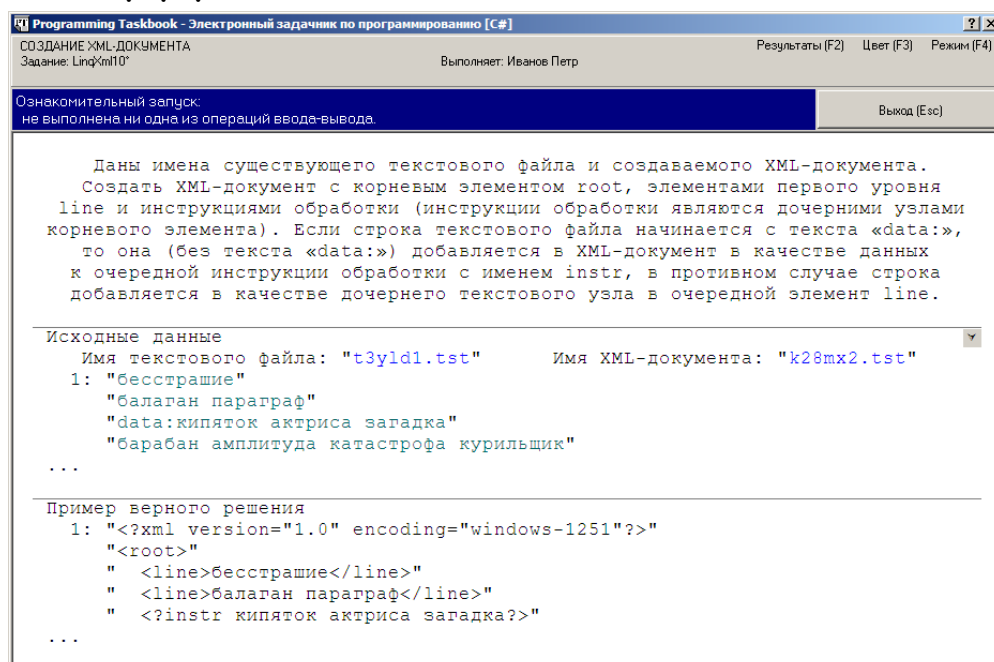


Создание XML-документа: LinqXml10

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Xml.Linq;
using PT4;

namespace PT4Tasks
{
    public class MyTask : PT
    {
        // При решении задач группы LinqXml доступны следующие
        // дополнительные методы расширения, определенные в задачнике:
        //
        // Show() и Show(cmt) - отладочная печать последовательности,
        //   cmt - строковый комментарий;
        //
        // Show(e => r) и Show(cmt, e => r) - отладочная печать
        //   значений r, полученных из элементов e последовательности,
        //   cmt - строковый комментарий.

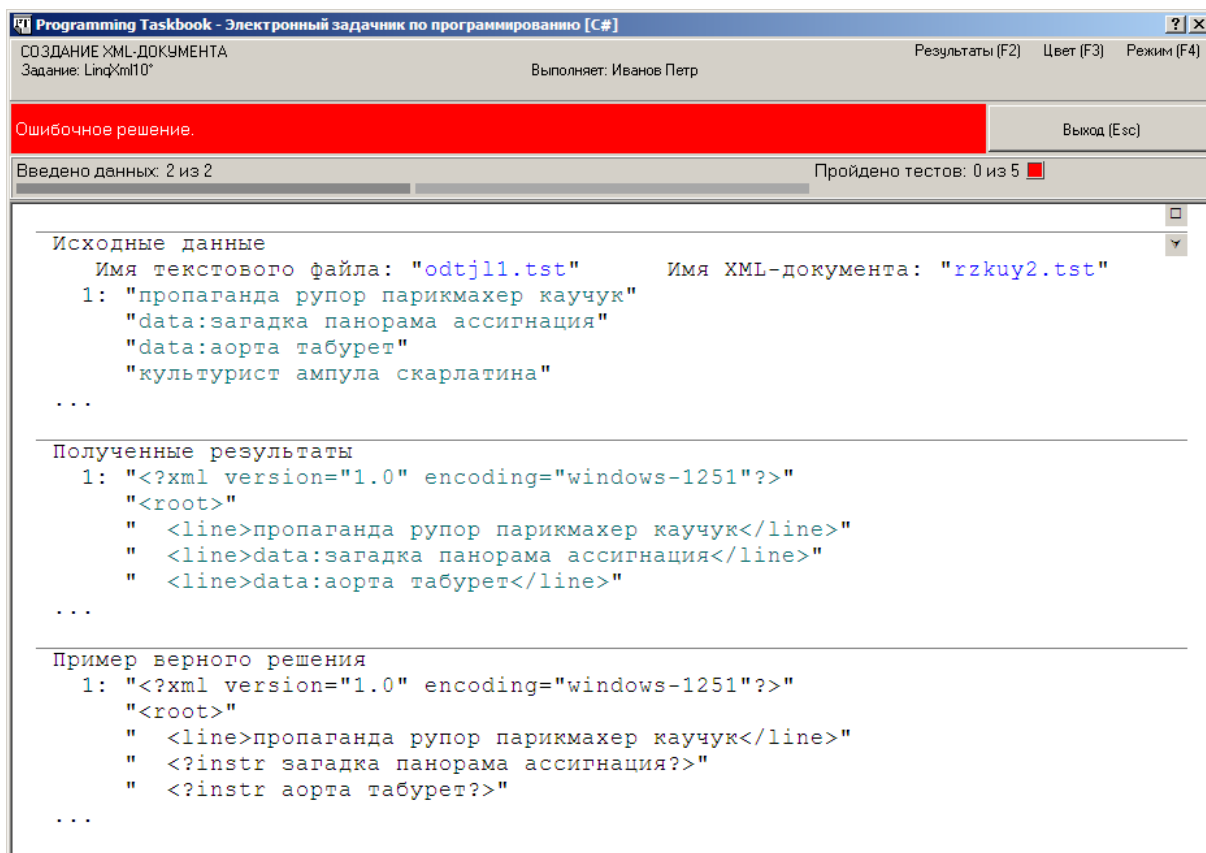
        public static void Solve()
        {
            Task("LinqXml10");
        }
    }
}
```



```

public static void Solve()
{
    Task("LinqXml10");
    var a = File.ReadAllLines(GetString(), Encoding.Default);
    XDocument d = new XDocument(
        new XDeclaration(null, "windows-1251", null),
        new XElement("root",
            a.Select(e => new XElement("line", e))));
    d.Save(GetString());
}

```



```

e => e.StartsWith("data:") ?
    new XProcessingInstruction("instr", e.Substring(5)) :
    new XElement("line", e)

```

При попытке откомпилировать полученную программу будет выведено сообщение об ошибке компиляции.

