Стандарты XML и JSON

Лекция для слушателей ЛШЮП

А.Г.Марчук, д.ф.-м.н., профессор

История формализмов текстового представления данных

- Компьютерные языки (XML и др.)
- Данные в языках программирования (JSON и др.)
- CSV (Comma Separated Values) – формат представления табличных данных
- TeX – язык разметки, применяемый для подготовки печатных форм

Эволюция понятий и языков разметки

- Гипертекст – композиция, в которой есть тексты, ссылки, мультимедиа контент
- Язык разметки (Markup Language) – правила включения в текст формализованных конструкций, превращающий его в иерархическое построение
- GML и SGML (Standard Generalized Markup Language) – язык и метаязык разметки
- HTML (HyperText Markup Language), 1986-1991 – прикладной язык для Web-страниц

Что такое XML?

- XML (eXtendable Markup Language), 1998 – это способ представления структурированных данных в виде текста
- Лаконичность и простота
- Использование Unicode
- Универсальность и тотальная применимость
- Наличие средств описания грамматик

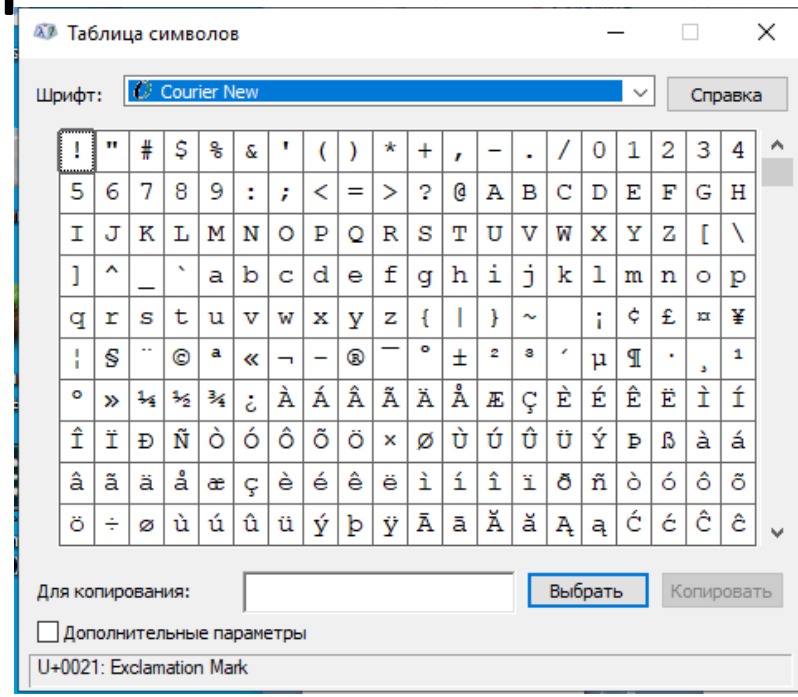
Как устроен формат XML?

```
<?xml version=“1.0” encoding=“windows-1251” ?>
<!DOCTYPE … Спецификации и определения
>
<mainelement>
  Тексты
   <sublement1 att1=“value1” att2=“value2”>
      Texts
      <subsubelement>
          …
       </subsubelement>
    </sublement1>
   Тексты
</mainelement>
```

