

Для выборки применяется метод **Where**. Выберем из бд все модели, производитель которых - "Samsung":

```
1 using(PhoneContext db = new PhoneContext())
2 {
3     var phones = db.Phones.Where(p=> p.Company.Name=="Samsung");
4     foreach (Phone p in phones)
5         Console.WriteLine("{0}.{1} - {2}", p.Id, p.Name, p.Price);
6 }
```

Для выборки одного объекта мы можем использовать метод **Find()**. Данный метод не является методом Linq, он определен у класса **DbSet**:

```
1 Phone myphone = db.Phones.Find(3); // выберем элемент с id=3
```

Но в качестве альтернативы мы можем использовать методы Linq **First()/FirstOrDefault()**. Они получают первый элемент выборки, который соответствует определенному условию. Использование метода **FirstOrDefault()** является более гибким, так как если выборка пуста, то он вернет значение null. А метод **First()** в той же ситуации выбросит ошибку.

```
1 Phone myphone = db.Phones.FirstOrDefault(p=>p.Id==3);
2 if(myphone!=null)
3     Console.WriteLine(myphone.Name);
```

Теперь сделаем проекцию. Допустим, нам надо добавить в результат выборки название компании. Мы можем использовать метод **Include** для подсоединения к объекту связанных данных из другой таблицы: `var phones = db.Phones.Include(p=>p.Company)`. Но не всегда нужны все свойства выбираемых объектов. В этом случае мы можем применить метод **Select** для проекции извлеченных данных на новый тип:

```
1 using(PhoneContext db = new PhoneContext())
2 {
3     var phones = db.Phones.Select(p => new
4     {
5         Name = p.Name,
6         Price = p.Price,
7         Company = p.Company.Name
8     });
9     foreach (var p in phones)
10         Console.WriteLine("{0} ({1}) - {2}", p.Name, p.Company, p.Price);
11 }
```

В итоге метод **Select** из полученных данных спроецирует новый тип. В данном случае мы получим данные анонимного типа, но это также может быть определенный пользователем тип. Например:

```
1 public class Model
2 {
3     public string Name { get; set; }
4     public string Company { get; set; }
5     public int Price { get; set; }
6 }
```

И спроецируем выборку на этот тип:

```
1 var phones = db.Phones.Select(p => new Model
2 {
3     Name = p.Name,
4     Price = p.Price,
5     Company = p.Company.Name
6 });
7 foreach (Model p in phones)
8     Console.WriteLine("{0} ({1}) - {2}", p.Name, p.Company, p.Price);
```