```

e => e.StartsWith("data:") ?
    new XProcessingInstruction("instr", e.Substring(5)) :
    new XElement("line", e) as XElement

```

Анализ содержимого XML-документа: LinqXml20

Programming Taskbook - Электронный задачник по программированию [C#] ? X

АНАЛИЗ СОДЕРЖИМОГО XML-ДОКУМЕНТА Результаты (F2) Цвет (F3) Режим (F4)
 Задание: LinqXml20* Выполняет: Иванов Петр

Ознакомительный запуск: не выполнена ни одна из операций ввода-вывода. Выход (Esc)

Дан XML-документ, содержащий хотя бы один элемент первого уровня. Для каждого элемента первого уровня найти его элементы-потомки, имеющие максимальное количество атрибутов. Перебирая элементы первого уровня в порядке их появления в XML-документе, вывести имя элемента, число N – максимальное количество атрибутов у его потомков (значение N может быть равно 0) – и имена потомков, имеющих N атрибутов (имена потомков выводить в алфавитном порядке; среди этих имен могут быть совпадающие). Если элемент первого уровня не содержит элементов-потомков, то в качестве значения N выводить -1, а в качестве имени потомка – текст «no child».

Исходные данные

Имя XML-документа: "cobdg.tst"

```
1: "<?xml version="1.0" encoding="windows-1251"?>"
   "<root d1="-15">"
   "  <bb>запевала<?sentence7 -85?></bb>"
   "  <bb>"
   ...
```

Пример верного решения

Элемент первого уровня	N	Элементы-потомки
"bb"	-1	"no child"
"bb"	1	"a"
		"cc"
"d"	2	"cc"
		"d"
"cc"	0	"dd"
"a"	2	"d"
		"dd"
"cc"	2	"b"

Programming Taskbook - Электронный задачник по программированию [C#] ? X

АНАЛИЗ СОДЕРЖИМОГО XML-ДОКУМЕНТА Результаты (F2) Цвет (F3) Режим (F4)
 Задание: LinqXml20* Выполняет: Иванов Петр

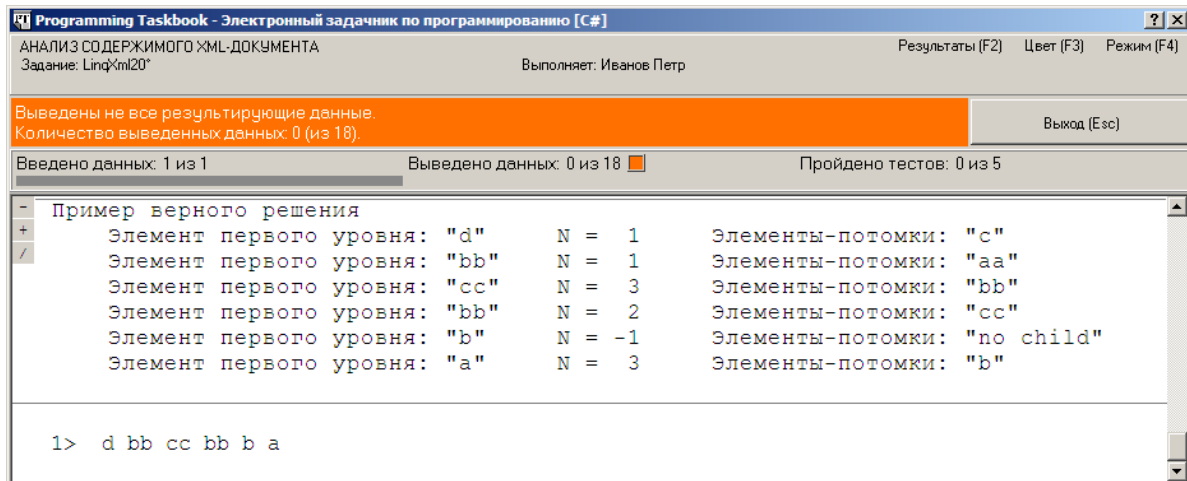
Ознакомительный запуск: не выполнена ни одна из операций ввода-вывода. Выход (Esc)

Исходные данные

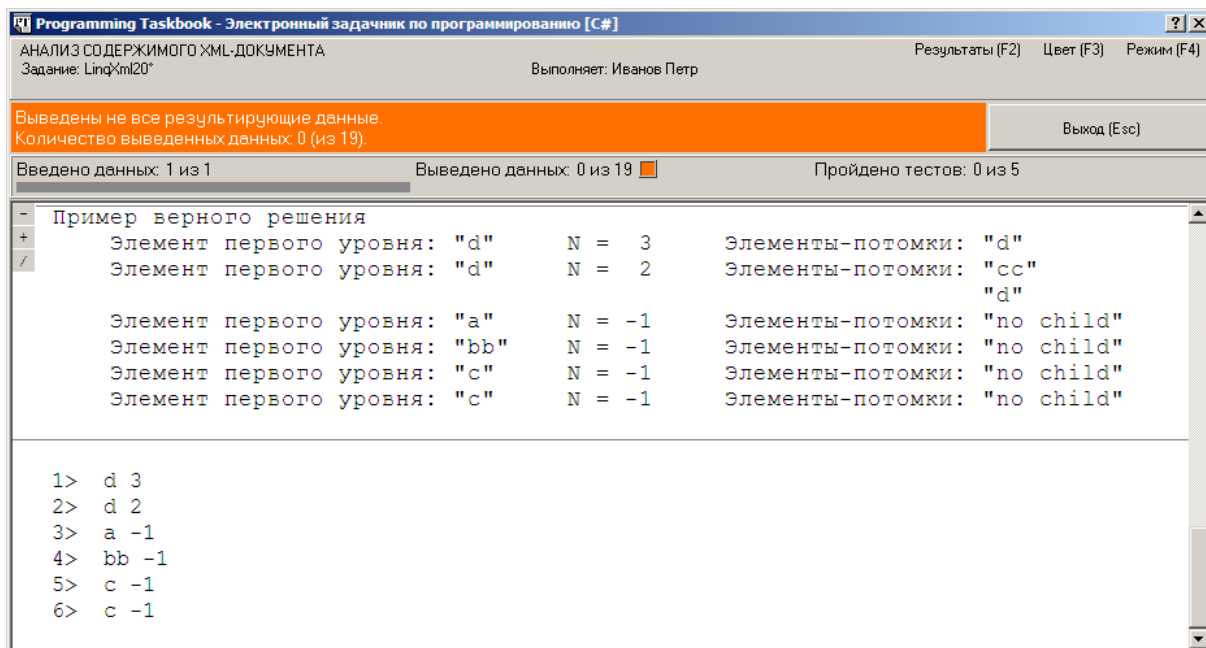
Имя XML-документа: "cobdg.tst"