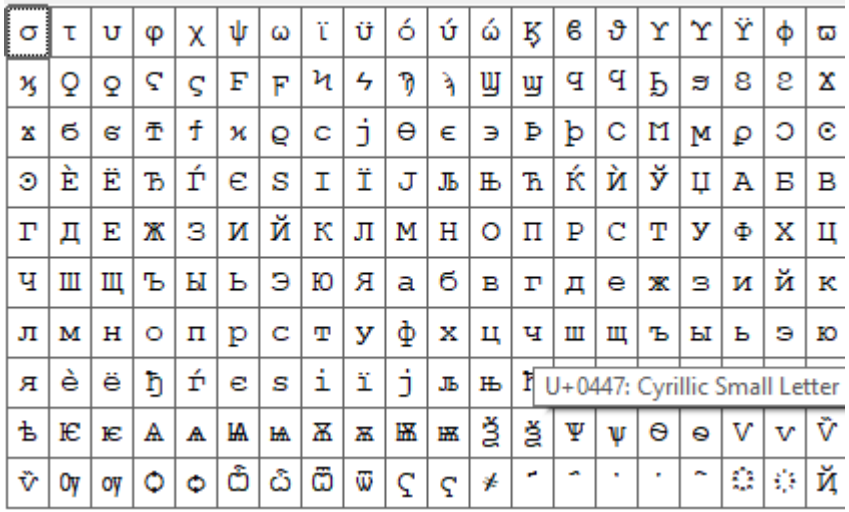
Алфавит

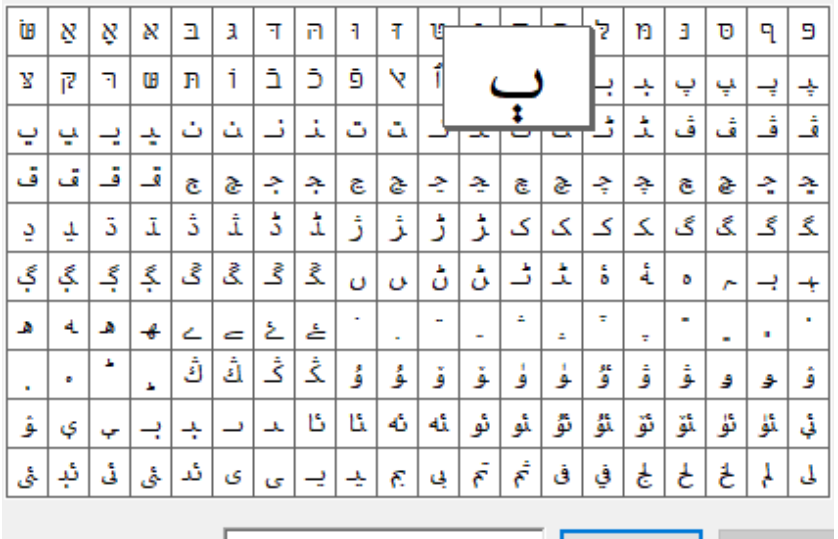
Текст – это набор символов. Что такое символ?

Старый подход: символ это байт (8 битов). Кодируется до 256 символов



Новый подход: символ это 2 байта (16 битов). Кодируется до 65536 символов





Кодировки

ISO 646

- ASCII

Кодировки Microsoft Windows:

- Windows-1250 для языков Центральной Европы, которые используют латинское написание букв (польский, чешский, словацкий, венгерский, словенский, хорватский, румынский и албанский)
- Windows-1251 для кириллических алфавитов
- Windows-1252 для западных языков
- Windows-1253 для греческого языка
- Windows-1254 для турецкого языка
- Windows-1255 для иврита
- Windows-1256 для арабского языка
- Windows-1257 для балтийских языков
- Windows-1258 для вьетнамского языка

MacRoman, MacCyrillic

КОИ8 (KOI8-R, KOI8-U…), КОИ-7

Code Page (8 битов) – таблица из 256 значений символов Unicode

Кодирование задается прямо в XML-документе

<?xml version=“1.0” encoding=“utf-8” ?>
…

Типовые кодировки: utf-8, windows-1251, koi8-r
Редкая кодировка: utf-16 (по 2 байта на символ)

Лексическая структура XML-документа

<?xml version=‘1.0’ encoding=‘…’ ?>
…
<element att1=‘value1’ att2=‘value “2”’ att3=“value ‘3’ and ‘4’”>
… Произвольный текст, кавычки в нем – просто символы …
… Специальные символы: < > & " (“) ' (‘)

… Числовые коды символов: &#D; напр. **å** — (в десятичной форме) представляет букву «а» с маленьким кружком над ней (используется, например, в норвежском языке);
水 — (в шестнадцатеричном) представляет китайский символ для воды
</element>

Лексическая структура XML-документа (2)

<?xml version=‘1.0’ encoding=‘…’ ?>
<!-- Это многострочный комментарий -->
<element att1=‘value1’ att2=‘value “2”’ att3=“value ‘3’ and ‘4’”>
Секция CDATA:
<![CDATA[
Здесь может быть что угодно, включая специальные символы < > & …

