PIX Robotics
делает умнее

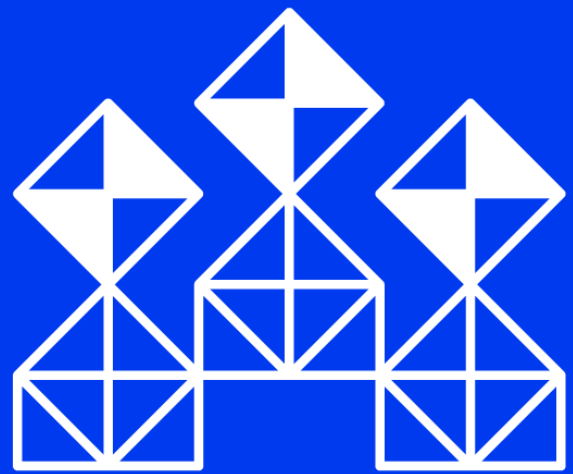
Как приручить XDocument? XmlDocument vs XDocument

DFD — Developers for Developers

Кирилл Пронин
Разработчик PIX RPA
kirill.pronin@pix.ru

Цель – ответить на эти вопросы:

- Как я столкнулся с XML историями?
- Что такое XML и как он выглядит?
- Применение XDocument & XMLDocument?
- XDocument – в чем его прикол? Почему он стал модным?
- В каких случаях стоит забыть о XMLDocument?
- А есть тесты производительности?



PIX Robotics
делает умнее

Немного о себе

Кирилл Пронин

Разработчик программного обеспечения, PIX Robotics

kirill.pronin@pix.ru

в области прикладной математики (аэродинамики)

6+ лет опыта в разработке десктопных приложений

8+ лет опыта в педагогике технических дисциплин

PIX Robotics
делает умнее

PIX Studio — среда разработки

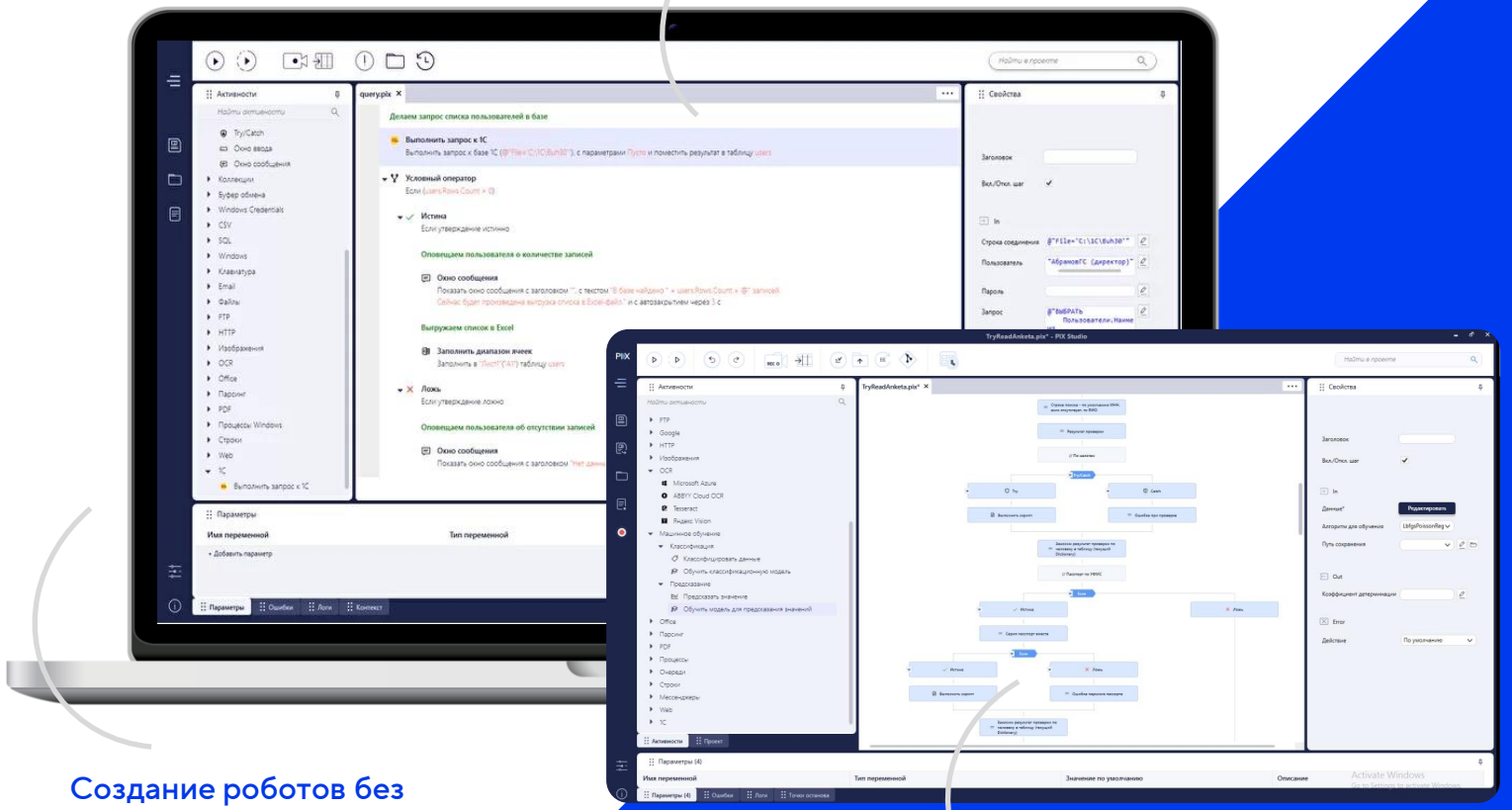
Среда разработки обеспечивает:

Создание логики процесса для робота, возможные отклонения и исключения.

Два интерфейса отображения алгоритма робота: в виде схемы и в виде списка команд.

Интерактивную среду обучения моделей ИИ AutoML Smart Activities.