```
1: "<?xml version="1.0" encoding="windows-1251"?>"
2: "<root d1="-15">"
3: "  <bb>запевала<?sentence7 -85?></bb>"
4: "  <bb>"
5: "    <cc>"
6: "      <a a1="-67" />"
7: "      <cc c1="-74">капитализм</cc>"
8: "    </cc>"
9: "    <!--retry-->"
10: "  </bb>"
11: "  <d c1="29">"
12: "    <bb>"
13: "      <d d1="72" d2="75">амплитуда<!--polygonal--></d>"
14: "    </bb>"
15: "    <cc c1="-62" d2="-82">настройщик</cc>"
16: "  </d>"
17: "  <cc b2="-9">"
18: "    <dd>каторжник</dd>"
19: "    <?degree2 -40?>"
20: "    <!--inflection-->"
21: "    <!--member-->"
22: "  </cc>"
23: "  <a d1="-45" d2="1">"
24: "    <dd d1="7" b2="-64">арматура</dd>"
25: "    <aa a1="44">капитализм</aa>"
26: "    <d b2="82" b3="79">"
27: "      <d>алгебра</d>"
```

```
XDocument d = XDocument.Load(GetString());
foreach (var e1 in d.Root.Elements())
{
  Show(e1.Name.LocalName);
}
```



```
Show(e1.Name.LocalName);
int max = e1.Descendants()
    .Select(e => e.Attributes().Count())
    .DefaultIfEmpty(-1).Max();
ShowLine(max);
```



```
var a = e1.Descendants()
    .Where(e => e.Attributes().Count() == max)
    .Select(e => e.Name.LocalName)
    .OrderBy(e => e)
    .DefaultIfEmpty("no child");
```

```
public static void Solve()
{
    Task("LinqXml20");
    XDocument d = XDocument.Load(GetString());
    foreach (var e1 in d.Root.Elements())
    {
        Put(e1.Name.LocalName);
        int max = e1.Descendants()
            .Select(e => e.Attributes().Count())
            .DefaultIfEmpty(-1).Max();
        Put(max);
        var a = e1.Descendants()
            .Where(e => e.Attributes().Count() == max)
            .Select(e => e.Name.LocalName)
            .OrderBy(e => e)
            .DefaultIfEmpty("no child");
        foreach (var e in a)
            Put(e);
    }
}
```

```
public static void Solve()
{
    Task("LinqXml20");
    XDocument d = XDocument.Load(GetString());
    foreach (var e1 in d.Root.Elements())
    {
        Put(e1.Name.LocalName);
        var max = (from e in e1.Descendants()
            select e.Attributes().Count())
            .DefaultIfEmpty(-1).Max();
        Put(max);
        var a = (from e in e1.Descendants()
            where e.Attributes().Count() == max
            let name = e.Name.LocalName
            orderby name
            select name)
            .DefaultIfEmpty("no child");
        foreach (var e in a)
            Put(e);
    }
}
```

Преобразование XML-документа: LinqXml28, LinqXml32, LinqXml37

LinqXml28. Дан XML-документ. Удалить дочерние текстовые узлы для всех элементов третьего уровня. Если текстовый узел является единственным дочерним узлом элемента, то после его удаления элемент должен быть представлен в виде комбинированного тега.

Указание. Использовать метод `OfType<XText>`.

```
public static void Solve()
{
    Task("LinqXml28");
    string name = GetString();
    XmlDocument d = XmlDocument.Load(name);
    d.Root.Elements().Elements().Elements()
        .Nodes().OfType<XText>().Remove();
    d.Save(name);
}
```

```
public static void Solve()
{
    Task("LinqXml28");
    string name = GetString();
    XmlDocument d = XmlDocument.Load(name);
    (from e in d.DescendantNodes()
     where e is XText && e.Ancestors().Count() == 4
     select e).Remove();
    d.Save(name);
}
```

LinqXml32. Дан XML-документ и строки S_1 и S_2 . В строке S_1 записано имя одного из элементов исходного документа, строка S_2 содержит допустимое имя элемента XML. Перед каждым элементом второго уровня с именем S_1 добавить элемент с именем S_2 . Добавленный элемент должен содержать последний атрибут и первый дочерний элемент последующего элемента (если они есть). Если элемент S_1 не имеет дочерних элементов, то добавленный перед ним элемент S_2 должен быть представлен в виде комбинированного тега.

Указание. Использовать метод `FirstOrDefault`.

```
public static void Solve()
{
    Task("LinqXml32");
    string name = GetString(), s1 = GetString(), s2 = GetString();
    XDocument d = XDocument.Load(name);
    foreach (var e in d.Root.Elements().Elements(s1))
        e.AddBeforeSelf(new XElement(s2,
            e.LastAttribute, e.Elements().FirstOrDefault()));
    d.Save(name);
}
```

LinqXml37. Дан XML-документ. Для каждого элемента второго уровня, имеющего потомков, добавить к его текстовому содержимому текстовое содержимое всех элементов-потомков, после чего удалить все его узлы-потомки, кроме дочернего текстового узла.

Указание. Использовать свойство `Value` класса `XElement`.

