**Вопрос 1** **Выберите верное утверждение о данных строках кода:**

**a) var result = Customers.Where(c=>c.Name.StartsWith("O"));**

**b) var result = Customers.Where(c=>c.Name.StartsWith("O")).ToArray();**

Разница только в представлении данных. И в одном и в другом случае происходите немедленное формирование набора данных в переменной result

Обе строки содержат синтаксические ошибки

Нет никакой разницы между строками a) и b)

Разница в механизме выполнения. В первой строке используется отложенное выполнение, во второй строке немедленное выполнение, так как вызывается метод ToArray

**Ответ:** Разница в механизме выполнения. В первой строке используется отложенное выполнение, во второй строке немедленное выполнение, так как вызывается метод ToArray

**Вопрос 2** **Выберите верные утверждения об механизме отложенного выполнения в LINQ:**

Узнать о том обладает ли метод механизмом можно с использованием документации. Например, найти соответствующий раздел в MSDN

Все методы LINQ работают только по механизму отложенного исполнения

Отложенное выполнение это устаревший механизм и его оставили только для совместимости

Отложенное выполнение означает, что вычисление результата выражения откладывается до тех пор, пока это не будет необходимо

Узнать о том обладает ли метод механизмом отложенного выполнения невозможно

В LINQ нет такого понятия, как отложенное выполнение

**Ответ:** Узнать о том обладает ли метод механизмом можно с использованием документации. Например, найти соответствующий раздел в MSDN   
Отложенное выполнение означает, что вычисление результата выражения откладывается до тех пор, пока это не будет необходимо

**Вопрос 3** **Выберите верные утверждения о данном фрагменте кода:**

**var result = Customers.Where(c=>c.Name.StartsWith("O"));**

**int count = result.Count();**

**Console.WriteLine(count);**

**foreach(var t in result){**

**Console.WriteLine(t.Name+" "+t.LastName);**

**}**

В этом коде есть потенциальные проблемы с производительностью. Вызов метода Count порождает проход по всем данным внутри источника данных

Повторный проход по данным осуществляется внутри foreach

Этот фрагмент кода не содержит синтаксических ошибок

Не рекомендуется использовать Count. Данный метод объявлен устаревшим

В данном коде содержится синтаксическая ошибка

**Ответ:** В этом коде есть потенциальные проблемы с производительностью. Вызов метода Count порождает проход по всем данным внутри источника данных   
Этот фрагмент кода не содержит синтаксических ошибок

**Вопрос 4** **Выберите существующие ключевые слова LINQ:**

into

right outer join

left outer join

inner join

join

let

**Ответ:** into   
join   
let

**Вопрос 5** **Выберите верное утверждения о данном фрагменте кода:**

**from customer in Customers**

**join sale in Sales**

**on customer.Id equals sale.IdCust**

**select distinct customer**

Этот фрагмент кода не содержит синтаксических ошибок

В данном коде содержится синтаксическая ошибка, так как вместо join нужно использовать inner join

В данном коде содержится синтаксическая ошибка, так как distinct не существует в виде ключевого слова в синтаксисе LINQ

В данном коде содержится синтаксическая ошибка, так как вместо equals нужно использовать =

**Ответ:** В данном коде содержится синтаксическая ошибка, так как distinct не существует в виде ключевого слова в синтаксисе LINQ

**Вопрос 6** **Выберите вариант ответа, который заставит LINQ-запрос немедленно исполниться:**

ToList

WHERE

SELECT

ORDERBY

**Ответ:** ToList

**Вопрос 7**: **Выберите верное утверждения о данном фрагменте кода:**

**from customer in Customers**

**join sale in Sales**

**on customer.Id = sale.IdCust**

**select customer**

В данном коде содержится синтаксическая ошибка, так как вместо = нужно использовать contains

В данном коде содержится синтаксическая ошибка, так как вместо = нужно использовать equals

В данном коде содержится синтаксическая ошибка, так как вместо join нужно использовать inner join

Этот фрагмент кода не содержит синтаксических ошибок

**Ответ:** В данном коде содержится синтаксическая ошибка, так как вместо = нужно использовать equals

**Вопрос 8**: **Выберите тот фрагмент кода, который будет решать ту же задачу, что и исходный код и не содержать синтаксических ошибок:**