Построчное отображение алгоритма вместо графических схем
— проще и удобнее для разработчика



Создание роботов без использования кода

250+
встроенных элементов конструктора

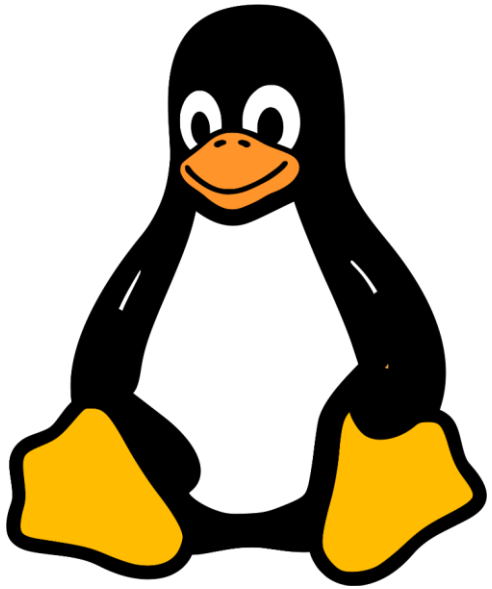
Создание робота с помощью моделирования схем алгоритмов — удобнее для архитектора бизнес-процессов

PIX Robotics
делает умнее

Время интересных историй

Excel

Но был 1 момент...



Но вот еще момент:

alt linux

ASTRA LINUX®

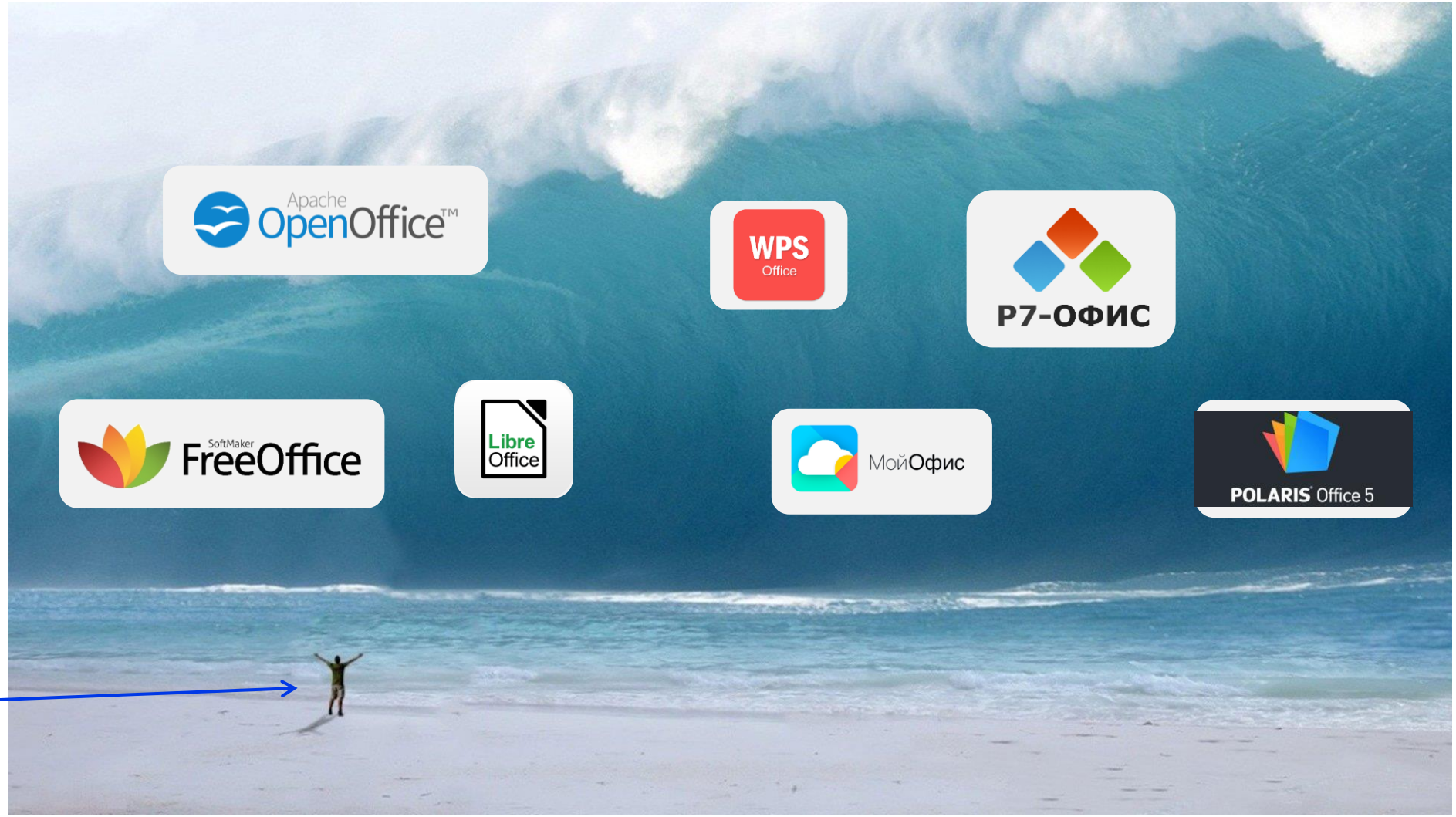
РЕДОС

ROSA

Linux-подобных ОС много, и каждый реализует свой аналог табличного редактора...

PIX Robotics
делает умнее

Другими словами

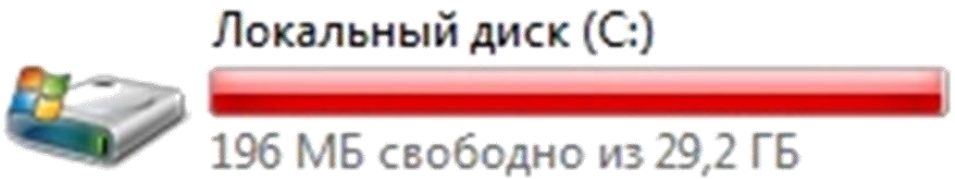


PIX Robotics
делает умнее

Пути решения

1 ПУТЬ API. Лучше сделать 6 штук, а еще потом 8.

Каждая API не легковесная, а использование достигает лишь 5% ...