```
public static void Solve()
{
    Task("LinqXml37");
    string name = GetString();
    XDocument d = XDocument.Load(name);
    var a = from e in d.Root.Elements().Elements()
            where !e.IsEmpty
            select e;
    foreach (var e in a)
        e.Value = e.Value;
    d.Save(name);
}
```

Programming Taskbook - Электронный задачник по программированию [C#]

ПРЕОБРАЗОВАНИЕ XML-ДОКУМЕНТА
Задание: LinqXml37*
Выполняет: Иванов Петр

Результаты (F2) Цвет (F3) Режим (F4)

Задание выполнено! Выход (Esc)

Введено данных: 1 из 1
Пройдено тестов: 5 из 5

Исходные данные

Имя XML-документа: "ozaxk.tst"

```

1: "<?xml version="1.0" encoding="windows-1251"?>"
2: "<root a1="89" c2="22">"
3: "  <a c2="28" d3="58">"
4: "    <b b3="66">"
5: "      <c c2="-81" b3="73">диод</c>"
6: "    <dd></dd>"
7: "  </b>"
8: "  <a a1="95">"
9: "    <aa b2="47" a3="90">каблук</aa>"
10: "    <dd b2="96" />"
11: "    <cc c1="11" c2="17"></cc>"
12: "    <c a2="70">пропаганда</c>"
13: "  </a>"
14: "</root>"
15: "<cc>табурет<?giving1 -25?></cc>"
16: "<d a2="44">полустанок<!--sleeve--></d>"
17: "<b>"
18: "  <d>"
19: "    <!--teleprocessing-->"
20: "  </d>"
21: "  <b>пропаганда</b>"
22: "  <bb>"
23: "    <c></c>"
24: "  </bb>"
25: "  <!--abstract-->"
26: "  <?omission7 26?>"
27: "  </b>"
28: "  <a a1="-98">kozyрек<!--recognition--></a>"
29: "    <dd b2="76" b3="93">"
30: "      <aa c2="7">парабола<?manufacturer10 -75?></aa>"
31: "    </dd>"

```

Programming Taskbook - Электронный задачник по программированию [C#]

ПРЕОБРАЗОВАНИЕ XML-ДОКУМЕНТА
Задание: LinqXml37*
Выполняет: Иванов Петр

Результаты (F2) Цвет (F3) Режим (F4)

Задание выполнено! Выход (Esc)

Введено данных: 1 из 1
Пройдено тестов: 5 из 5

Полученные результаты

```

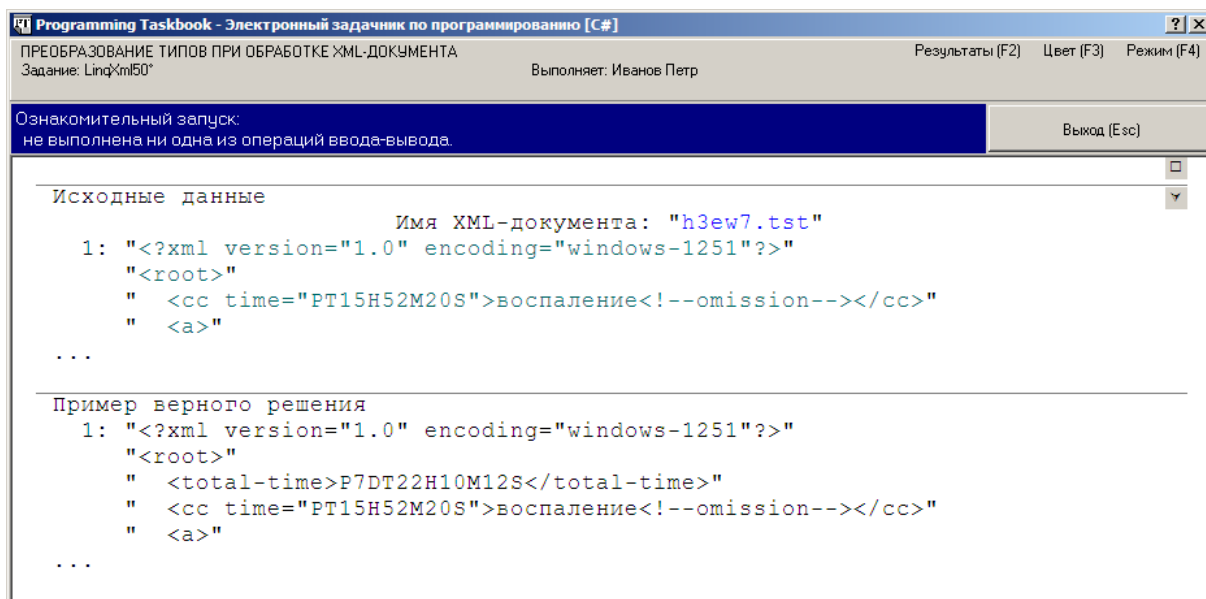
1: "<?xml version="1.0" encoding="windows-1251"?>"
2: "<root a1="89" c2="22">"
3: "  <a c2="28" d3="58">"
4: "    <b b3="66">диод</b>"
5: "  <a a1="95">каблукпропаганда</a>"
6: "  </a>"
7: "  <cc>табурет<?giving1 -25?></cc>"
8: "  <d a2="44">полустанок<!--sleeve--></d>"
9: "  <b>"
10: "    <d></d>"
11: "    <b>пропаганда</b>"
12: "    <bb></bb>"
13: "    <!--abstract-->"
14: "    <?omission7 26?>"
15: "    </b>"
16: "    <a a1="-98">kozyрек<!--recognition--></a>"
17: "      <dd b2="76" b3="93">"
18: "        <aa c2="7">парабола</aa>"
19: "      </dd>"
20: "    </a>"
21: "    <d c1="-1" a2="-9">"
22: "      <dd a1="93">акустикакрановщик</dd>"
23: "    <!--shell-->"
24: "    </d>"
25: "    <dd a2="99">католик</dd>"
26: "    <c b2="99">кипяток<!--enclose--></c>"
27: "</root>"

```


Преобразование типов при обработке XML-документа: LinqXml50

LinqXml50. Дан XML-документ. С каждым элементом документа связывается некоторый промежуток времени (в днях, часах, минутах и секундах). Этот промежуток либо явно указывается в атрибуте `time` данного элемента (в формате, принятом в стандарте XML), либо, если данный атрибут отсутствует, считается равным одному дню. Добавить в начало набора дочерних узлов корневого элемента элемент `total-time` со значением, равным суммарному значению промежутков времени, связанных со всеми элементами первого уровня.

Указание. Использовать приведение объекта `XAttribute` к `Nullable`-типу `TimeSpan?` и операцию `??`.



The screenshot shows a window titled "Programming Taskbook - Электронный задачник по программированию [C#]". The task is "ПРЕОБРАЗОВАНИЕ ТИПОВ ПРИ ОБРАБОТКЕ XML-ДОКУМЕНТА" (Type Conversion in XML Document Processing) with the task name "LinqXml50". The user is "Иванов Петр". The window displays the following XML input and output:

```
Имя XML-документа: "h3ew7.tst"
1: "<?xml version="1.0" encoding="windows-1251"?>"
   "<root>"
   " <cc time="PT15H52M20S">воспаление<!--omission--></cc>"
   " <a>"
   ...

