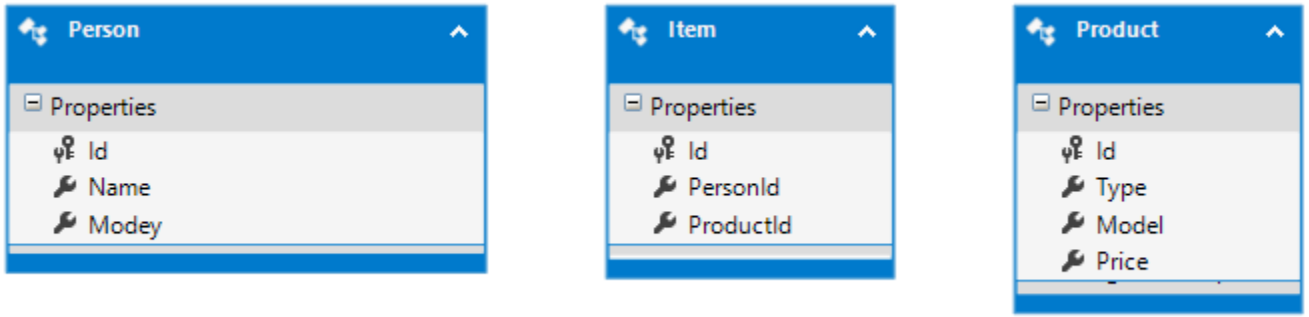
LINQ

Данни


Person
Properties
Id
Name
Modey
Item
Properties
Id
PersonId
ProductId
Product
Properties
Id
Type
Model
Price

Данни

```
var products = new Product[]
{
    new Product { Id = 1, Type = "Phone", Model = "OnePlus", Price = 1000 },
    new Product { Id = 2, Type = "Phone", Model = "Apple", Price = 2000 },
    new Product { Id = 3, Type = "Phone", Model = "Samsung", Price = 1500 },
    new Product { Id = 4, Type = "TV", Model = "Samsung 32", Price = 200 },
    new Product { Id = 5, Type = "TV", Model = "Samsung 64", Price = 2000 },
    new Product { Id = 6, Type = "TV", Model = "Samsung 64 OLED", Price = 7000 },
    new Product { Id = 7, Type = "Car", Model = "VW Golf", Price = 30000 },
    new Product { Id = 8, Type = "Car", Model = "VW Passat", Price = 60000 },
    new Product { Id = 9, Type = "Car", Model = "Audi A8", Price = 120000 }
};

var people = new Person[]
{
    new Person{ Id = 1, Name = "Ivan Ivanov", Money = 150000 },
    new Person{ Id = 2, Name = "Dragan Draganov", Money = 250000 },
    new Person{ Id = 3, Name = "Ivelin Ivelinov", Money = 350000 }
};

var items = new Item[]
{
    new Item{ PersonId = 1, ProductId = 1, Amount = 1 },
    new Item{ PersonId = 1, ProductId = 4, Amount = 1 },
    new Item{ PersonId = 1, ProductId = 5, Amount = 1 },
    new Item{ PersonId = 1, ProductId = 7, Amount = 1 },
    new Item{ PersonId = 2, ProductId = 2, Amount = 1 },
    new Item{ PersonId = 2, ProductId = 6, Amount = 1 },
    new Item{ PersonId = 2, ProductId = 7, Amount = 1 },
    new Item{ PersonId = 2, ProductId = 9, Amount = 1 },
    new Item{ PersonId = 3, ProductId = 3, Amount = 1 },
    new Item{ PersonId = 3, ProductId = 8, Amount = 1 },
    new Item{ PersonId = 3, ProductId = 7, Amount = 1 }
};
```

Материализиране

```
var firstProdust = products
    .FirstOrDefault();


var lastProduct = products
    .Last();


var productArray = products
    .ToArray();

var productArray = products
    .ToList();

var productArray = products
    .Single();
```

Условие (Where)

```
var cheapProducts = products
    .Where(c => c.Price < 10000)
    .ToArray();

foreach(var cheapProduct in cheapProducts)
{
    Console.WriteLine("Type: " + cheapProduct.Type + "Model: " + cheapProduct.Model);
}
```

Сортиране (OrderBy, ThenBy)

```
var orderedCheapProducts = products
    .Where(c => c.Price < 10000)
    .OrderBy(o => o.Price)
    .ToArray();

foreach(var cheapProduct in orderedCheapProducts)
{
    Console.WriteLine("Type: " + cheapProduct.Type + "Model: " + cheapProduct.Model);
}
```

Пропускане, подбиране (Skip, Take)

```
var orderedCheapProducts2 = products
    .OrderBy(o => o.Type)
    .ThenBy(o => o.Model)
    .Skip(1)
    .Take(2)
    .ToArray();
```

Проекция (Select)

```
var orderedCheapProductsText = products
    .Where(c => c.Price < 10000)
    .OrderBy(o => o.Price)
    .Select(s => "Type: " + s.Type + "Model: " + s.Model)
    .ToArray();

foreach(var cheapProductText in orderedCheapProductsText)
{
    Console.WriteLine(cheapProductText);
}
```

Агрегиране (Aggregate)

```
var cheapProductsText = products
    .Where(c => c.Price < 10000)
    .Select(s => "Type: " + s.Type + "Model: " + s.Model + Environment.NewLine)
    .Aggregate((f, s) => f + s);

Console.WriteLine(cheapProductsText);
```

