



XML – WARSZTATY PRAKTYCZNE W ZAKRESIE NIEZBĘDNYM DO REALIZACJI ZADAŃ ZWIĄZANYCH Z INSPIRE

Przemysław Biegański, Michał Kubicki
Wrocławski Instytut Zastosowań Informatyki
Przestrzennej i Sztucznej inteligencji



Warszawa, wrzesień-październik 2016 r



PROGRAM SZKOLENIA

- Wprowadzenie do języka XML
- Podstawy składni XML
- Język ścieżek XML (XPath)
- Wprowadzenie do schematów XSD
- Schemat XSD – budowa oraz notacja
- Walidacja dokumentów XML
- Interpretacja błędów walidacji
- XML a INSPIRE – rola oraz zastosowanie



WPROWADZENIE DO XML



SŁOWEM WSTĘPU

INTEROPERACYJNOŚĆ

- Zdolność do współdziałania, komunikowania.
- Umożliwia dokonywanie transferu danych pomiędzy różnymi systemami GIS.
- Obecnie mamy do czynienia z rozproszonymi systemami i usługami GIS, które współpracują za pośrednictwem sieci Internet.



SŁOWEM WSTĘPU

STANDARYZACJA DANYCH PRZESTRZENNYCH

- Standaryzacja umożliwia porozumiewanie się w warunkach różnorodności.
- Celem standaryzacji danych przestrzennych jest zapewnienie ich odpowiedniej jakości przy uwzględnieniu postępu technologicznego oraz rosnących wymagań ze strony użytkowników.
- Standaryzacja w GISie – zapewnienie interoperacyjności systemów.

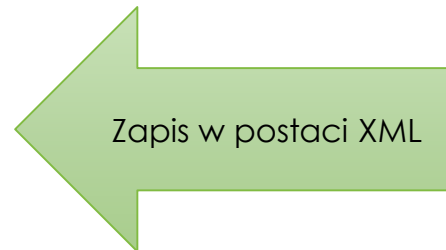


WPROWADZENIE DO XML



<http://learncebuano.com/wp-content/uploads/2015/07/Simple-Filipino-Fluency.png>

XML jest językiem przeznaczonym do opisu informacji oraz danych w postaci tekstu.





WPROWADZENIE DO XML

- Extensible Markup Language – **rozszerzalny** język znaczników.
- Prosty format tekstowy przeznaczony do wymiany danych w Internecie.

```
<tytuł> Nowa książka </tytuł>  
<autor> Pewien znany autor </autor>
```

XML jest łatwy do interpretacji, zarówno przez oprogramowanie, jak i przez użytkownika.



WPROWADZENIE DO XML

Czym jest XML?

- Język został zaprojektowany do opisu danych, nie natomiast do ich graficznej prezentacji.
- Znaczniki (tagi) nie są predefiniowane, należy je zdefiniować samemu.
- Jest językiem samo opisującym się (z ang. self describing).

XML jest formatem reprezentacji danych,
nie zawiera więc informacji o tym jak:

- wyświetlać dane,
- przetwarzać dane.



WPROWADZENIE DO XML

Dlaczego XML?

- Dokumenty XML można w łatwy sposób czytać oraz interpretować.
- Do opracowania dokumentu XML wystarczy dowolny edytor tekstowy.
- Dokumenty XML można przetwarzać za pomocą komercyjnych narzędzi np. Altova XMLSpy, oXygen XML itd.
- Praca z plikami XML jest też możliwa za pomocą oprogramowania niekomercyjnego/open source, np. Notepad++, XMLFox, CAM Template Editor.