Пример верного решения
1: "<?xml version="1.0" encoding="windows-1251"?>"
   "<root>"
   " <total-time>P7DT22H10M12S</total-time>"
   " <cc time="PT15H52M20S">воспаление<!--omission--></cc>"
   " <a>"
   ...
```

```
public static void Solve()
{
    Task("LinqXml50");
    string name = GetString();
    XDocument d = XDocument.Load(name);
    TimeSpan t = d.Root.Elements()
        .Select(e => (TimeSpan?)e.Attribute("time") ??
            new TimeSpan(24, 0, 0))
        .Aggregate(TimeSpan.Zero, (a, e) => a + e);
    Show(t);
}
```

Programming Taskbook - Электронный задачник по программированию [C#]

ПРЕОБРАЗОВАНИЕ ТИПОВ ПРИ ОБРАБОТКЕ XML-ДОКУМЕНТА
Задание: LinqXml150* Выполняет: Иванов Петр Результаты (F2) Цвет (F3) Режим (F4)

Ошибочное решение. Выход (Esc)

Введено данных: 1 из 1 Пройдено тестов: 0 из 5

Исходные данные

Имя XML-документа: "ngf5p.tst"

```
1: "<?xml version="1.0" encoding="windows-1251"?>"
   "<root>"
   " <dd time="PT5H25M1S">"
   "   <cc>"
   ...
```

Полученные результаты

```
1: "<?xml version="1.0" encoding="windows-1251"?>"
   "<root>"
   " <dd time="PT5H25M1S">"
   "   <cc>"
   "     <dd time="PT1H10M57S" />"
   ...
```

Пример верного решения

```
1: "<?xml version="1.0" encoding="windows-1251"?>"
   "<root>"
   " <total-time>P5DT5H25M1S</total-time>"
   " <dd time="PT5H25M1S">"
   "   <cc>"
   ...
```

1> 5.05:25:01

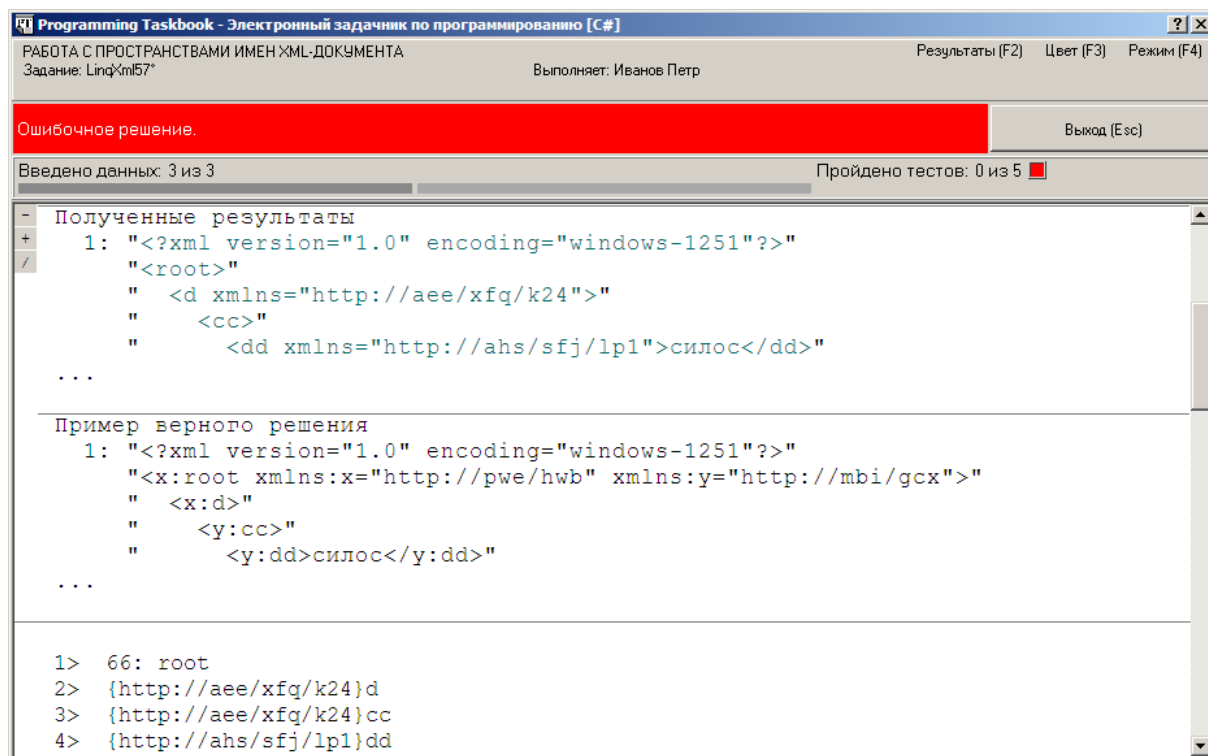
```
d.Root.AddFirst(new XElement("total-time", t));
d.Save(name);
```

```
public static void Solve()
{
    Task("LinqXml150");
    string name = GetString();
    XDocument d = XDocument.Load(name);
    d.Root.AddFirst(new XElement("total-time",
        d.Root.Elements()
            .Select(e => (TimeSpan?)e.Attribute("time") ??
                new TimeSpan(24, 0, 0))
            .Aggregate(TimeSpan.Zero, (a, e) => a + e)));
    d.Save(name);
}
```

Работа с пространствами имен XML-документа: LinqXml57

LinqXml57. Дан XML-документ и строки S_1 и S_2 , содержащие различные пространства имен. Удалить в документе определения исходных пространств имен и определить в корневом элементе два префикса пространств имен: префикс x , связанный с S_1 , и префикс y , связанный с S_2 . Снабдить префиксом x элементы нулевого и первого уровня, а префиксом y — элементы последующих уровней.