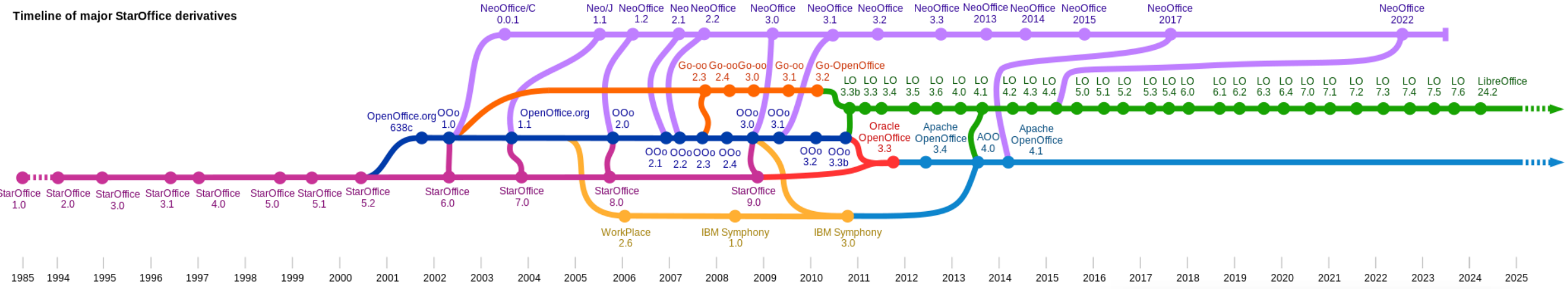


PIX Robotics
делает умнее

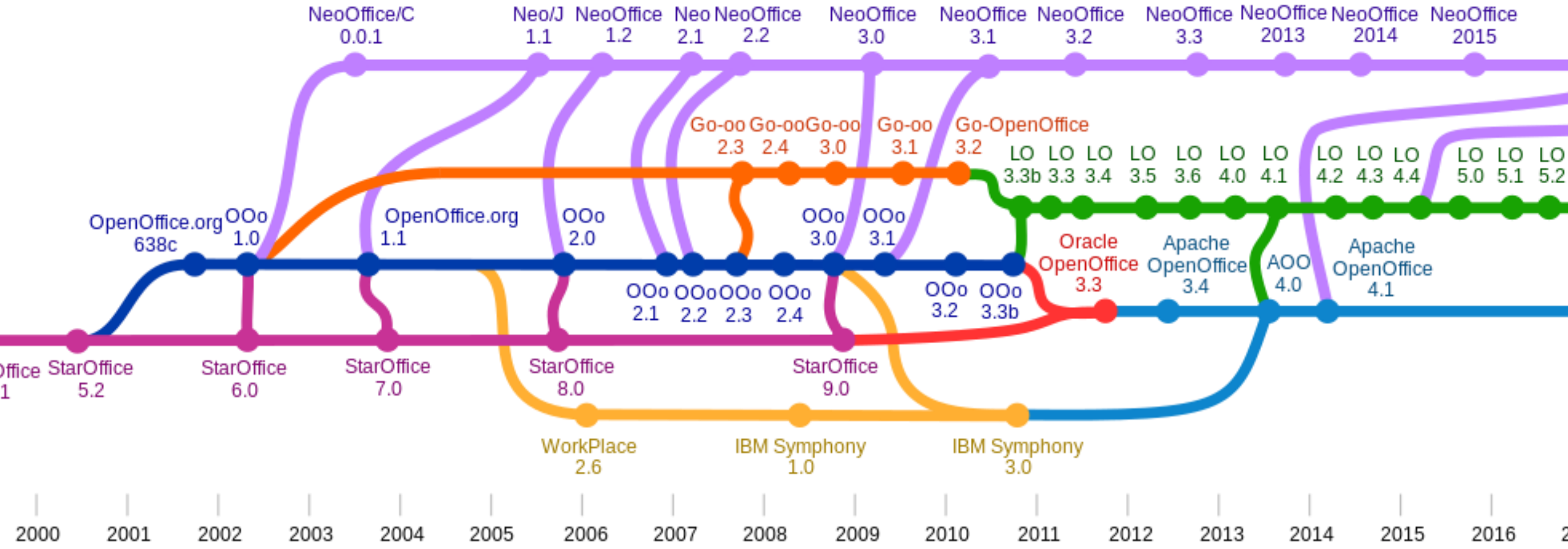
Пути решения

2

ODF (Open Document Format) + общий предок.



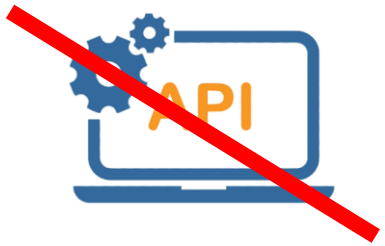
PIX Robotics
делает умнее



PIX Robotics
делает умнее

Итог

Разрабатываем для ... Apache OpenOffice™

Но для него нет ... 

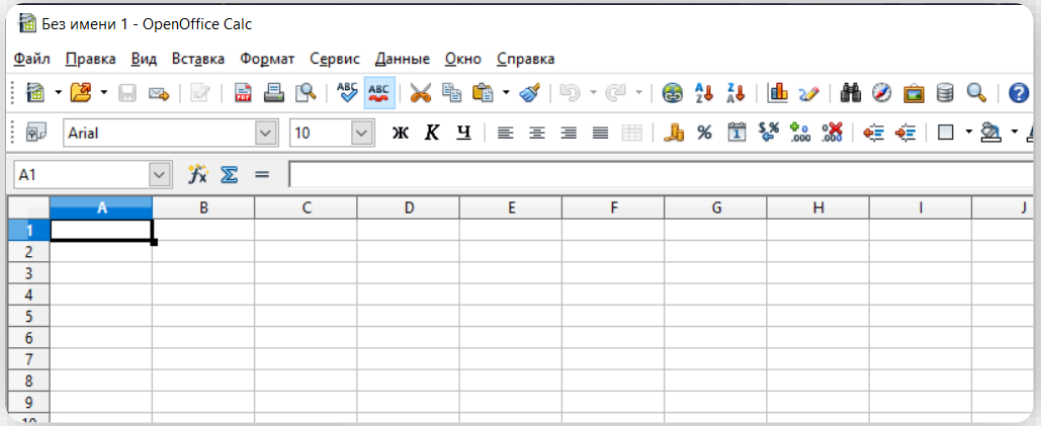
Значит делаем всё ... 