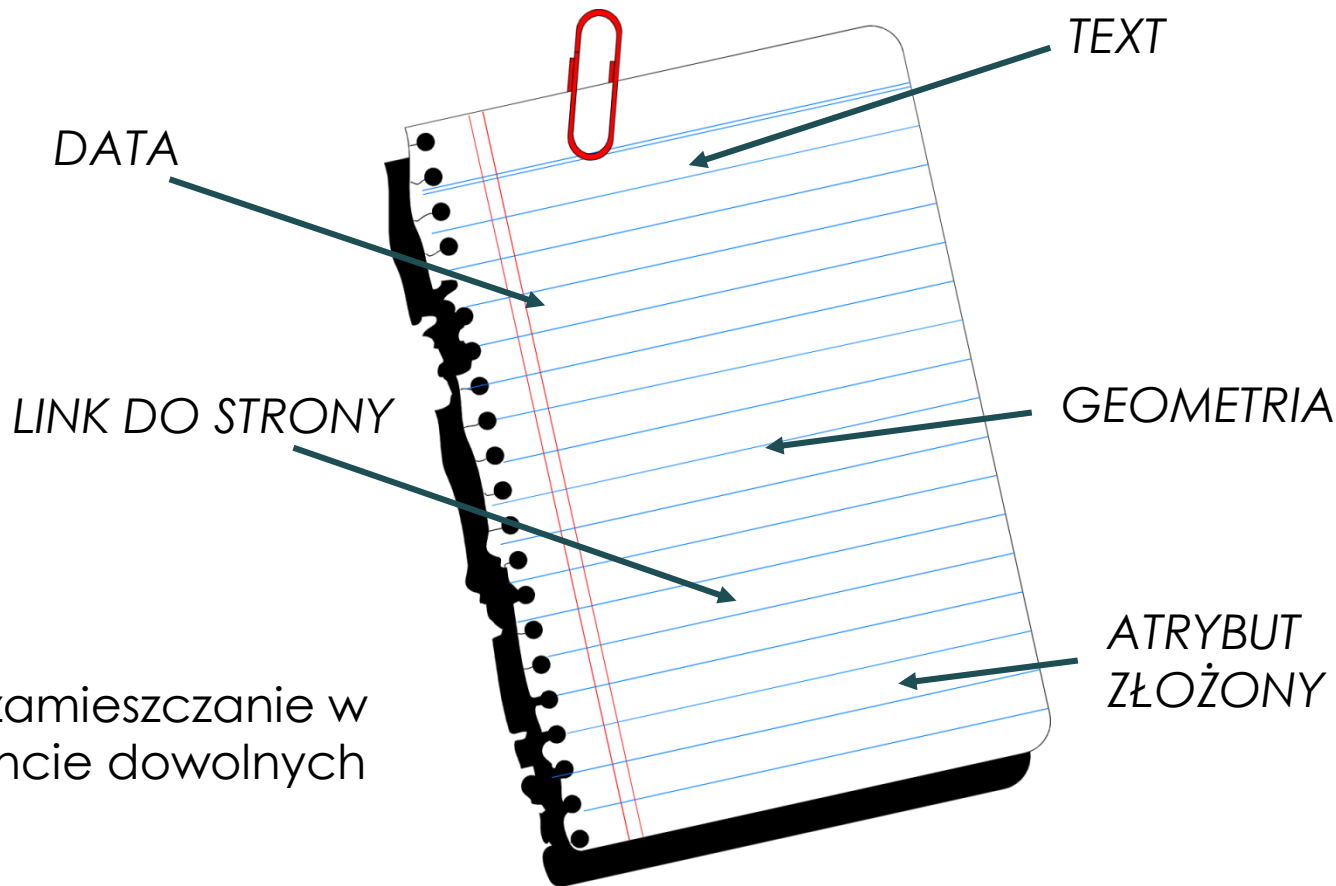


http://4.bp.blogspot.com/-OvASJCVmhWw/VGuJax4V6YI/AAAAAAAAAJbl/7wPQ_f9CZkY/s1600/xml.png



WPROWADZENIE DO XML

Dlaczego XML?