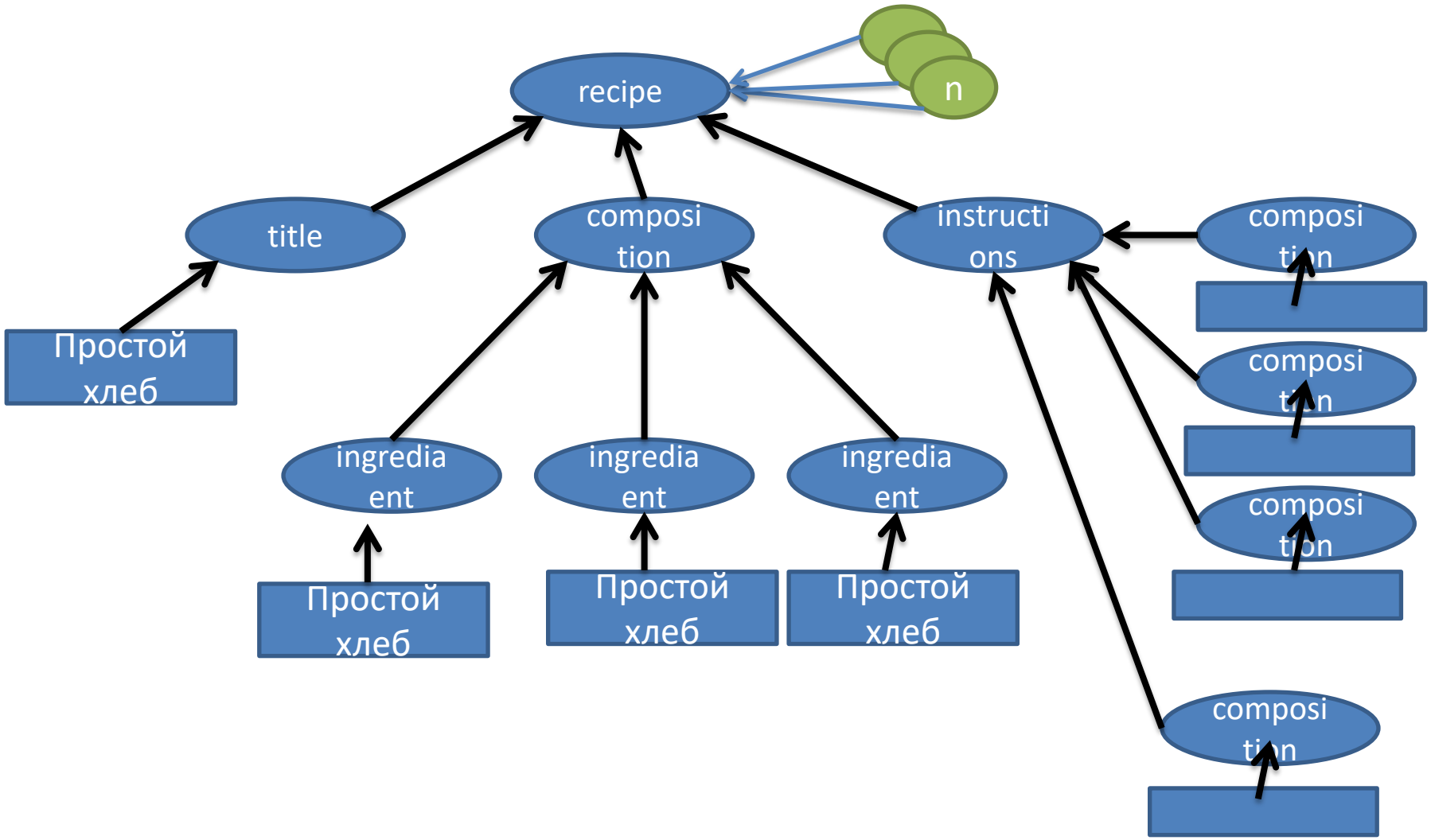
и завершается
]]>

</element>

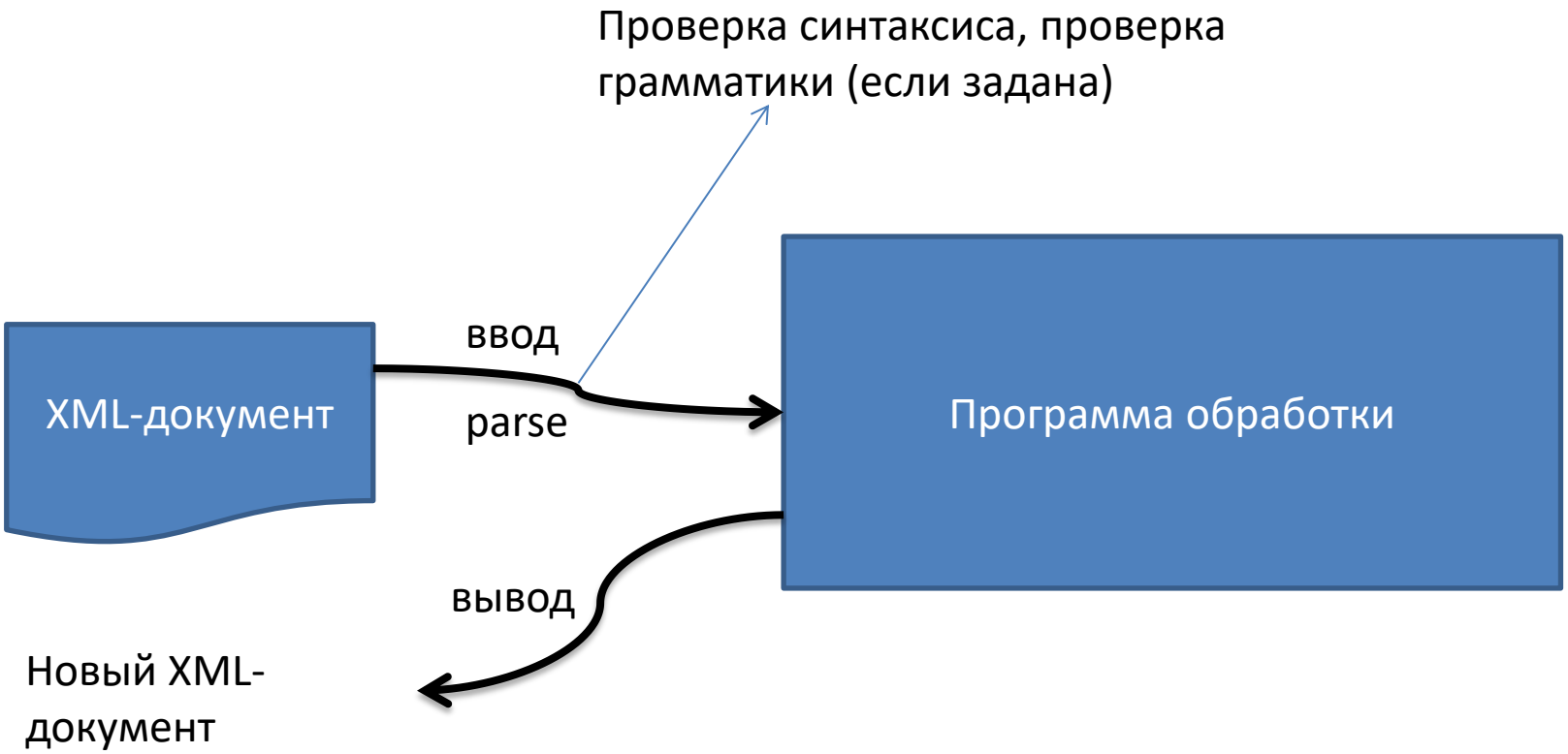
Структура XML-документа

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE recipe>
<recipe name="хлеб" preptime="5min" cooktime="180min">
   <title> Простой хлеб </title>
   <empty></empty> <empty />
   <composition>
      <ingredient amount="3" unit="стакан">Мука</ingredient>
      <ingredient amount="0.25" unit="грамм">Дрожжи</ingredient>
      <ingredient amount="1.5" unit="стакан">Тёплая вода</ingredient>
   </composition>
