}

КОД ПЛАГИНА  
using PhoneApp.Domain.DTO;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace GeneratorEmployeePlugin

{

public class UserDTO : DataTransferObject

{

public int id { get; set; }

public string name { get; set; }

public string phone { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Text;

using System.Threading.Tasks;

using PhoneApp.Domain;

using PhoneApp.Domain.Attributes;

using PhoneApp.Domain.DTO;

using PhoneApp.Domain.Interfaces;

namespace GeneratorEmployeePlugin

{

[Author(Name = "Vadim Bondarenko")]

public class Plugin : IPluggable

{

private static readonly NLog.Logger logger = NLog.LogManager.GetCurrentClassLogger();

public IEnumerable<DataTransferObject> Run(IEnumerable<DataTransferObject> args)

{

logger.Info("Generating employees");

using (HttpClient client = new HttpClient())

{

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

HttpResponseMessage response = client.GetAsync("https://jsonplaceholder.typicode.com/users").Result;

if (response.IsSuccessStatusCode)

{

string json = response.Content.ReadAsStringAsync().Result;

var usersList = Newtonsoft.Json.JsonConvert.DeserializeObject<List<UserDTO>>(json);

var employeesList = usersList.Select(u => new EmployeesDTO

{

Name = u.name,

}).ToList();

foreach (var employee in employeesList)

{

employee.AddPhone(employee.Phone);

}

logger.Info($"Generated {employeesList.Count()} employees");

return employeesList.Cast<DataTransferObject>();

}

else

{

logger.Error($"API request failed: {response.ReasonPhrase}");

return null;

}

}

}

}

}