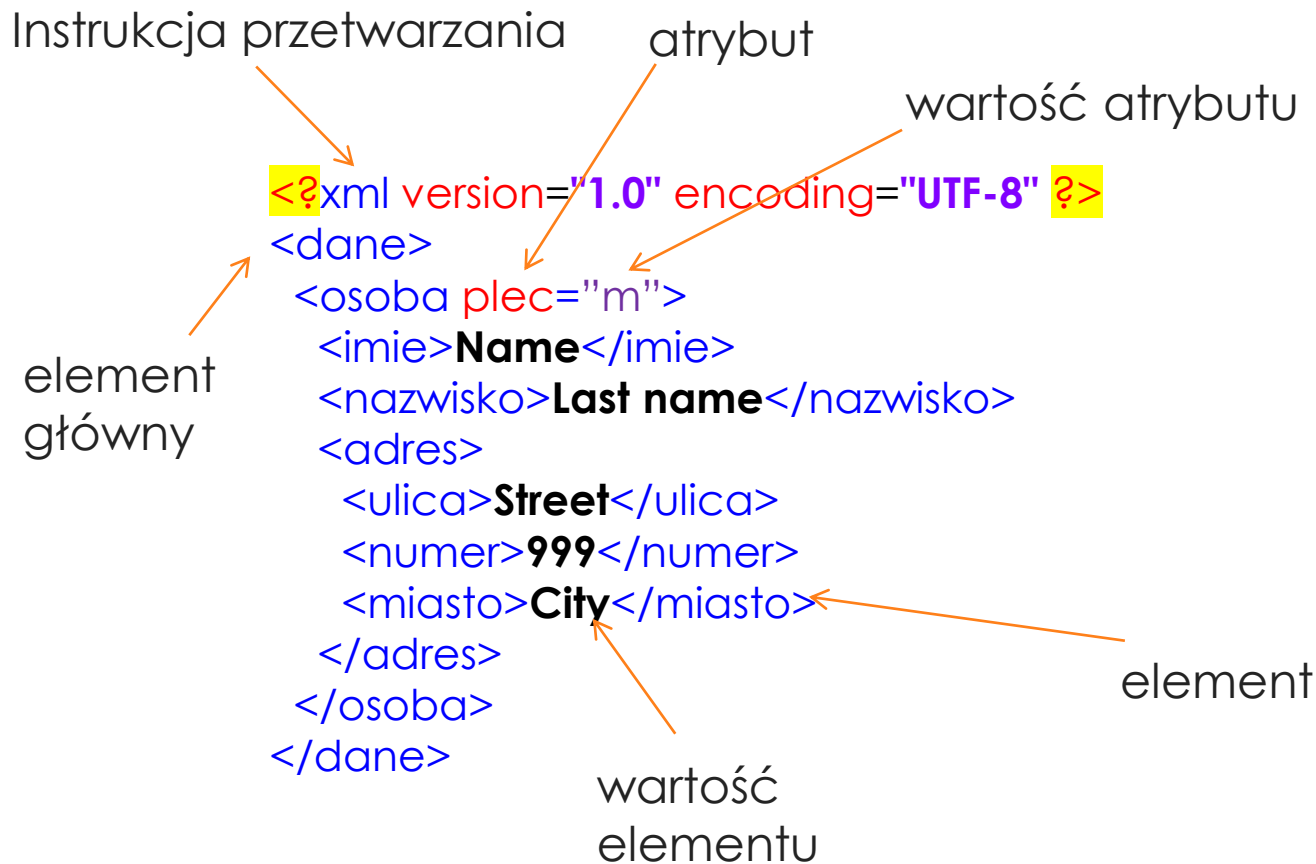
XML dopuszcza zamieszczanie w jednym dokumencie dowolnych typów danych.

<http://www.clipartkid.com/images/140/paper-free-stock-photo-illustration-of-a-blank-notebook-paper-DxEBTK-clipart.png>

