|  |
| --- |
| EPAM Systems, RD Dep. |
| Контрольный тест  NET.C#.12 Введение в LINQ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial version | Анжелика Кравчук |  |  |  |
| 1.1 | Review and corrections. | Владимир Тихон |  |  |  |
| 1.2 | Improved answer sections. | Oleg |  |  |  |
| 1.3 | Added new questions | Oleg |  |  |  |

ФИО Студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата опроса\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Пояснения:**

1. Студент заполняет колонки:
2. «Верно?». Если верно, то укажите «Да», если нет – «нет».
3. «Объяснения и комментарии»:
   * ответы на вопросы «Почему не правильно? Почему правильно?»;
   * при необходимости пошаговое описание с указанием состояния переменных на каждом шаге.
   * нужно изложить объяснение, которое покажет понимание теории и умение применять знания на практике
   * в некоторых вопросах и вариантах ответах нужно исследовать практическую задачу, разобраться и написать объяснение.
   * в некоторых вариантах желательно, указать как исправить код, чтобы он успешно компилировался или выполнялся без ошибок.
4. Используя стиль «EDU\_Student».
5. Преподаватель заполняет колонки «Верный ответ», «Правильное объяснение», «Зачтено».
6. А также задает вопросы и даёт свои комментарии в секции «Объяснения и комментарии».
7. Используя стиль «EDU\_Reviewer».
8. При выполнении можно и нужно пользоваться:
   1. Лекцией;
   2. MSDN;
   3. Учебниками вроде Шилда и т.п.
9. **Что будут выведено на экран в результате выполнения следующего блока кода**

. . .

var values = new List<int>();

values.Add(1);

values.Add(5);

values.Add(15);

values.Add(11);

var result = from i in values

where i > 10

select i;

values[1] = 20;

foreach(var value in result)

{

Console.WriteLine(value);

}

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | 11  15  20 | Нет. |  |  |  |
|  | 20  15  11 | Да. Операция select имеет отложное выполнение и выполняется при перечислении последовательности. Т.к. значение первого элемента поменяли до перечисления, то select применится к последовательности с элементами: 20, 5, 15, 11. |  |  |
|  | 1  20  15  11 | Нет. |  |  |
|  | 11  15 | Нет. |  |  |
|  | 15  11 | Нет. |  |  |

1. **Что будут выведено на экран в результате выполнения следующего блока кода**

. . .

List<int> list = new List<int>() { 1, 2, 3 };

var x = list.GroupBy(i => { Console.Write(i); return i; });

var y = list.ToLookup(i => { Console.Write(i); return i; });

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | 123 | Да. Т.к. операция groupby имеет отложное выполнение, а lookup – непосредственное. |  |  |  |
|  | 123123 | Нет. Результат будет верным, если дописать foreach (var value in x){} в 4 строку. |  |  |
|  | Ничего | Нет. |  |  |

1. **Что будут выведено на экран в результате выполнения следующего блока кода**

. . .

List<int> list = new List<int>() { 1, 2, 3, 4, 5 };

List<int> all = list.FindAll( i => { Console.Write(i); return i < 3; });

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | 12 | Нет. |  |  |  |
|  | 123 | Нет. |  |  |
|  | Ничего | Нет. |  |  |
|  | 12345 | Да. Метод класса List<T> имеет непосредственное выполнение. |  |  |

1. **Что будут выведено на экран в результате выполнения следующего блока кода**

. . .

var people = new List<Person>()

{

new Person() {Name = "Jon", Age = 18},

new Person() {Name = "Jane", Age = 21},

new Person() {Name = "Alice", Age = 25},

new Person() {Name = "Bob", Age = 31}

};

var results = from p in people

where p.Age > 20

group p by p.Age%10

into grouped

orderby grouped.Key

select grouped;

foreach (var g in results)

{

Console.WriteLine("Key {0} - Count {1}",g.Key, g.Count());

}

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | Key 2 - Count 2  Key 3 - Count 1 | Нет. |  |  |  |
|  | Key 2 - Count 2  Key 1 - Count 3 | Нет. |  |  |
|  | Key 1 - Count 2  Key 5 - Count 1 | Да. Выбираю людей, чей возраст больше 20, группирую по Age%10, т.е. в группы, с ключами 1 и 5. Сортирую по возрастанию key. |  |  |
|  | Key 1 - Count 2  Key 5 - Count 1  Key 8 - Count 1 | Нет. |  |  |
|  | Key 1 - Count 1  Key 2 - Count 2  Key 8 - Count 1 | Нет. |  |  |

1. **What does the following expression evaluate to?**

. . .

string[] colors = { "green", "brown", "blue", "red" };

colors.Max (c => c.Length)

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | 5 | Да. Получаю максимальное значение длины строки. |  |  |  |
|  | green | Нет. |  |  |
|  | brown | Нет. |  |  |
|  | Compile-time error | Нет. |  |  |
|  | Exception thrown | Нет. |  |  |  |

1. **What does this expression evaluate to?**

. . .

string[] colors = { "green", "brown", "blue", "red" };

colors.OrderBy (c => c.Length).Single()

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | 3 | Нет. |  |  |  |
|  | red | Нет. |  |  |
|  | Compile-time error | Нет. |  |  |
|  | Exception thrown | Да. Single возвращает единственный элемент последовательности и генерирует исключение, если число элементов отлично от 1. |  |  |

1. **Given the following statement:**

. . .

string[] colors = { "green", "brown", "blue", "red" };

var query =

from c in colors

where c.Length > 3

orderby c.Length

select c;

. . .