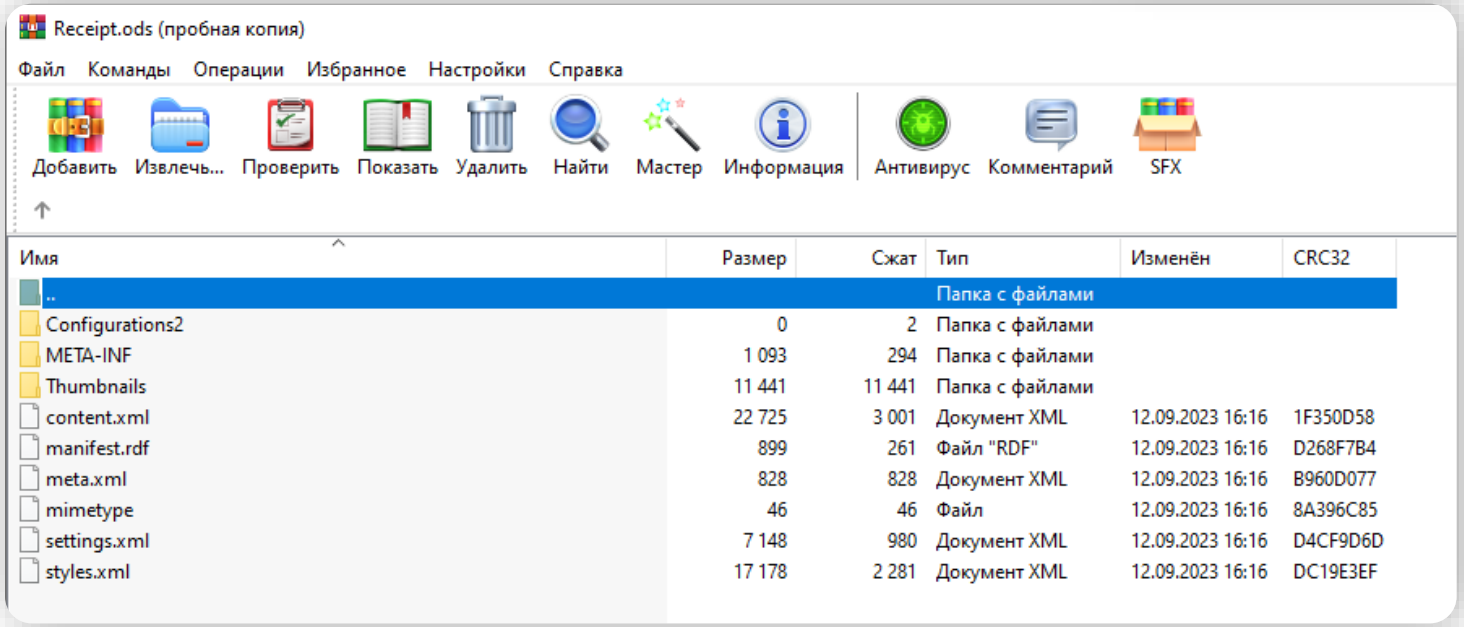
PIX Robotics
делает умнее

Время теории

*.ods файл — это архив, внутри которого располагаются все настройки и данные.

Сами данные в виде xml файла хранятся в content.xml





Значит, открываем архив, считываем только content и в результате получаем XML-код.

Ура! Мы поняли, как открыть и получить данные в код.

PIX Robotics
делает умнее

Все офисное ПО базируется на XML. Что такое XML и как он выглядит?

```xml
<book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>
        An in-depth look at creating applications
        with XML.
    </description>
</book>
```

XML – это гибкий текстовый формат, который позволяет представлять иерархические данные. Он состоит из элементов, атрибутов и текстового содержимого.

PIX Robotics
делает умнее

Все офисное ПО базируется на XML. Что такое XML и как он выглядит?

```xml
<office:spreadsheet>
    <table:table table:name="Лист1" table:style-name="ta1" table:print="false">
        <table:table-column table:style-name="co1" table:default-cell-style-name="ce2"/>
        <table:table-column table:style-name="co2" table:default-cell-style-name="ce4"/>
        <table:table-column table:style-name="co3" table:default-cell-style-name="ce6"/>
        <table:table-row table:style-name="ro1">
            <table:table-cell table:style-name="Default" table:number-columns-repeated="3"/>
        </table:table-row>
        <table:table-row table:style-name="ro2">
            <table:table-cell table:style-name="ce1" office:value-type="string">
                <text:p>№</text:p>
            </table:table-cell>
            <table:table-cell table:style-name="ce3" office:value-type="string">
                <text:p>Работы/материалы</text:p>
            </table:table-cell>
            <table:table-cell table:style-name="ce3" office:value-type="string">
                <text:p>Стоимость</text:p>
            </table:table-cell>
        </table:table-row>
        ...
```

```
<Лист>
 <Строчка>
  <Ячейка>
   <Тип> Значение </Тип>
  </Ячейка>
  <Ячейка>
   <Тип> Значение </Тип>
  </Ячейка>
 </Строчка>
</Лист>
```

PIX Robotics
делает умнее

Ну а как обработать XML формат?

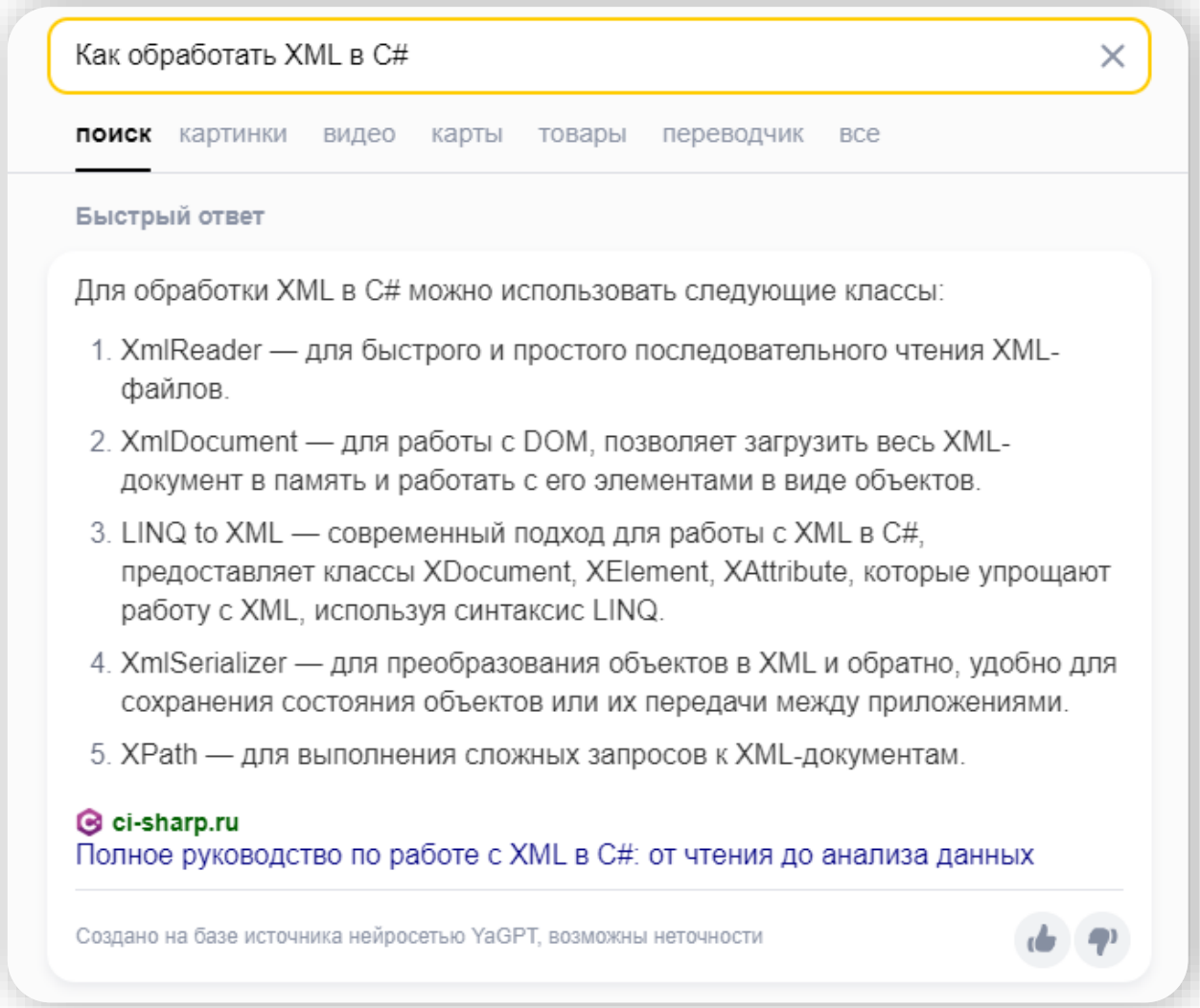
Самый популярный — это XMLDocument. На нем написано множество статей, примеров, Nuget'ов, но хорош ли он?