```
public static void Solve()
{
    Task("LinqXml57");
    string name = GetString(), s1 = GetString(), s2 = GetString();
    XDocument d = XDocument.Load(name);
    d.Descendants().Show(e => e.Name + "\n");
}
```



```
d.Root.Add(new XAttribute(XNamespace.Xmlns + "x", s1),
    new XAttribute(XNamespace.Xmlns + "y", s2));
```

```
foreach (var e in d.Descendants())
{
    string s = e.Ancestors().Count() <= 1 ? s1 : s2;
    e.Name = (XNamespace)s + e.Name.LocalName;
}
```

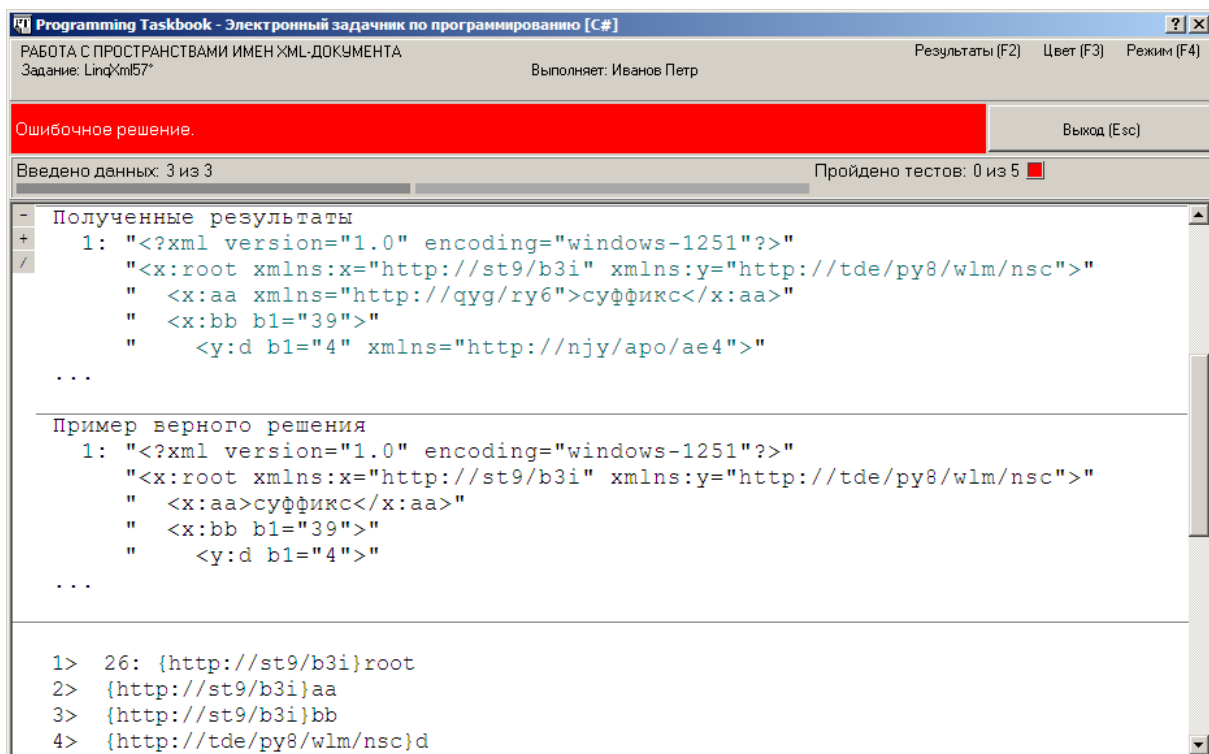
Вариант:

```

XNamespace ns1 = (XNamespace)s1, ns2 = (XNamespace)s2;
foreach (var e in d.Descendants())
{
    XNamespace ns = e.Ancestors().Count() <= 1 ? ns1 : ns2;
    e.Name = ns + e.Name.LocalName;
}

d.Save(name);

```

**Неверное исправление:**

```
e.Attribute("xmlns").Remove();
```

Верное исправление:

```
e.Attributes("xmlns").Remove();
```

Вариант верного исправления:

```
d.Descendants().Attributes("xmlns").Remove();
```

```
public static void Solve()
{
    Task("LinqXml57");
    string name = GetString(), s1 = GetString(), s2 = GetString();
    XDocument d = XDocument.Load(name);
    d.Root.Add(new XAttribute(XNamespace.Xmlns + "x", s1),
        new XAttribute(XNamespace.Xmlns + "y", s2));
    XNamespace ns1 = (XNamespace)s1, ns2 = (XNamespace)s2;
    foreach (var e in d.Descendants())
    {
        XNamespace ns = e.Ancestors().Count() <= 1 ? ns1 : ns2;
        e.Name = ns + e.Name.LocalName;
    }
    d.Descendants().Attributes("xmlns").Remove();
    d.Save(name);
}
```

Дополнительные задания на обработку XML-документов: *LinqXml61*, *LinqXml82*

LinqXml61. Дан XML-документ с информацией о клиентах фитнес-центра. Образец элемента первого уровня:

```
<record>
  <id>10</id>
  <date>2000-05-01T00:00:00</date>
  <time>PT5H13M</time>
</record>
```

Здесь *id* — код клиента (целое число), *date* — дата с информацией о годе и месяце, *time* — продолжительность занятий (в часах и минутах) данного клиента в течение указанного месяца. Преобразовать документ, изменив элементы первого уровня следующим образом:

```
<time id="10" year="2000" month="5">PT5H13M</time>
```

Порядок следования элементов первого уровня не изменять.

```

Programming Taskbook - Электронный задачник по программированию [C#]
ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ НА ОБРАБОТКУ XML-ДОКУМЕНТОВ
Задание: LinqXml61
Выполняет: Иванов Петр
Результаты (F2) Цвет (F3) Режим (F4)

Ознакомительный запуск:
не выполнена ни одна из операций ввода-вывода.
Вывод (Esc)

- 28: " <record>"
+ 29: " <id>8</id>"
/ 30: " <date>2005-03-01T00:00:00</date>"
31: " <time>P1DT1H32M</time>"
32: " </record>"
33: " <record>"
34: " <id>128</id>"
35: " <date>2006-09-01T00:00:00</date>"
36: " <time>PT17H53M</time>"
37: " </record>"
38: " </fitness>"

Пример верного решения
1: "<?xml version="1.0" encoding="windows-1251"?>"
2: "<fitness xmlns="http://h1b/yun/of0">"
3: " <time id="35" year="2006" month="9">PT19H32M</time>"
4: " <time id="195" year="2006" month="9">PT23H</time>"
5: " <time id="161" year="2005" month="1">PT5H7M</time>"
6: " <time id="195" year="2005" month="1">PT5H57M</time>"
7: " <time id="31" year="2005" month="4">P1DT6H16M</time>"
8: " <time id="8" year="2005" month="3">P1DT1H32M</time>"
9: " <time id="128" year="2006" month="9">PT17H53M</time>"
10: "</fitness>"

```

