```
 1 namespace CoffeeShopSample.Logic
 2 {
 3     public class CoffeeType
 4     {
 5         public int CoffeeTypeId { get; set; }
 6         public string Name { get; set; }
 7         public string CountryOfOrigin { get; set; }
 8         public decimal Price { get; set; }
 9     }
10 }
```

```
 1 using System.Data.Entity;
 2
 3 namespace CoffeeShopSample.Logic
 4 {
 5     public class AppDbContext : DbContext
 6     {
 7         public AppDbContext(string connectionString):base(connectionString)
 8         {
 9
10         }
11         public IDbSet<CoffeeType> CoffeeTypes { get; }
12         public IDbSet<CoffeePreference> CoffeePreferences { get; }
13     }
14 }
```

```
1 namespace CoffeeShopSample.Logic
2 {
3     public class CoffeePreference
4     {
5         public int AustralianCoffeeSearch { get; set; }
6     }
7 }
```

```
 1 using System.Collections.Generic;
 2 using System.Linq;
 3
 4 namespace CoffeeShopSample.Logic
 5 {
 6     public class ExampleDataAccessLayer
 7     {
 8         private readonly AppDbContext _context;
 9
10         public ExampleDataAccessLayer(AppDbContext context)
11         {
12             _context = context;
13         }
14
15         public IEnumerable<CoffeeType> GetVietnameseCoffeeTypes()
16         {
17             return _context.CoffeeTypes.Where(ct => ct.CountryOfOrigin == "Vietnam");
18         }
19
20         public decimal GetAmericanCoffeePrice()
21         {
22             var americanCoffees = _context.CoffeeTypes
23                 .Where(ct => ct.CountryOfOrigin == "America")
24                 .ToList();
25
26             return americanCoffees[0].Price;
27         }
28
29
30
31
32
33
34
35
```

```csharp
36
37         public IEnumerable<CoffeeType> SearchAustralianCoffeeTypes()
38         {
39             var types = GetAllCoffeeTypes()
40                 .Where(ct => ct.CountryOfOrigin == "Australia")
41                 .ToList();
42
43             var cp = _context.CoffeePreferences.SingleOrDefault();
44             cp.AustralianCoffeeSearch++;
45             _context.SaveChanges();
46
47             return types;
48         }
49
50         private IEnumerable<CoffeeType> GetAllCoffeeTypes()
51         {
52             return _context.CoffeeTypes.ToList();
53         }
54
55         public void IncreasePrices(IEnumerable<CoffeeType> coffeeTypes, double percentage)
56         {
57             foreach (var coffeeType in coffeeTypes)
58             {
59                 var type = _context.CoffeeTypes.Single(ct => ct.CoffeeTypeId == coffeeType.CoffeeTypeId);
60                 type.Price *= 1 + (decimal) percentage;
61                 _context.SaveChanges();
62             }
63         }
64     }
65 }
```