А вот самый малоизвестный — XDocument.

Поэтому первое впечатление...

XMLDocument

XDocument



PIX Robotics
делает умнее

Время написать первый код для будущего обработчика табличек

```
using ZipArchive zArchive = new(stream);
ZipArchiveEntry? entry = zArchive.GetEntry("content.xml");

// Проверка на правильное считывание файла
if (entry is null)
{
    throw new InvalidOperationException();
}

var streamXML = entry.Open();
XmlDocument doc = new XmlDocument();
doc.Load(streamXML);

XmlNamespaceManager nmsManager = new XmlNamespaceManager(doc.NameTable);

foreach (KeyValuePair<string,string> eachNamespace in _namespaces)
    nmsManager.AddNamespace(eachNamespace.Key, eachNamespace.Value);
```

А что в Namespaces?

PIX Robotics
делает умнее

```
private static Dictionary<string,string> _namespaces = new Dictionary<string, string>
{
        {"table", "urn:oasis:names:tc:opendocument:xmlns:table:1.0"},
        {"office", "urn:oasis:names:tc:opendocument:xmlns:office:1.0"},
        {"style", "urn:oasis:names:tc:opendocument:xmlns:style:1.0"},
        {"text", "urn:oasis:names:tc:opendocument:xmlns:text:1.0"},
        {"draw", "urn:oasis:names:tc:opendocument:xmlns:drawing:1.0"},
        {"fo", "urn:oasis:names:tc:opendocument:xmlns:xsl-fo-compatible:1.0"},
        {"dc", "http://purl.org/dc/elements/1.1/"},
        {"meta", "urn:oasis:names:tc:opendocument:xmlns:meta:1.0"},
        {"number", "urn:oasis:names:tc:opendocument:xmlns:datastyle:1.0"},
        {"presentation", "urn:oasis:names:tc:opendocument:xmlns:presentation:1.0"},
        {"svg", "urn:oasis:names:tc:opendocument:xmlns:svg-compatible:1.0"},
        {"chart", "urn:oasis:names:tc:opendocument:xmlns:chart:1.0"},
        {"dr3d", "urn:oasis:names:tc:opendocument:xmlns:dr3d:1.0"},
        {"math", "http://www.w3.org/1998/Math/MathML"},
        {"form", "urn:oasis:names:tc:opendocument:xmlns:form:1.0"},
        {"script", "urn:oasis:names:tc:opendocument:xmlns:script:1.0"},
        {"ooo", "http://openoffice.org/2004/office"},
        {"ooow", "http://openoffice.org/2004/writer"},
        {"oooc", "http://openoffice.org/2004/calc"},
        {"dom", "http://www.w3.org/2001/xml-events"},
        {"xforms", "http://www.w3.org/2002/xforms"},
        {"xsd", "http://www.w3.org/2001/XMLSchema"},
        {"xsi", "http://www.w3.org/2001/XMLSchema-instance"},
        {"rpt", "http://openoffice.org/2005/report"},
        {"of", "urn:oasis:names:tc:opendocument:xmlns:of:1.2"},
        {"rdfa", "http://docs.oasis-open.org/opendocument/meta/rdfa#"},
        {"config", "urn:oasis:names:tc:opendocument:xmlns:config:1.0"}
};
```

PIX Robotics
делает умнее

Но допустим, где значения ячеек?

```
private DataTable GetSheet(XmlNode tableNode, XmlNamespaceManager nmsManager)
{
    DataTable sheet = new DataTable(tableNode.Attributes["table:name"].Value);

    XmlNodeList rowNodes = tableNode.SelectNodes("table:table-row", nmsManager);

    int rowIndex = 0;
    foreach (XmlNode rowNode in rowNodes)
        this.GetRow(rowNode, sheet, nmsManager, ref rowIndex);

    return sheet;
}
```

не забываем про
NamespaceManager

PIX Robotics
делает умнее

Но допустим, где значения ячеек?

```
private void GetRow(XmlNode rowNode, DataTable sheet, XmlNamespaceManager nmsManager, ref int rowIndex)
{
    XmlAttribute rowsRepeated = rowNode.Attributes["table:number-rows-repeated"];
    if (rowsRepeated == null || int.Parse(rowsRepeated.Value, CultureInfo.InvariantCulture) == 1)
    {
        while (sheet.Rows.Count < rowIndex)
            sheet.Rows.Add(sheet.NewRow());

        DataRow row = sheet.NewRow();

        XmlNodeList cellNodes = rowNode.SelectNodes("table:table-cell", nmsManager);
        int cellIndex = 0;
        foreach (XmlNode cellNode in cellNodes)
        {
            var value = GetCell(cellNode, row, ref cellIndex);
        ...
```

не забываем про
NamespaceManager

PIX Robotics
делает умнее

Но допустим, где значения ячеек?

```
private string GetCell(XmlNode cellNode, DataRow row, ref int cellIndex)
{
    XmlAttribute cellRepeated = cellNode.Attributes["table:number-columns-repeated"];
    string cellValue = this.ReadCellValue(cellNode);
    return cellValue;
}
```

```
private string ReadCellValue(XmlNode cell)
{
    XmlAttribute cellVal = cell.Attributes["office:value"];

    if (cellVal == null)
        return cell.InnerText;
    else
        return cellVal.Value;
}
```

Получили значение

PIX Robotics
делает умнее

С сохранением и вставкой все куда проще:
принцип полной перезаписи информации в файл.