   <instructions>
      <step> Смешать все ингредиенты и тщательно замесить. </step>
      <step> Закрыть тканью и оставить на один час в тёплом помещении. </step>
   <!-- <step> Почитать вчерашнюю газету. </step> - это сомнительный шаг... -->
      <step> Замесить ещё раз, положить на противень и поставить в духовку.
      </step>
   </instructions>
</recipe>
```

Логическая структура XML-значения



Ввод-вывод, DOM – Document Object Model



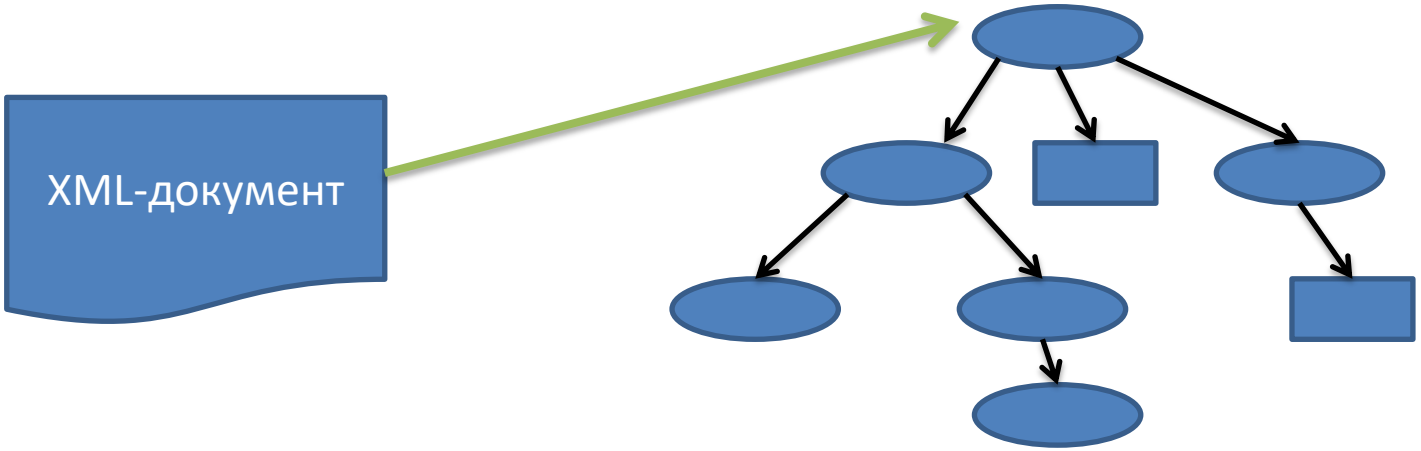
Два способа ввода

1 Событийный ввод (Read)