```
d.Root.Elements()
```

```
.Select(e => new XElement(ns + "time",
    new XAttribute("id", e.Element(ns + "id").Value),
    new XAttribute("year",
        ((DateTime)e.Element(ns + "date")).Year),
    new XAttribute("month",
        ((DateTime)e.Element(ns + "date")).Month),
    e.Element(ns + "time").Value));
```

```
public static void Solve()
```

```
{
    Task("LinqXml61");
    string name = GetString();
    XDocument d = XDocument.Load(name);
    XNamespace ns = d.Root.Name.Namespace;
    d.Root.ReplaceNodes(d.Root.Elements()
        .Select(e => new XElement(ns + "time",
            new XAttribute("id", e.Element(ns + "id").Value),
            new XAttribute("year",
                ((DateTime)e.Element(ns + "date")).Year),
            new XAttribute("month",
                ((DateTime)e.Element(ns + "date")).Month),
            e.Element(ns + "time").Value));
    d.Save(name);
}
```

```
d.Root.ReplaceNodes(
    from e in d.Root.Elements()
    select new XElement(ns + "time",
```

LinqXml82. Дан XML-документ с информацией о задолженности по оплате коммунальных услуг. Образец элемента первого уровня (смысл данных тот же, что и в LinqXml76, в качестве имени элемента первого уровня указываются номера дома и квартиры, разделенные символом «-» (дефис) и снабженные префиксом `addr`, а в качестве значения этого элемента указывается размер задолженности для данной квартиры):

```
<addr12-23>1245.64</addr12-23>
```

Преобразовать документ, выполнив группировку данных по номеру дома, а в пределах каждого дома — по номеру этажа. Изменить элементы первого уровня следующим образом:

```
<house12>
  <floor1 count="0" total-debt="0" />
  ...
  <floor6 count="1" total-debt="1245.64" />
  ...
  <floor9 count="3" total-debt="3142.7" />
</house12>
```

Имя элемента первого уровня должно иметь префикс `house`, после которого указывается номер дома, имя элемента второго уровня должно иметь префикс `floor`, после которого указывается номер этажа. Атрибут `count` равен числу задолжников на данном этаже, атрибут `total-debt` определяет суммарную задолженность по данному этажу, округленную до двух дробных знаков (незначащие нули не отображаются). Если на данном этаже отсутствуют задолжники, то для соответствующего элемента второго уровня значения атрибутов `count` и `total-debt` должны быть равны 0. Элементы первого уровня должны быть отсортированы по возрастанию номеров домов, а их дочерние элементы — по возрастанию номеров этажей.

```
public static void Solve()
{
    Task("LinqXml82");
    string name = GetString();
    XDocument d = XDocument.Load(name);
    XNamespace ns = d.Root.Name.Namespace;
    var a = d.Root.Elements()
        .Select(e =>
        {
            string[] s = e.Name.LocalName.Substring(4).Split('-');
            return new
            {
                house = int.Parse(s[0]),
                floor = (int.Parse(s[1]) - 1) % 36 / 4 + 1,
                debt = (double)e
            };
        })
    .Show();
}
```


}

Programming Taskbook - Электронный задачник по программированию [C#]
 ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ НА ОБРАБОТКУ XML-ДОКУМЕНТОВ
 Задание: LinqXml82? Выполняет: Иванов Петр
 Результаты (F2) Цвет (F3) Режим (F4)

Ошибочное решение. Вывод (Esc)

Введено данных: 1 из 1 Пройдено тестов: 0 из 5

```

- Исходные данные
+
/
      Имя XML-документа: "i6kw3.tst"
1: "<?xml version="1.0" encoding="windows-1251"?>"
2: "<debts xmlns="http://rk4/x8g/ta5">"
3: " <addr162-100>1984.46</addr162-100>"
4: " <addr17-77>1752.8</addr17-77>"
5: " <addr111-26>1901.17</addr111-26>"
6: " <addr17-100>469.56</addr17-100>"
7: " <addr23-123>1383.6</addr23-123>"
8: " <addr162-84>1064.7</addr162-84>"
9: " <addr111-100>441.55</addr111-100>"
10: " <addr23-127>676.56</addr23-127>"
11: " <addr162-86>658.81</addr162-86>"
12: " <addr111-1>164.79</addr111-1>"
13: " <addr17-6>926.53</addr17-6>"
14: "</debts>"
  
```

```

1> 11: { house = 162, floor = 7, debt = 1984,46 }
2> { house = 17, floor = 2, debt = 1752,8 }
3> { house = 111, floor = 7, debt = 1901,17 }
4> { house = 17, floor = 7, debt = 469,56 }
5> { house = 23, floor = 4, debt = 1383,6 }
6> { house = 162, floor = 3, debt = 1064,7 }
7> { house = 111, floor = 7, debt = 441,55 }
8> { house = 23, floor = 5, debt = 676,56 }
9> { house = 162, floor = 4, debt = 658,81 }
10> { house = 111, floor = 1, debt = 164,79 }
11> { house = 17, floor = 2, debt = 926,53 }
  
```

```

a.GroupBy(e => e.house,
  (k, ee) => new XElement(ns + ("house" + k)))

a.OrderBy(e => e.house)
  .GroupBy(e => e.house,
  (k, ee) => new XElement(ns + ("house" + k)))

d.Root.ReplaceNodes(a.OrderBy(e => e.house)
  .GroupBy(e => e.house,
  (k, ee) => new XElement(ns + ("house" + k))));

d.Save(name);
  
```

Programming Taskbook - Электронный задачник по программированию [C#]
 ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ НА ОБРАБОТКУ XML-ДОКУМЕНТОВ
 Задание: LinqXml82* Выполняет: Иванов Петр Результаты (F2) Цвет (F3) Режим (F4)

Ошибочное решение. Выход (Esc)

Введено данных: 1 из 1 Пройдено тестов: 0 из 5

```

- Полученные результаты
+ 1: "<?xml version='1.0' encoding='windows-1251'?>"
/ 2: "<debts xmlns='http://zt9/kwr/pmb/r12'"
3: " <house64 />"
4: " <house69 />"
5: " <house75 />"
6: " <house103 />"
7: " <house152 />"
8: " <house191 />"
9: "</debts>"
  
```