Сума (Sum)

```
var allProductPrices = products
    .Select(s => s.Price)
    .Aggregate((f, s) => f + s);

var allProductPrices2 = products
    .Sum(s => s.Price);
```

Минимална / Максимална стойност (Min / Max)

```
var mostExpensivePrice = products
    .Max(m => m.Price);
```

Средна стойност (Average)

```
var allProductPricesAverage = products
    .Average(s => s.Price);
```

Уникалност (Distinct)

```
var productCategories = products
    .Select(s => s.Model)
    .Distinct();
```

Групиране (GroupBy)

```
var productsByCategories = products
    .GroupBy(g => g.Type)
    .ToArray();

foreach (var productsByCategory in productsByCategories)
{
    Console.WriteLine(productsByCategory.Key);

    foreach (var product in productsByCategory)
    {
        Console.WriteLine("\t" + product.Model);
    }
}
```

Свързване „едно към едно“ (Join)

```
var personItems = people
    .Join(
        items,
        o => o.Id,
        i => i.PersonId,
        (o, i) => new
        {
            Person = o,
            ProductId = i.ProductId
        })
    .ToArray();
```

Свързване „едно към много“ (GroupJoin)

```
var personItems = people
    .GroupJoin(
        items,
        o => o.Id,
        i => i.PersonId,
        (o, i) => new
        {
            Person = o,
            ProductsIds = i
                .Select(s => s.ProductId)
                .ToArray()
        })
    .ToArray();
```

Множествена проекция (SelectMany)

```
var personItemsArr = people
    .GroupJoin(
        items,
        o => o.Id,
        i => i.PersonId,
        (o, i) => new
        {
            Person = o,
            Items = i
        })
    .SelectMany(m => m.Items
        .Select(s => new
        {
            Person = m.Person,
            Item = s
        }))
    .ToArray();
```

Задача

- Да се състави LINQ израз връщащ всички лица и притежаваните от тях продукти. Резултатът да е във вид на масив, всеки елемент на който е анонимен тип притежаващ свойство Person от тип Person и свойство Products от тип Product[]

Задача

```
var personItems = people
    .GroupJoin(
        items,
        o => o.Id,
        i => i.PersonId,
        (o, i) => new
        {
            Person = o,
            Products = i
                .Join(
                    products,
                    o2 => o2.ProductId,
                    i2 => i2.Id,
                    (o2, i2) => i2)
                .ToArray()

        })
    .ToArray();
```

Задача

- Да се състави LINQ израз връщащ всички лица и притежаваните от тях продукти. Резултатът да е във вид на текстов низ.

Задача

```
var personItems = people
    .GroupJoin(
        items,
        o => o.Id,
        i => i.PersonId,
        (o, i) => new
        {
            PersonName = o.Name,
            Products = i
                .Join(
                    products,
                    o2 => o2.ProductId,
                    i2 => i2.Id,
                    (o2, i2) => i2)
                .Select(s => "\t" + s.Model + Environment.NewLine)
                .Aggregate((f, s) => f + s)

        })
    .Select(s => s.PersonName + Environment.NewLine + s.Products)
    .Aggregate((f, s) => f + s);
```

Задача

- Да се състави LINQ израз връщащ всички лица и притежаваните от тях продукти. Резултатът да е във вид на масив, всеки елемент на който е анонимен тип притежаващ свойство Person от тип Person и свойство Products от тип Product[]
- Да не се използват методите Join и GroupJoin

Задача

```
var personItems = people
    .Select(s => new
    {
        Person = s,
        Products = items
            .Where(c => c.PersonId == s.Id)
            .Select(s2 => products
                .Single(s3 => s3.Id == s2.ProductId))
            .ToArray()
    })
    .ToArray();
```

Задача

- Да се състави LINQ израз връщащ за всяко лице общата стойност на пари и притежавани вещи. Резултатът да е във вид на анонимен ти със свойство Person от тип Person и свойство Value от тип Decimal.

Задача

```
var peopleValue = people
    .Select(s => new
    {
        Person = s,
        Value =
            s.Money +
            items
                .Where(c => c.PersonId == s.Id)
                .Select(s2 => products
                    .Where(c => c.Id == s2.ProductId)
                    .Select(s3 => s3.Price * s2.Amount)
                    .Single())
                .Sum()
    })
    .ToArray();
```

Задача

- Да се състави LINQ израз връщащ за всяко лице какви продукти би могъл да си закупи със притежаваната от него сума (за всеки продукт бройка). Резултатът да е във вид на масив от анонимен тип, притежаващ свойство Person от тип Person и свойство Products от тип String[]

Задача

```
var personProducts = people
    .Select(s => new
    {
        Person = s,
        Products = products
            .Select(s2 =>
                " Type:" + s2.Type +
                " Model:" + s2.Model +
                " Count:" + ((Int32)(s.Money / s2.Price)).ToString())
            .ToArray()
    })
    .ToArray();
```

Задача

- Да се състави LINQ израз връщащ за всеки тип продукт, кои лица го притежават. Резултатът да е във вид на анонимен тип притежаващ свойство Type от тип String и свойство People от тип Person[].

Задача

```
var productTypes = items
    .Select(s => new
    {
        Product = products
            .Where(c => c.Id == s.ProductId)
            .Single(),
        Person = people
            .Where(c => c.Id == s.PersonId)
            .Single()
    })
    .GroupBy(g => g.Product.Type)
    .Select(s => new
    {
        Type = s.Key,
        People = s
            .Select(s2 => s2.Person)
            .Distinct()
            .ToArray()
    })
    .ToArray();
```