Но запомним правило:
«МЫ С NAMESPACE MANAGER ХОДИМ ПАРОЙ»

PIX Robotics
делает умнее

А потом я вижу XDocument

```
string filePath = "путь_к_файлу.ods";

// Загрузка файла .ods в XDocument
XDocument document = XDocument.Load(filePath);

// Находим таблицу в файле
XElement table = document.Descendants().FirstOrDefault(e => e.Name.LocalName == "table");

// Создаем новую ячейку
XElement newCell = new XElement(XName.Get("table-cell", "urn:oasis:names:tc:opendocument:xmlns:table:1.0"));

// Устанавливаем значение ячейки
newCell.Add(new XElement(XName.Get("p", "urn:oasis:names:tc:opendocument:xmlns:text:1.0"), "Значение ячейки"));

// Добавляем новую ячейку в таблицу
table.Add(newCell);

// Сохраняем изменения обратно в файл .ods
document.Save(filePath);
Console.WriteLine("Ячейка успешно добавлена!");
```

PIX Robotics
делает умнее

Немножко сравним?

```
public void AppendChildNode_XmlDocument(XmlDocument doc)
{
    // Создание нового элемента
    XmlElement newElement = doc.CreateElement("book");

    // Добавление атрибута к элементу
    newElement.SetAttribute("category", "fiction");

    // Создание и добавление дочерних элементов
    XmlElement titleElement = doc.CreateElement("title");
    titleElement.InnerText = "The Great Gatsby";
    newElement.AppendChild(titleElement);

    // Добавление нового элемента в корневой элемент документа
    doc.DocumentElement.AppendChild(newElement);
}
```

PIX Robotics
делает умнее

Немножко сравним?

```
public void AppendChildNode_XDocument(XDocument doc)
{
    // Создание нового элемента
    XElement newElement = new XElement("book",
        new XAttribute("category", "fiction"),
        new XElement("title", "The Great Gatsby"),
        new XElement("author", "F. Scott Fitzgerald")
    );

    // Добавление нового элемента в корневой элемент документа
    doc.Root.Add(newElement);
}
```

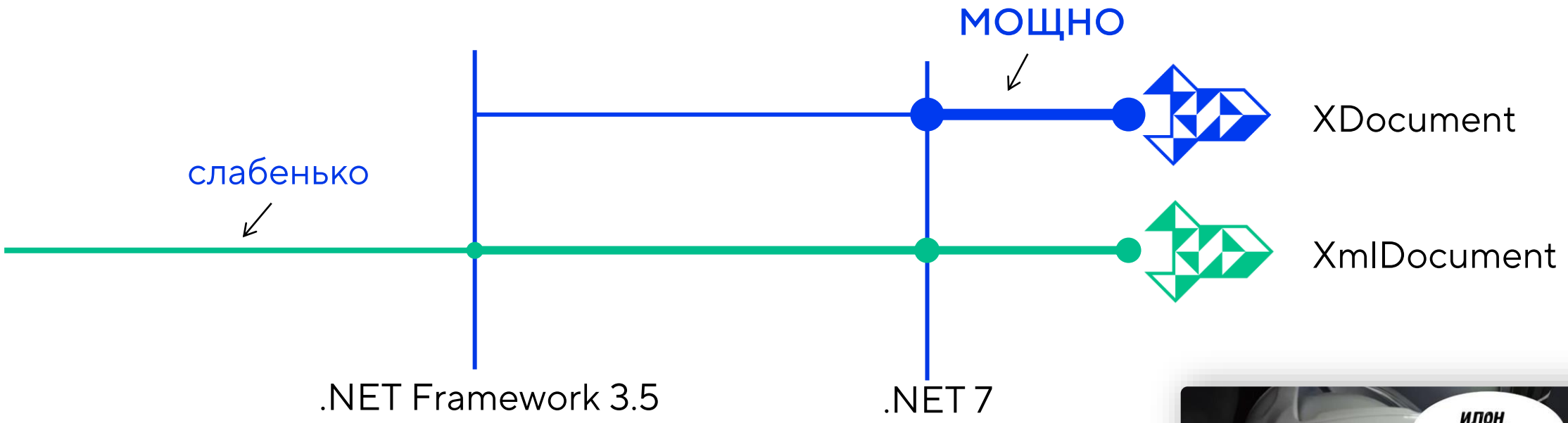
PIX Robotics
делает умнее

Семь раз отмерь — и один раз XDocument

XDocument — это класс, который представляет XML-документ в виде объекта в памяти.

В отличие от XmlDocument, XDocument предоставляет более современный и удобный API для работы с XML.

А когда он появился?



PIX Robotics
делает умнее

Плюсы и минусы

Сравнение	XMLDocument	XDocument
Версия .Net (Framework)	1.0 +	3.5 +
Сторонняя поддержка	-	Unity, Xbox 360+, Windows Phone
Namespace	System.Xml	System.Xml.Linq
Применение LINQ	-	+
Namespaces	Namespace Manager добавь и будет счастье	На уровне элемента (без подключения доп. контента)
XPath	В принципе можно, сам создашь?	System.Xml.XPath

PIX Robotics
делает умнее

Плюсы и минусы

Некоторые из достоинств XDocument по сравнению с XmlDocument:

1. Более простой и интуитивно понятный синтаксис
2. Удобное добавление и удаление элементов
3. Поддержка пространств имен
4. Использование нескольких файлов и фрагментов
5. Поддержка LINQ-запросов

XMLDocument

XDocument

PIX Robotics
делает умнее

Теперь к смысловой части в прямом значении

XmlDocument & XDocument помнят абсолютно всю структуру XML-документа Но...

XmlNode (XmlDocument) работает, так как по другому не умеет.