PODSTAWY SKŁADNI XML

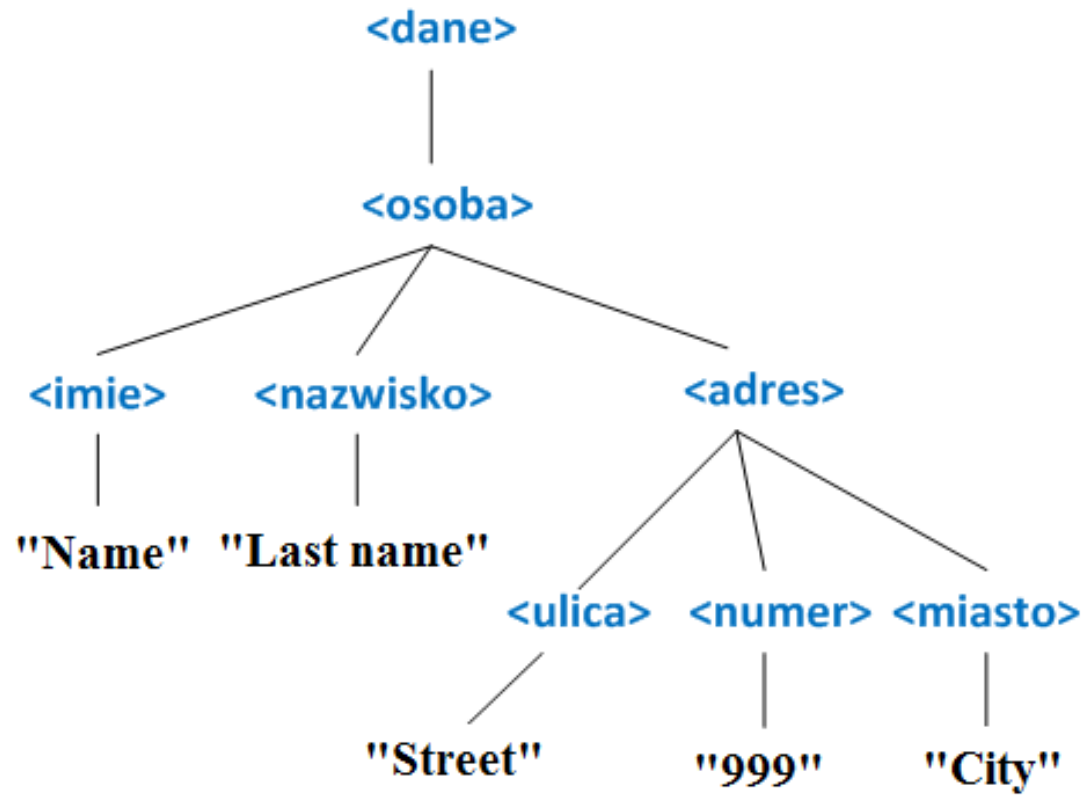


PRZYKŁAD PLIKU XML





DRZEWO DOKUMENTU XML





DANE TABELARYCZNE, A XML

ID-Działki	Właściciel	Powierzchnia
1	Adam Nowak	12.5
2	Jan Kowalski	10.0

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Tabela>
```

```
<Parcela>
```

```
<ID-Działki>1</Parcel-ID>
```

```
<Właściciel>Adam Nowak</Właściciel>
```

```
<Powierzchnia>12.5</Powierzchnia>
```

```
</Parcela>
```

```
<Parcela>
```

```
<ID-Działki>2</Parcel-ID>
```

```
<Właściciel>Jan Kowalski</Właściciel>
```

```
<Powierzchnia>10.0</Powierzchnia>
```

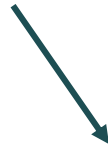
```
</Parcela>
```

```
</Tabela>
```




DEKLARACJA PLIKU XML

Wersja specyfikacji XML



`<?xml version="1.0" encoding="UTF-8" ?>`

Rodzaj dokumentu



Kodowanie znaków
w dokumencie

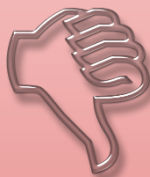




PODSTAWY SKŁADNI XML

Wszystkie elementy w pliku XML muszą mieć znacznik zamykający.

`<p>This is a paragraph`
`<p>This is another paragraph`



`<p>This is a paragraph</p>`
`<p>This is another paragraph</p>`

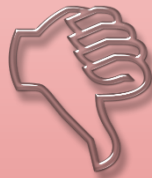




PODSTAWY SKŁADNI XML

W przypadku znaczników XML,
wielkość liter MA znaczenie.

`<i>This text is bold and
italic</i>`



`<i>This text is bold and
italic</i>`

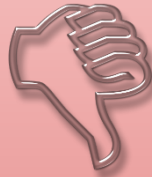




PODSTAWY SKŁADNI XML

Każdy dokument XML musi mieć dokładnie jeden znacznik główny (z ang. root element).

```
<child>TEXT</child>  
<child>TEXT</child>
```



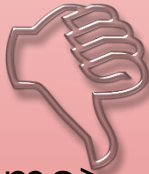
```
<root>  
  <child>TEXT</child>  
  <child>TEXT</child>  
</root>
```






PODSTAWY SKŁADNI XML

Plik XML może posiadać atrybuty w postaci par nazwa=wartość. Wartości należy zawsze zapisywać w cudzysłowie.



