

Текст программы:

```
namespace rk1
{
    public class Detail
    {
        public int Id { get; set; }

        public string Name { get; set; }

        public decimal Price { get; set; }

        public int SupplierId { get; set; }

        public Detail(int id, string name, decimal price, int supplierId)
        {
            Id = id;

            Name = name;

            Price = price;

            SupplierId = supplierId;
        }
    }

    public class Supplier
    {
        public int Id { get; set; }

        public string Name { get; set; }

        public Supplier(int id, string name)
        {
            Id = id;

            Name = name;
        }
    }

    public class DetailSupplier
    {
        public int DetailId { get; set; }
```

```

public int SupplierId { get; set; }

public DetailSupplier(int detailId, int supplierId)
{
    DetailId = detailId;
    SupplierId = supplierId;
}
}

internal class Program
{
    static void Main(string[] args)
    {
        var suppliers = new List<Supplier>
        {
            new(1, "Основной отдел поставщиков"),
            new(2, "Вспомогательный отдел комплектующих"),
            new(3, "Склад запчастей болты и гайки"),
            new(4, "Другой отдел поставщиков"),
            new(5, "отдел перекупа деталей"),
            new(6, "Поставщик про100деталь")
        };

        var details = new List<Detail>
        {
            new(1, "Болт М8", 15.50m, 1),
            new(2, "Гайка М10", 8.75m, 2),
            new(3, "Шайба А4", 3.20m, 3),
            new(4, "Винт М6", 12.80m, 3),
            new(5, "Подшипник 6205", 45.90m, 3)
        };

        var detailSuppliers = new List<DetailSupplier>
        {
            new(1, 1),

```

```
new(2, 2),
new(3, 3),
new(4, 3),
new(5, 3),
new(1, 4),
new(2, 5),
new(3, 6),
new(4, 6),
new(5, 6)
};
```

```
// Соединение данных многие-ко-многим
```

```
var manyToManyTemp = from supplier in suppliers
    join ds in detailSuppliers on supplier.Id equals ds.SupplierId
    select new { SupplierName = supplier.Name, ds.SupplierId, ds.DetailId };
```

```
Console.WriteLine("Задание E1");
```

```
Console.WriteLine("Список всех поставщиков, у которых в названии присутствует слово  
\"отдел\", и список их деталей:");
```

```
var res1 = from supplier in suppliers
    where supplier.Name.Contains("отдел")
    join ds in detailSuppliers on supplier.Id equals ds.SupplierId into supplierDetails
    from sd in supplierDetails
    join detail in details on sd.DetailId equals detail.Id
    group detail by supplier into supplierGroup
    select new
    {
        SupplierName = supplierGroup.Key.Name,
        Details = supplierGroup.Select(d => new { d.Name, d.Price }).ToList()
    };
};
```

```
foreach (var item in res1)
{
```

```

Console.WriteLine($"Поставщик: {item.SupplierName}");

foreach (var detail in item.Details)
{
    Console.WriteLine($" Деталь: {detail.Name}, Цена: {detail.Price} руб.");
}
}

```

```

Console.WriteLine("\nЗадание E2");

```

```

Console.WriteLine("Список поставщиков со средней ценой деталей в каждом отделе,
отсортированный по средней цене:");

```

```

var res2Unsorted = from supplier in suppliers
    join ds in detailSuppliers on supplier.Id equals ds.SupplierId
    join detail in details on ds.DetailId equals detail.Id
    group detail by supplier into supplierGroup
    let avgPrice = Math.Round(supplierGroup.Average(d => d.Price), 2)
    select new
    {
        SupplierName = supplierGroup.Key.Name,
        AvgPrice = avgPrice,
        DetailsCount = supplierGroup.Count()
    };

```

```

var res2 = res2Unsorted.OrderBy(x => x.AvgPrice);

```

```

foreach (var item in res2)
{
    Console.WriteLine($"Поставщик: {item.SupplierName}, Средняя цена: {item.AvgPrice}
руб. (детали: {item.DetailsCount})");
}

```

```

Console.WriteLine("\nЗадание E3");

```

```
Console.WriteLine("Список всех деталей, у которых название начинается с буквы \"Б\", и  
названия их поставщиков:");
```

```
var res3 = from detail in details  
           where detail.Name.StartsWith("Б")  
           join temp in manyToManyTemp on detail.Id equals temp.DetailId into  
detailSuppliersTemp  
           from dst in detailSuppliersTemp  
           join supplier in suppliers on dst.SupplierId equals supplier.Id  
           select new  
{  
    DetailName = detail.Name,  
    DetailPrice = detail.Price,  
    SupplierName = supplier.Name  
};  
  
foreach (var item in res3)  
{  
    Console.WriteLine($"Деталь: {item.DetailName}, Цена: {item.DetailPrice} руб. ->  
Поставщик: {item.SupplierName}");  
}  
}  
}
```

Результаты:

```
Задание E1  
Список всех поставщиков, у которых в названии присутствует слово "отдел", и список их деталей:  
  
Поставщик: Основной отдел поставщиков  
Деталь: Болт М8, Цена: 15,50 руб.  
  
Поставщик: Вспомогательный отдел комплектующих  
Деталь: Гайка М10, Цена: 8,75 руб.  
  
Поставщик: Другой отдел поставщиков  
Деталь: Болт М8, Цена: 15,50 руб.  
  
Поставщик: отдел перекупа деталей  
Деталь: Гайка М10, Цена: 8,75 руб.  
  
Задание E2  
Список поставщиков со средней ценой деталей в каждом отделе, отсортированный по средней цене:  
Поставщик: Вспомогательный отдел комплектующих, Средняя цена: 8,75 руб. (детали: 1)  
Поставщик: отдел перекупа деталей, Средняя цена: 8,75 руб. (детали: 1)  
Поставщик: Основной отдел поставщиков, Средняя цена: 15,50 руб. (детали: 1)  
Поставщик: Другой отдел поставщиков, Средняя цена: 15,50 руб. (детали: 1)  
Поставщик: Склад запчастей болты и гайки, Средняя цена: 20,63 руб. (детали: 3)  
Поставщик: Поставщик про100деталь, Средняя цена: 20,63 руб. (детали: 3)  
  
Задание E3  
Список всех деталей, у которых название начинается с буквы "Б", и названия их поставщиков:  
Деталь: Болт М8, Цена: 15,50 руб. -> Поставщик: Основной отдел поставщиков  
Деталь: Болт М8, Цена: 15,50 руб. -> Поставщик: Другой отдел поставщиков
```