XNode (XDocument) дает возможность разработчику выбрать нужный вариант записи.

```
XDocument doc = XDocument.Load(ms);

doc.Save("SomeFile.xml", SaveOptions.DisableFormatting);
```

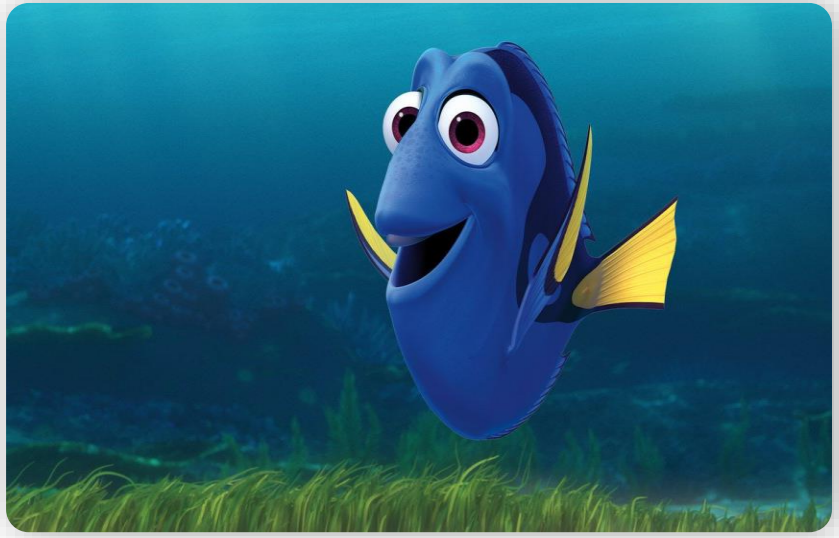
DisableFormatting	1	Сохранение всех незначительных пробелов при сериализации.
None	0	Форматирование (отступ) XML при сериализации.
OmitDuplicateNamespaces	2	Удаление дубликатов объявлений пространств имен при сериализации.

PIX Robotics
делает умнее

Немного про память

XmlDocument загружает всю структуру XML-документа в память, что может потребовать большое количество памяти, особенно при работе с большими файлами XML

XDocument использует более эффективный подход к загрузке и обработке XML-документов. Он использует ленивую загрузку, что означает, что он загружает только ту часть XML-документа, которая необходима для выполнения операций, и не загружает весь документ в память сразу.



PIX Robotics
делает умнее

Время написать второй код для будущего обработчика табличек

```
XDocument doc = XDocument.Load(streamXML);

var reader = doc.CreateReader();
XmlNamespaceManager nmsManager = new XmlNamespaceManager(reader.NameTable);

foreach (KeyValuePair<string, string> eachNamespace in _namespaces)
    nmsManager.AddNamespace(eachNamespace.Key, eachNamespace.Value);

var nodes = doc.XPathSelectElements("/office:document-content/office:body/office:spreadsheet/table:table",
nmsManager);
// Больше nmsManager не нужен

foreach (XElement node in nodes)
    GetSheet(node);
```

А что в Namespaces?

PIX Robotics
делает умнее

```
private static Dictionary<string,string> _namespaces = new Dictionary<string, string>
{
        {"table", "urn:oasis:names:tc:opendocument:xmlns:table:1.0"},
        {"office", "urn:oasis:names:tc:opendocument:xmlns:office:1.0"},
        {"style", "urn:oasis:names:tc:opendocument:xmlns:style:1.0"},
        {"text", "urn:oasis:names:tc:opendocument:xmlns:text:1.0"},
        {"draw", "urn:oasis:names:tc:opendocument:xmlns:drawing:1.0"},
        {"fo", "urn:oasis:names:tc:opendocument:xmlns:xsl-fo-compatible:1.0"},
        {"dc", "http://purl.org/dc/elements/1.1/"},
        {"meta", "urn:oasis:names:tc:opendocument:xmlns:meta:1.0"},
        {"number", "urn:oasis:names:tc:opendocument:xmlns:datastyle:1.0"},
        {"presentation", "urn:oasis:names:tc:opendocument:xmlns:presentation:1.0"},
        {"svg", "urn:oasis:names:tc:opendocument:xmlns:svg-compatible:1.0"},
        {"chart", "urn:oasis:names:tc:opendocument:xmlns:chart:1.0"},
        {"dr3d", "urn:oasis:names:tc:opendocument:xmlns:dr3d:1.0"},
        {"math", "http://www.w3.org/1998/Math/MathML"},
        {"form", "urn:oasis:names:tc:opendocument:xmlns:form:1.0"},
        {"script", "urn:oasis:names:tc:opendocument:xmlns:script:1.0"},
        {"ooo", "http://openoffice.org/2004/office"},
        {"ooow", "http://openoffice.org/2004/writer"},
        {"oooc", "http://openoffice.org/2004/calc"},
        {"dom", "http://www.w3.org/2001/xml-events"},
        {"xforms", "http://www.w3.org/2002/xforms"},
        {"xsd", "http://www.w3.org/2001/XMLSchema"},
        {"xsi", "http://www.w3.org/2001/XMLSchema-instance"},
        {"rpt", "http://openoffice.org/2005/report"},
        {"of", "urn:oasis:names:tc:opendocument:xmlns:of:1.2"},
        {"rdfa", "http://docs.oasis-open.org/opendocument/meta/rdfa#"},
        {"config", "urn:oasis:names:tc:opendocument:xmlns:config:1.0"}
};
```

PIX Robotics
делает умнее

Время написать второй код для будущего обработчика табличек

```
public static List<string> GetSheet(XElement tableNode)
{
    return tableNode.Elements()
                    .Where(x => x.Name.LocalName == "table-row")
                    .SelectMany(GetRow)
                    .ToList();
}
```

```
public static List<string> GetRow(XElement rowNode)
{
    return rowNode.Elements()
                  .Where(x => x.Name.LocalName == "table-cell")
                  .Select(GetCell)
                  .ToList();
}
```

PIX Robotics
делает умнее

Время написать второй код для будущего обработчика табличек

```
public static string GetCell(XElement cellNode)
{
    string? textValue = cellNode.Elements().FirstOrDefault(x => x.Name.LocalName == "p")?.Value;
    return textValue;
}
```

Получили значение

PIX Robotics
делает умнее

Немножко лайфхаков XDocument

Какие методы самые-самые для XDocument?

AddAfterSelf(Object)/AddBeforeSelf(Object) - Добавляет содержимое после/до данного узла.

Ancestors() - Возвращает коллекцию элементов-предков узла.

Descendants() - Возвращает коллекцию подчиненных узлов для данного документа или элемента.

Elements() - Возвращает коллекцию дочерних элементов.

IsAfter(XNode)/IsBefore(XNode) - Определяет, предшествует/следует ли текущий узел.