```
<person sex=female>  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```



```
<person sex="female">  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```



XML – PRZESTRZENIE NAZW

Przestrzeń nazw (z ang. **namespace**) stosuje się w celu uniknięcia konfliktów dotyczących nazw elementów.

```
<table>
  <tr>
    <td>Treść</td>
    <td>Treść</td>
  </tr>
</table>
```

```
<table>
  <price>Treść</name>
  <width>Treść</width>
  <length>Treść</length>
</table>
```

Jednakowa nazwa elementu,
lecz inne znaczenie:

- „tabela” (po lewej)
- „stół” (po prawej)



XML – PRZESTRZENIE NAZW

Konfliktów można unikać poprzez stosowanie **prefiksów** przed nazwami elementów.

```
<a:table>
  <a:tr>
    <a:td>Treść</a:td>
    <a:td>Treść</a:td>
  </a:tr>
</a:table>
```

a?
„tabela”

```
<b:table>
  <b:price>Treść</b:name>
  <b:width>Treść</b:width>
  <b:length>Treść</b:length>
</b:table>
```

b?
„stół”



XML – PRZESTRZENIE NAZW

Przestrzenie nazw muszą zostać zdefiniowane w dokumencie. Każdy **prefix** musi zostać zdefiniowany przez atrybut **xmlns**.

xmlns:prefix = „URI”

```
<a:table xmlns:a="http://www.w3.org/TR/html4/">
<a:table>
  <a:tr>
    <a:td>Treść</a:td>
    <a:td>Treść</a:td>
  </a:tr>
</a:table>
```



Przykładowa deklaracja.



XML – PRZESTRZENIE NAZW

```
<a:table xmlns:a=http://www.w3.org/TR/html4/  
xmlns:b=http://www.w3.org/TR/html3/>
```

```
<a:table>  
  <a:tr>  
    <b:td>Treść</a:td>  
    <b:td>Treść</a:td>  
  </a:tr>  
</a:table>
```



XML – DOMYŚLNA PRZESTRZEŃ NAZW

`xmlns="namespaceURI"`