**from sale in Sales**

**join customer in Customers on sale.IdCust equals customer.Id**

**join salesman in SalesMen on sale.IdSal equals salesman.Id**

**select new{**

**CustomerLastName = customer.LastName,**

**SalesmanLastName = salesman.LastName,**

**sale.Product**

**}**

Sales.InnerJoin(

SalesMen, sal=>sal.IdSal, s=>s.Id, (sal,s)=>new{

sal,

s

}).InnerJoin(Customers,fq=>fq.sal.IdCust, c=>c.Id, (sal,c) => new{

sal,

c

}).Select(

r=>new{

Customer = r.c.LastName,

Salesman = r.sal.s.LastName

}

)

Sales.CrossJoin(

SalesMen, sal=>sal.IdSal, s=>s.Id, (sal,s)=>new{

sal,

s

}).CrossJoin(Customers,fq=>fq.sal.IdCust, c=>c.Id, (sal,c) => new{

sal,

c

}).Select(

r=>new{

Customer = r.c.LastName,

Salesman = r.sal.s.LastName

}

)

Sales.LeftJoin(

SalesMen, sal=>sal.IdSal, s=>s.Id, (sal,s)=>new{

sal,

s

}).RightJoin(Customers,fq=>fq.sal.IdCust, c=>c.Id, (sal,c) => new{

sal,

c

}).Select(

r=>new{

Customer = r.c.LastName,

Salesman = r.sal.s.LastName

}

)

Sales.Join(

SalesMen, sal=>sal.IdSal, s=>s.Id, (sal,s)=>new{

sal,

s

}).Join(Customers,fq=>fq.sal.IdCust, c=>c.Id, (sal,c) => new{

sal,

c

}).Select(

r=>new{

Customer = r.c.LastName,

Salesman = r.sal.s.LastName

}

)

**Ответ:** Sales.Join(  
SalesMen, sal=>sal.IdSal, s=>s.Id, (sal,s)=>new{  
 sal,  
 s  
}).Join(Customers,fq=>fq.sal.IdCust, c=>c.Id, (sal,c) => new{  
 sal,  
 c  
}).Select(  
r=>new{  
 Customer = r.c.LastName,  
 Salesman = r.sal.s.LastName  
}  
)

**Вопрос 9**: **Данный фрагмент код должен получить набор данных, отсортировать полученные данные сначала по имени по возрастанию, а затем отсортировать по фамилии по убыванию.**

**Выберите верное утверждения о данном фрагменте кода относительно полученной информации:**

**Customers.Join(Sales, c=>c.Id, s => s.IdCust,**

**(c,s) => new{**

**c.Name,**

**c.LastName,**

**}**

**).Distinct().OrderBy(t=>t.Name).OrderByDescending(p=>p.LastName)**

Этот фрагмент кода не содержит синтаксических ошибок, но вместо OrderByDescending нужно использовать ThenByDescending, иначе вторая сортировка по убыванию будет работать некорректно

Этот фрагмент кода не содержит синтаксических ошибок, но вместо OrderBy нужно использовать ThenBy, иначе первая сортировка по возрастанию будет работать некорректно

В данном коде содержится синтаксическая ошибка, так как OrderByDescending не существует в виде отдельного метода

В данном коде содержится синтаксическая ошибка, так как вместо метода Distinct нужно использовать ключевое слово distinct

В данном коде содержится синтаксическая ошибка, так как вместо OrderBy нужно использовать ThenBy

**Ответ:** Этот фрагмент кода не содержит синтаксических ошибок, но вместо OrderByDescending нужно использовать ThenByDescending, иначе вторая сортировка по убыванию будет работать некорректно

**Вопрос 10** **Выберите верные утверждения о данном фрагменте кода:**

**var result = Enumerable.Empty<Customers>();**

**try{**

**result = Customers.Where(c=>c.Name.StartsWith("O"));**

**}**

**catch(Exception ex){**

**Console.WriteLine(ex.Message);**

**}**

**foreach(var t in result){**

**Console.WriteLine(t.Name+" "+t.LastName);**

**}**

Механизм обработки исключений в данном фрагменте кода реализован неправильно. Проблема в том, что здесь используется механизм отложенного выполнения и реальное получение данных произойдет только в foreach, где не реализована обработка исключений

Механизм обработки исключений в данном фрагменте кода реализован правильно. Перехвачены все возможные исключения при работе с механизмами LINQ

В данном коде содержится синтаксическая ошибка

Этот фрагмент кода не содержит синтаксических ошибок

**Ответ:** Механизм обработки исключений в данном фрагменте кода реализован неправильно. Проблема в том, что здесь используется механизм отложенного выполнения и реальное получение данных произойдет только в foreach, где не реализована обработка исключений   
Этот фрагмент кода не содержит синтаксических ошибок