Для XElement:

Attributes() - Возвращает коллекцию атрибутов этого элемента.

Время бенчмарков. Тест №1: Load-Append-Remove на простом xml

Допустим, у меня есть библиотека книжек, и там указано около 10 книг по такой структуре.

Проведем тест на основе Load-Append-Remove для XMLDoc и XDoc.

```xml
<book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>
        An in-depth look at creating applications
        with XML.
    </description>
</book>
```

Method	Mean	Error	StdDev	Median	Gen0	Gen1	Allocated
LoadXml_XmlDocument_Load	34.25 us	0.617 us	1.014 us	34.25 us	8.1787	0.0610	33.47 KB
LoadXml_XDocument_Load	32.62 us	0.652 us	1.389 us	32.71 us	7.0190	-	28.74 KB
LoadXml_XmlDocument_FromFile	260.04 us	6.999 us	20.638 us	267.92 us	8.3008	0.2441	34.51 KB
LoadXml_XDocument_FromFile	259.72 us	7.547 us	22.251 us	266.30 us	6.8359	0.4883	29.75 KB
AppendChildNode_XmlDocument	33.73 us	0.664 us	0.764 us	33.92 us	8.3008	0.0610	33.93 KB
AppendChildNode_XDocument	34.10 us	0.680 us	1.059 us	34.00 us	7.0801	-	29.03 KB
RemoveChildNode_XmlDocument	36.91 us	0.685 us	1.430 us	36.75 us	8.7891	-	35.95 KB
RemoveChildNode_XDocument	32.37 us	0.644 us	1.193 us	32.29 us	7.0190	-	28.81 KB

PIX Robotics
делает умнее

Время бенчмарков. Тест №2: Взаимодействие с OpenOffice

Допустим, у меня есть заполненный табличный файл *.ods.

Проведем тест на основе Load-ReadData-Append-Remove для XMLDoc и XDoc.

Method	Mean	Error	StdDev	Gen0	Gen1	Allocated
LoadXml_XmlDocument_Load	223.6 us	2.07 us	1.84 us	35.1563	-	143.75 KB
LoadXml_XDocument_Load	230.2 us	2.55 us	2.13 us	25.6348	-	105.33 KB
ReadXmlCellData_XmlDocument	312.1 us	3.09 us	2.89 us	51.7578	13.6719	212.35 KB
ReadXmlCellData_XDocument	259.3 us	5.08 us	5.44 us	33.2031	1.4648	136.83 KB
AppendXmlCellData_XmlDocument	312.2 us	3.10 us	2.74 us	52.2461	0.9766	214.25 KB
AppendXmlCellData_XDocument	262.0 us	3.48 us	2.72 us	33.2031	0.4883	137 KB
RemoveXmlCellData_XmlDocument	316.9 us	2.49 us	2.08 us	51.7578	13.6719	212.35 KB
RemoveXmlCellData_XDocument	252.9 us	2.17 us	1.81 us	33.2031	1.4648	136.83 KB

PIX Robotics
делает умнее

Время бенчмарков. Тест №3 – Взаимодействие с КЛАДР

Сделал выгрузку из ГАР (КЛАДР) и взял дельту изменений по одной улице.

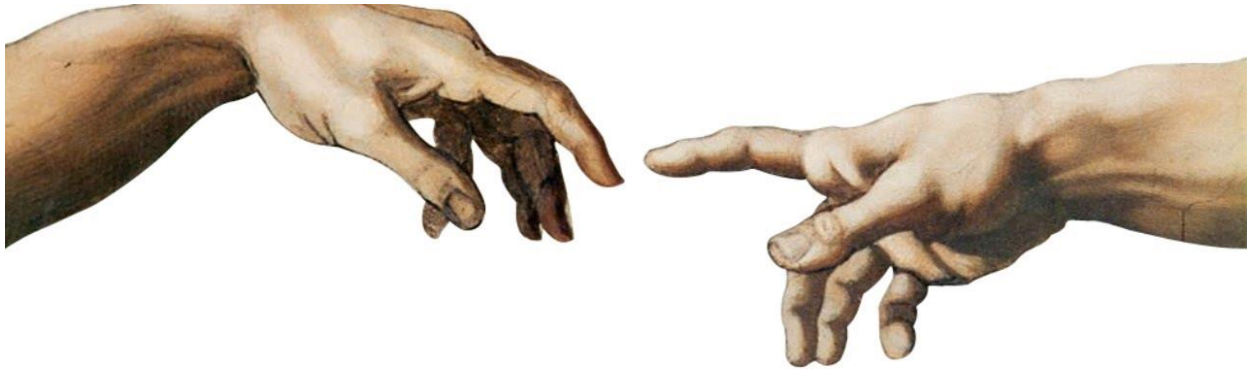
Проведем тест на основе Load-Append-Remove для XMLDoc и XDoc.

Method	Mean	Error	StdDev	Gen0	Gen1	Gen2	Allocated
LoadXml_XmlDocument_Load	3.176 s	0.0607 s	0.0674 s	127000.0000	66000.0000	5000.0000	731.37 MB
LoadXml_XDocument_Load	1.863 s	0.0179 s	0.0168 s	81000.0000	43000.0000	5000.0000	455.17 MB
LoadXml_XmlDocument_FromFile	3.064 s	0.0326 s	0.0272 s	127000.0000	66000.0000	5000.0000	731.37 MB
LoadXml_XDocument_FromFile	1.929 s	0.0149 s	0.0132 s	81000.0000	43000.0000	5000.0000	455.17 MB
AppendChildNode_XmlDocument	3.017 s	0.0515 s	0.0457 s	127000.0000	66000.0000	5000.0000	731.37 MB
AppendChildNode_XDocument	1.869 s	0.0358 s	0.0335 s	81000.0000	43000.0000	5000.0000	455.17 MB
RemoveChildNode_XmlDocument	3.006 s	0.0201 s	0.0168 s	127000.0000	66000.0000	5000.0000	731.37 MB
RemoveChildNode_XDocument	1.864 s	0.0250 s	0.0234 s	81000.0000	43000.0000	5000.0000	455.17 MB

PIX Robotics
делает умнее

Рекомендации по использованию

Всё в ваших ...



Главные опорные точки :

- Версия .NET
- Размер *.xml файлов для обработки
- Возможность изменять легаси

PIX Robotics
делает умнее

PIX Robotics
делает умнее

СПАСИБО ЗА ВНИМАНИЕ!

ВРЕМЯ ВОПРОСОВ

Кирилл Пронин

Разработчик PIX RPA

kirill.pronin@pix.ru