<?xml version="1.0" encoding="utf-8"?> <!DOCTYPE recipe> **<recipe** name="хлеб" preptime="5min" cooktime="180min"**> <title>** Простой хлеб **</title> <composition> <ingredient** amount="3" unit="стакан"**>**Мука**</ingredient>**

-При вводе порождаются события типа «появился новый элемент», «появился текст», «конец элемента» и др. Ввод в программу осуществляется через (пере)определение реакции на события

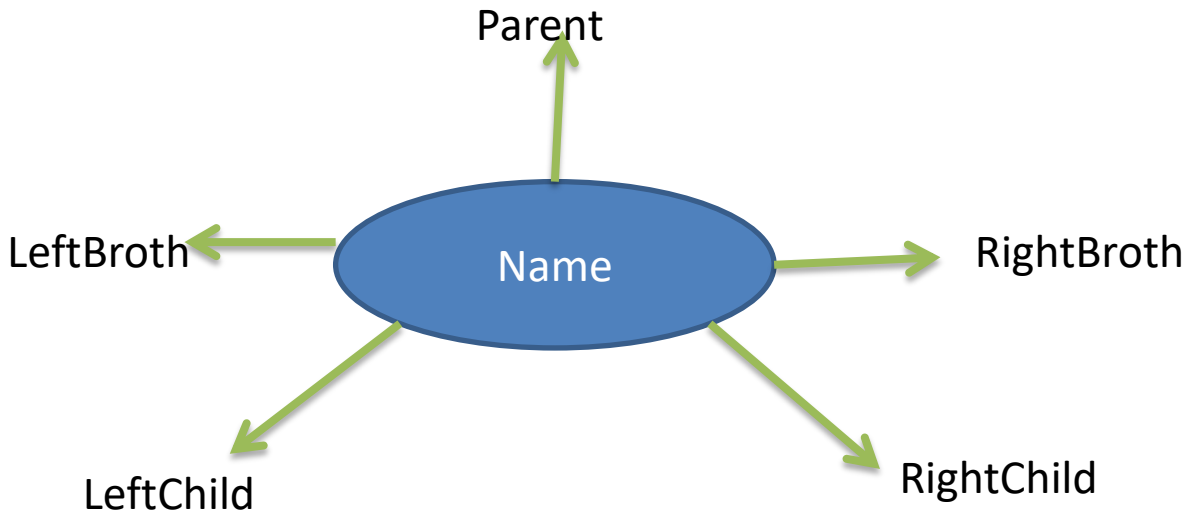
2 Построение дерева (Load)



Стандартное представление дерева: DOM – Document Object Model