**what type is query?**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | int | Нет. |  |  |  |
|  | string | Нет. |  |  |
|  | IEnumerable<int> | Нет. |  |  |
|  | IEnumerable<string> | Да. Производится фильтрация строк в linq to objects, следовательно, тип IEnumerable<string>. |  |  |
|  | IQueryable<int> | Нет. |  |  |  |
|  | IQueryable<string> | Нет. |  |  |  |

1. **What's the output from the following?**

. . .

string[] colors = { "green", "brown", "blue", "red" };

var query =

from c in colors

where c.Length == colors.Max (c => c.Length)

select c;

foreach (var element in query)

Console.WriteLine (element);

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | green followed by brown | Да. Происходит фильтрация строк. Выбираются строки, длина которых равна 5. |  |  |  |
|  | 5 followed by 5 | Нет. |  |  |
|  | Compile-time error | Нет. |  |  |
|  | Exception is thrown | Нет. |  |  |

1. **Assuming we make it syntactically correct, how many times does the Max subquery execute in the preceding example, when query is enumerated?**

. . .

string[] colors = { "green", "brown", "blue", "red" };

var query =

from c in colors

where c.Length == colors.Max (c => c.Length)

select c;

foreach (var element in query)

Console.WriteLine (element);

. . .

* **В объяснениях приведите оптимизированный вариант кода.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | once | Нет. |  |  |  |
|  | twice | Нет. |  |  |
|  | 3 times | Нет. |  |  |
|  | 4 times | Да. Оптимизированный вариант:  string[] colors = { "green", "brown", "blue", "red" };  int max = colors.Max (c => c.Length);  var query =  from c in colors  where c.Length ==  select c;  foreach (var element in query)  Console.WriteLine (element); |  |  |

1. **What does the following example output?**

. . .

string[] colors = { "green", "brown", "blue", "red" };

var list = new List<string> (colors);

IEnumerable<string> query = list.Where (c => c.Length == 3);

list.Remove ("red");

Console.WriteLine (query.Count());

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | 0 | Да. Операция where имеет отложное выполнение. |  |  |  |
|  | 1 | Нет. |  |  |
|  | 2 | Нет. |  |  |
|  | Exception thrown | Нет. |  |  |

1. **What's the output from the following?**

. . .

string[] colors = { "green", "brown", "blue", "red" };

var query = colors.Where (c => c.Contains ("e"));

query = query.Where (c => c.Contains ("n"));

Console.WriteLine (query.Count());

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | 1 | Да. Применяется фильтрация уже отфильтрованной последовательности. |  |  |  |
|  | 2 | Нет. |  |  |
|  | 3 | Нет. |  |  |
|  | 4 | Нет. |  |  |

1. **What's the output from the following?**

. . .

string[] colors = { "green", "brown", "blue", "red" };

string s = "e";

var query = colors.Where (c => c.Contains (s));

s = "n";

query = query.Where (c => c.Contains (s));

Console.WriteLine (query.Count());

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | 1 | Нет. |  |  |  |
|  | 2 | Да. Операция where имеет отложенное выполнение. Строке s сначала присвоили значение e. Но перед перечислением поменяли значение на n. Есть две строки, которые содержат символ n. |  |  |
|  | 3 | Нет. |  |  |
|  | 4 | Нет. |  |  |

1. **How does the compiler resolve the let clause, in the following query?**

. . .

string[] colors = { "green", "brown", "blue", "red" };

from c in colors

let middle = c.Substring (1, c.Length - 2)

where middle.Contains ("e")

select middle

. . .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | By translating it into a call to Enumerable.Let | Нет. |  |  |  |
|  | By expanding middle into c.Substring (1, c.Length - 2) in the where and select clauses | Да. Создается подстрока и из последовательности выбирается тот элемент, у которого заданная подстрока содержит символ e. |  |  |
|  | By projecting it into a temporary anonymous type | Нет. Проекция в анонимный тип => new {} |  |  |

1. **The compiler translates queries containing multiple generators (i.e., multiple from clauses) into:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Верно? | Вариант ответа | Объяснение и комментарии | Верный ответ | Правильное объяснение | Зачтено |
|  | Multiple Selects | Нет. |  |  |  |
|  | SelectMany | Да. В начала запроса вводится исходная переменная диапазона и входная последовательность. В любом другом мусте запроса он переводится как SelectMany. |  |  |
|  | Join | Нет. Устанавливает корреляцию между элементами двух последовательностей на основе сопоставления ключей. |  |  |