```
<table xmlns:h="http://www.w3.org/TR/html4/">
<table>
  <tr>
    <td>Treść</td>
    <td>Treść</td>
  </tr>
</table>
```



XML – PRZESTRZENIE NAZW OMÓWIENIE PRZYKŁADU

Plik omawiany przez trenera (przykład_wyklad.xml)
znajduje się w danych szkoleniowych, w folderze
pliki_do_omowienia.



CZEŚĆ PRAKTYCZNA (ĆWICZENIA)

Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział I – Podstawy składni XML

Ćwiczenie 1 – Tworzenie XML – typy proste
Ćwiczenie 2 – Tworzenie XML – typy złożone

JĘZYK ŚCIEŻEK XML



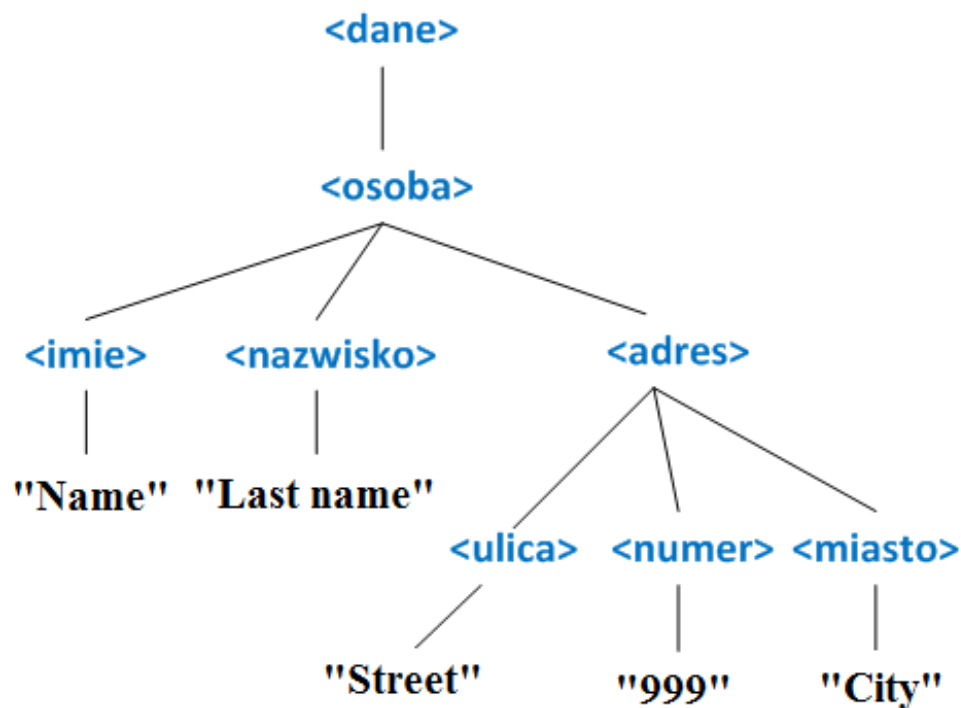


CZYM JEST XPATH?

- **XPath** (ang. XML Path Language), w wolnym tłumaczeniu *Język ścieżek XML* to język służący do adresowania części dokumentu XML.
- **XPath** został oryginalnie zaprojektowany dla XSLT, ale znajduje zastosowanie także w DOM oraz językach bazujących na XML.



CZYM JEST XPATH?



Istotne węzły:

- Root (korzeń)
- Element
- Attribute (atrybut)

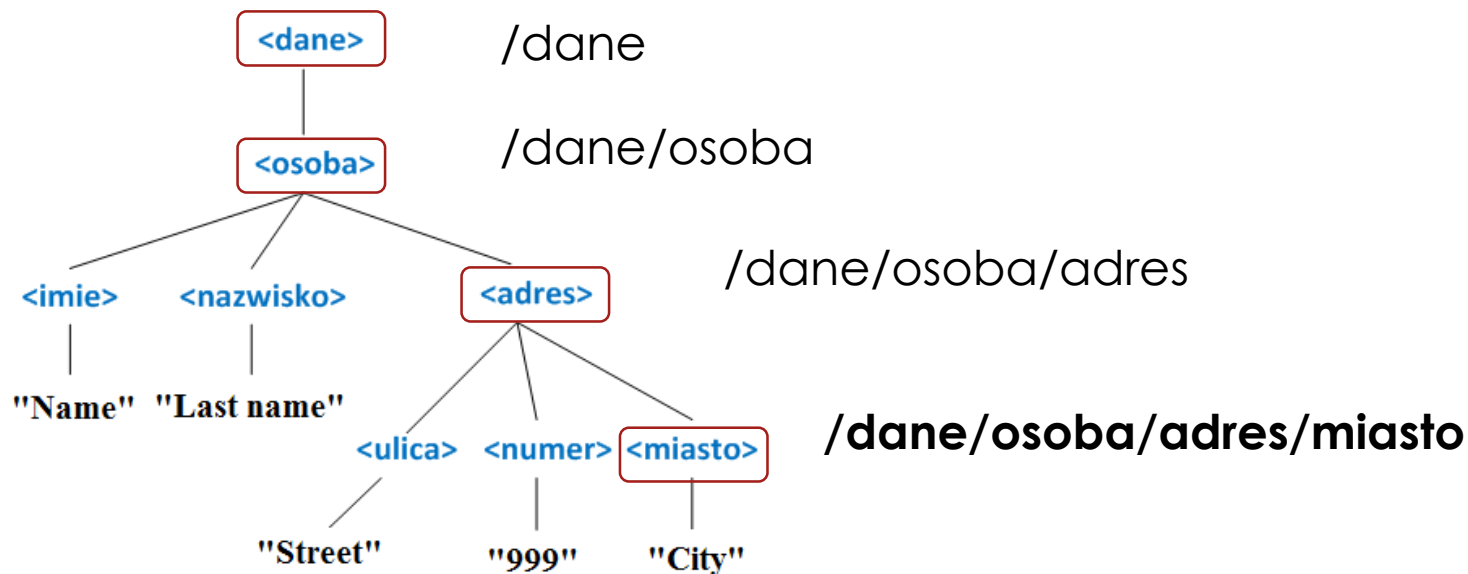
Istotne pojęcia:

- Child (dziecko)
- Parent (rodzic)
- Descendant (potomek)
- Ancestor (przodek)
- Sibling (rodzeństwo)



JĘZYK ŚCIEŻEK XML

Zakodowana ścieżka do elementów pliku XML.





JĘZYK ŚCIEŻEK XML

Ścieżka do atrybutów:

```
<?xml version="1.0" encoding="utf-8"?>
<dane>
  <osoba plec="m">
    <imie>Name</imie>
    <nazwisko>Last Name</nazwisko>
  </osoba>
</dane>
```

/dane/osoba/@plec

ścieżka do elementu atrybut

- dowolny element w drzewie
/dane/osoba/* – dotyczy zarówno elementu "imie" jak i "nazwisko"
- dowolny atrybut w elemencie
/dane/osoba/@*



JĘZYK ŚCIEŻEK XML

- element w dowolnym miejscu drzewa

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<dane>
```

```
  <klient>
```

```
    <imie>Name1</imie>
```

```
    <nazwisko>Last name1</nazwisko>
```

```
  </klient>
```

```
  <sprzedawca>
```

```
    <imie>Name2</imie>
```

```
    <nazwisko>Last name2</nazwisko>
```

```
  </sprzedawca>
```

```
</dane>
```

/dane//imie

- kilka ścieżek

/dane//imie | /dane//nazwisko



CZEŚĆ PRAKTYCZNA (ĆWICZENIA)

Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział II – Język XPATH

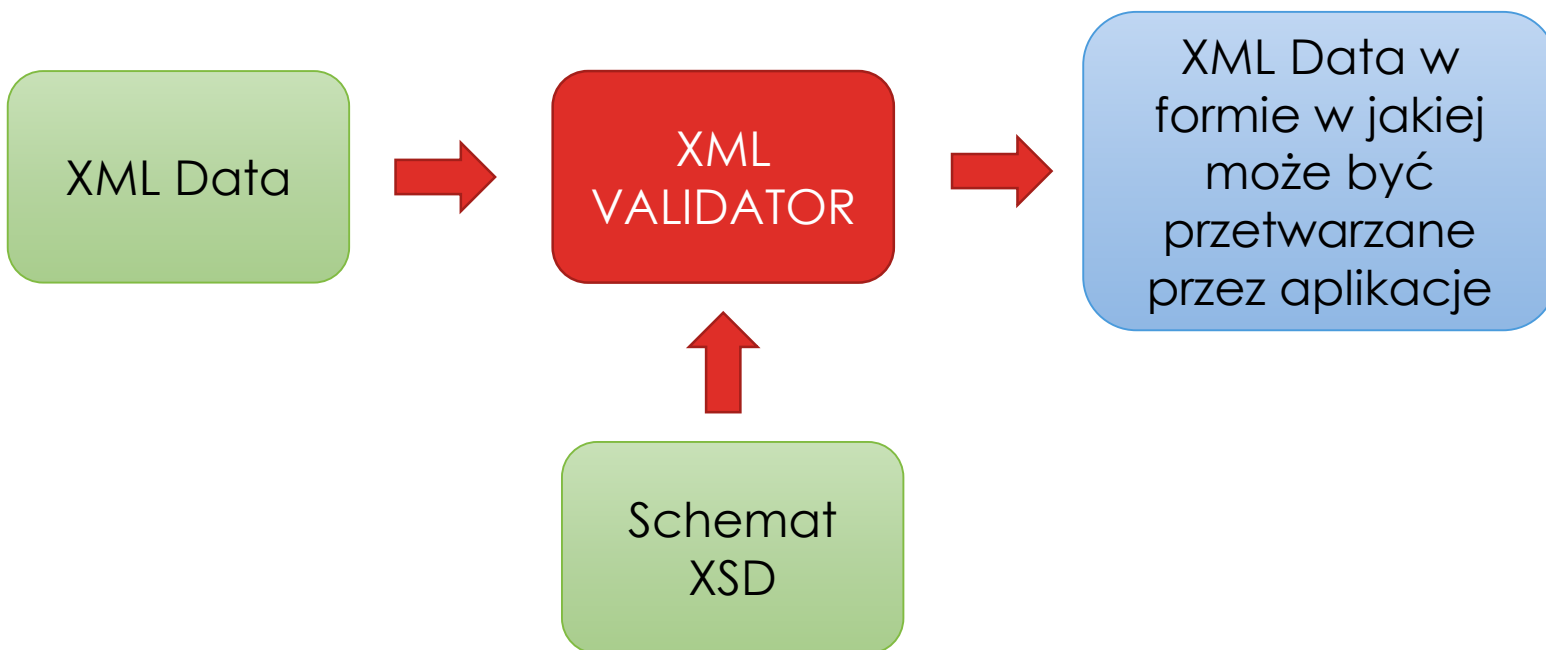
Ćwiczenie 3 – Ścieżki XPATH



SCHEMATY XSD



SCHEMATY XSD





CZYM JEST XSD?