```
<recipe name="хлеб" preptime="5min" cooktime="180min">
   <title> Простой хлеб </title>
   <composition>
      <ingredient amount="3" unit="стакан">Мука</ingredient>
      <ingredient amount="0.25" unit="грамм">Дрожжи</ingredient>
      <ingredient amount="1.5" unit="стакан">Тёплая вода</ingredient>
    </composition>
    <instructions>
       <step> Смешать все ингредиенты и тщательно замесить.
       ….
```

XMLDocument
XMLElement
XMLText
XMLNode
XMLAttribute



Функциональное объектное представление XML

Классы объектов:
XDocument
XElement
XAttribute
XName
XText

Конструкторы позволяют задавать иерархию элементов/подэлементов/аттрибутов/текстов

```
XElement html = new XElement(“html”,
        new XElement(“head”, …),
        new XElement(“body”,
                new XElement(“h1”, “Пример”),
                new XElement(“img”, new Xattribute(“src”, “images/pic1.jpg”),
…
```

Методы классов позволяют гибко программировать обработку в функциональном стиле

```
<?xml version="1.0" encoding="utf-8" ?>
<db>
  <person id="p_001">
    <name>Иванов</name>
    <age>21</age>
  </person>
  <person id="p_002">
    <name>Петров</name>
    <age>19</age>
  </person>
  <person id="p_003">
    <name>Сидоров</name>
    <age>22</age>
  </person>
</db>
```

Методы классов позволяют гибко программировать обработку в функциональном стиле

File.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<db>
  <person id="p_001">
    <name>Иванов</name>
    <age>21</age>
  </person>
  <person id="p_002">
    <name>Петров</name>
    <age>19</age>
  </person>
  <person id="p_003">
    <name>Сидоров</name>
    <age>22</age>
  </person>
</db>
```

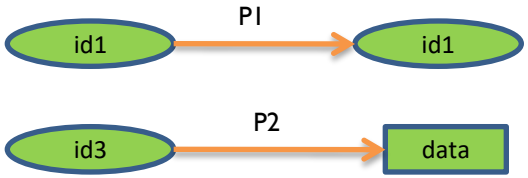
```
XElement db = XElement.Load("File.xml");
var query = db.Elements()
        .Where(x => x.Attribute("id").Value == "p_002")
        .First();
```

Онтологии

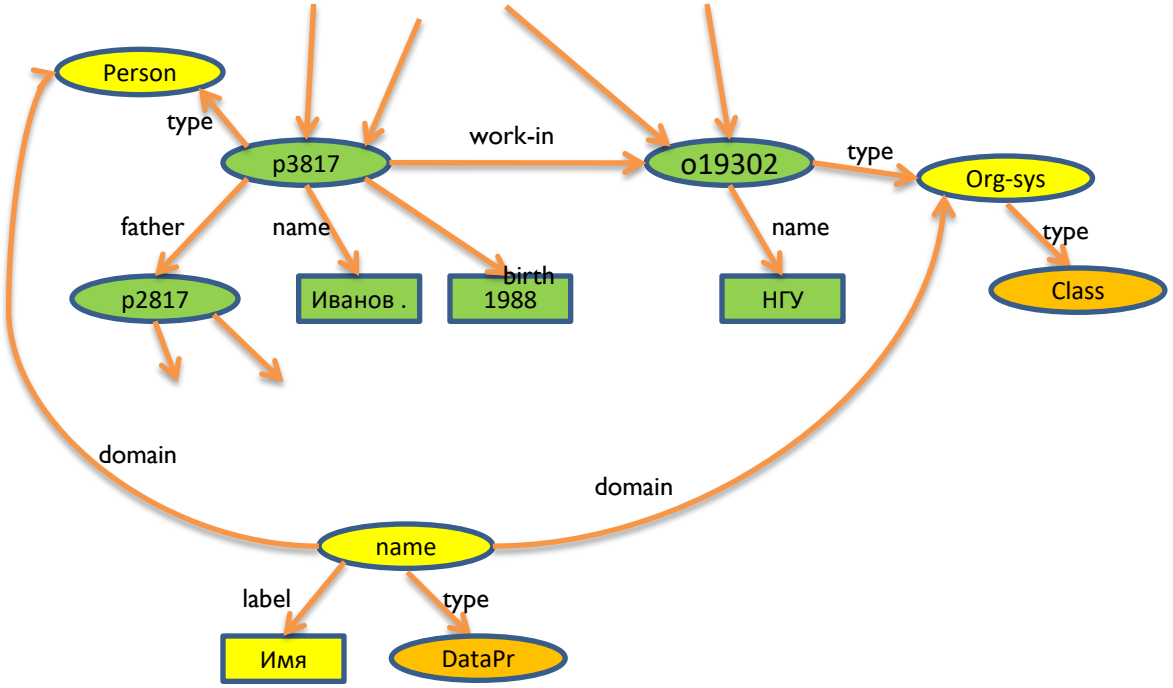
- Онтология – это концептуальная спецификация данных (Спецификация концептуализации)
- Это способ определения понятий как множеств объектов, свойств этих понятий, отношений между понятиями и терминов, соответствующим понятиям