```

1> 14: { house = 75, floor = 7, debt = 587,15 }
2> { house = 64, floor = 6, debt = 658,07 }
3> { house = 69, floor = 8, debt = 1473,31 }
4> { house = 69, floor = 7, debt = 792,69 }
5> { house = 191, floor = 2, debt = 1245,62 }
6> { house = 152, floor = 3, debt = 1116,44 }
7> { house = 191, floor = 7, debt = 1185,96 }
8> { house = 191, floor = 4, debt = 533,32 }
9> { house = 64, floor = 5, debt = 1278,61 }
10> { house = 75, floor = 8, debt = 423,23 }
11> { house = 152, floor = 2, debt = 101,92 }
12> { house = 191, floor = 8, debt = 1417,46 }
13> { house = 191, floor = 3, debt = 571,74 }
14> { house = 103, floor = 5, debt = 1675,02 }
  
```

```
var floors = Enumerable.Range(1, 9);
```

```
d.Root.ReplaceNodes(a.OrderBy(e => e.house)
    .GroupBy(e => e.house,
        (k, ee) => new XElement(ns + ("house" + k),
            floors.GroupJoin(ee, e1 => e1, e2 => e2.floor,
                (e1, ee2) => new XElement(ns + ("floor" + e1),
                    new XAttribute("count", ee2.Count()),
                    new XAttribute("total-debt", ee2.Sum(e => e.debt)))))));
```

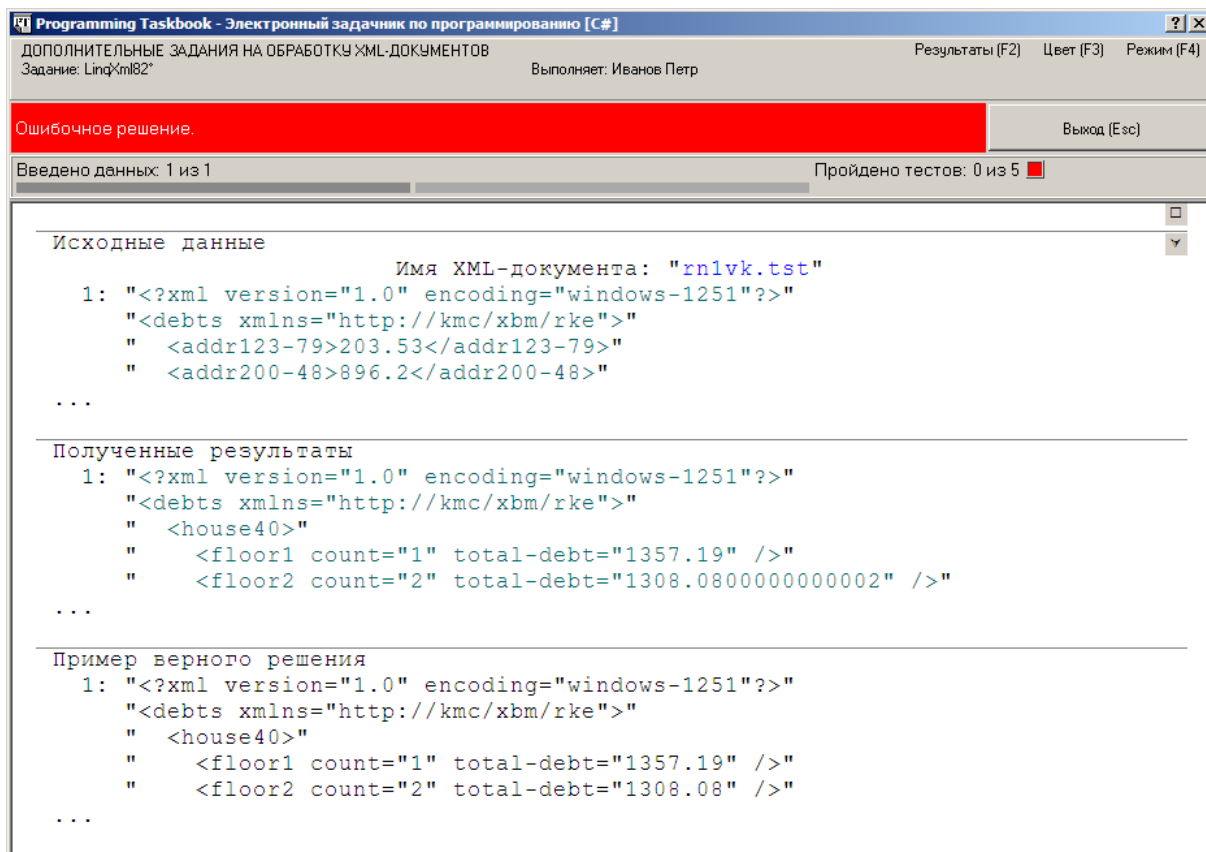
```
public static void Solve()
{
    Task("LinqXml82");
    string name = GetString();
    XDocument d = XDocument.Load(name);
    XNamespace ns = d.Root.Name.Namespace;
    var a = d.Root.Elements()
        .Select(e =>
        {
            string[] s = e.Name.LocalName.Substring(4).Split('-');
            return new
            {

```

```

        house = int.Parse(s[0]),
        floor = (int.Parse(s[1]) - 1) % 36 / 4 + 1,
        debt = (double)e
    };
});
var floors = Enumerable.Range(1, 9);
d.Root.ReplaceNodes(a.OrderBy(e => e.house)
    .GroupBy(e => e.house,
        (k, ee) => new XElement(ns + ("house" + k),
            floors.GroupJoin(ee, e1 => e1, e2 => e2.floor,
                (e1, ee2) => new XElement(ns + ("floor" + e1),
                    new XAttribute("count", ee2.Count()),
                    new XAttribute("total-debt", ee2.Sum(e => e.debt)))))));
d.Save(name);
}

```



```
Math.Round(ee2.Sum(e => e.debt), 2)
```

```

d.Root.ReplaceNodes(
    from e in a
    orderby e.house
    group e by e.house
    into ee
    select new XElement(ns + ("house" + ee.Key),
        from e1 in floors

```

```
join e2 in ee
on e1 equals e2.floor
into ee2
select new XElement(ns + ("floor" + e1),
  new XAttribute("count", ee2.Count()),
  new XAttribute("total-debt",
    Math.Round(ee2.Sum(e => e.debt), 2))));
```