Добавление онтологии

Базовые элементы



Фрагмент сети RDF



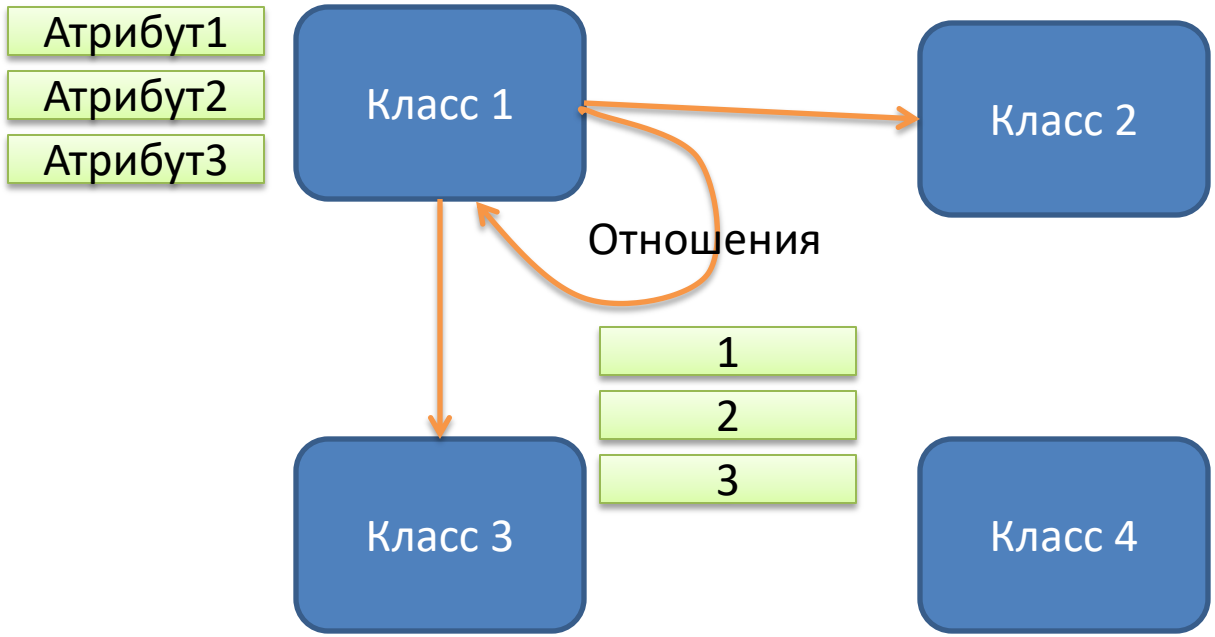
Онтология

Некоторые используемые онтологии

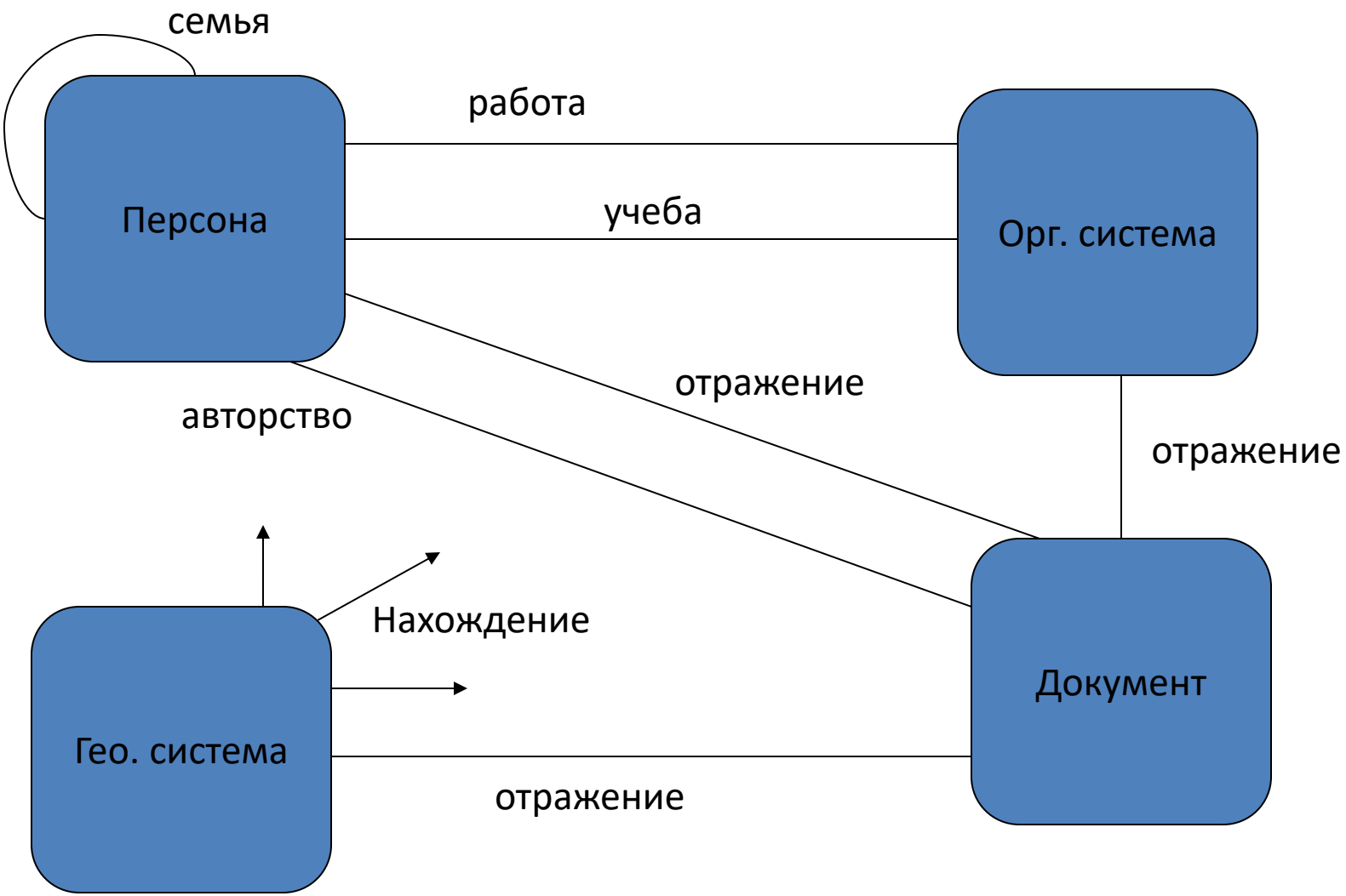
Friend-of-a-Friend (FOAF), vocabulary for describing people.
Dublin Core (DC) defines general metadata attributes. See also their new domains and ranges draft.
Semantically-Interlinked Online Communities (SIOC), vocabulary for representing online communities.
Description of a Project (DOAP), vocabulary for describing projects.
Simple Knowledge Organization System (SKOS), vocabulary for representing taxonomies and loosely structured knowledge.
Music Ontology provides terms for describing artists, albums and tracks.
Review Vocabulary, vocabulary for representing reviews.
Creative Commons (CC), vocabulary for describing license terms.

Онтология определяет:

- Классы (иерархию классов)
- Свойства (DatatypeProperty, ObjectProperty)



Базовая онтология



Всё!