- XSD = XML Schema Definition
- Język opisujący strukturę dokumentów XML, określa:
 - elementy, które mogą pojawić się w dokumencie,
 - atrybuty, które mogą pojawić się w dokumencie,
 - hierarchiczność elementów,
 - kolejność wystąpień elementów potomnych,
 - dopuszczalną liczbę wystąpień elementów potomnych,
 - typy danych dla elementów i atrybutów,
 - ... i wiele innych



CZYM JEST XSD?

- XSD został napisany w XML,
- XSD obsługuje przestrzenie nazw,
- XSD jest rekomendacją W3C,
- XSD jest rozszerzalny.



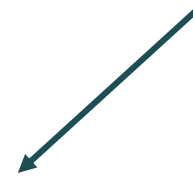
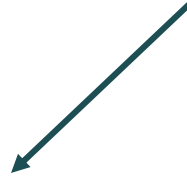
<http://1.bp.blogspot.com/37TkxfuQPHM/VLLEC21tBUI/AAAAAAAAA04/giRIm4JXH9k/s200-c/xsd.png>



SKŁADNIA XSD – ELEMENTY

XSD `<xs:element name=„Random_date„ type="xs:date"/>`

XML `< Random_date >2000-01-12T12:13:14Z</ Random_date >`





SKŁADNIA XSD – ELEMENTY

Default – wartość domyślna, stosuje się gdy brak wartości w pliku XML

Fixed – element w XML może mieć wyłącznie wartość zdefiniowaną w XSD

Przykład kodu XSD:

```
<xs:element name="Customer_name" type="xs:string" default="unknown"/>  
  
<xs:element name="Customer_location" type="xs:string" fixed=" UK"/>
```



SKŁADNIA XSD – RESTRYKCJE

Wykorzystywane w celu ograniczenia akceptowalnych wartości dla atrybutów oraz elementów dokumentu XML.

Element „age” może przyjmować jedynie wartości z zakresu od 0 do 120.

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
<xs:element name="car">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Audi"/>
      <xs:enumeration value="Golf"/>
      <xs:enumeration value="BMW"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Element „car” może przyjmować tylko trzy wartości: Audi, Golf, BMW.



SKŁADNIA XSD – TYPY RESTRYKCJI

Ograniczenie	Opis
enumeration	Lista akceptowalnych wartości
fractionDigits	Dopuszczalna ilość miejsc po przecinku
length	Dokładna ilość znaków/elementów
pattern	Zdefiniowane wyrażenie regularne
totalDigits	Deklaracja dokładnej ilość cyfr
whiteSpace	Deklaracja jak traktować białe znaki
maxLength	Maksymalna dopuszczalna ilość znaków
minLength	Minimalna dopuszczalna ilość znaków



SKŁADNIA XSD – LICZNOŚĆ

```
<xs:element name="hobbies"
```

Dolna granica 0, czyli nie musi wystąpić.

```
type="xs:integer"
```

```
minOccurs="0"
```

```
maxOccurs="unbounded"/>
```

Może wystąpić dowolną ilość razy. Nieograniczona górna granica.

```
<xs:element name="hobbies"
```

```
type="xs:string"
```

```
minOccurs="2"
```

```
maxOccurs="10"/>
```

Minimalna ilość wystąpień 2.

Maksymalna ilość wystąpień 10.



SKŁADNIA XSD – TYPY ZŁOŻONE

```
<xs:element name="Supplier">
```

```
  <xs:complexType>
```

← Deklaracja typu złożonego

```
    <xs:sequence>
```

← Deklaracja sekwencji

```
      <xs:element name="Phone" type="xs:integer"/>
```

```
      <xs:element name="Address" type="xs:string"/>
```

```
    </xs:sequence>
```

```
  </xs:complexType>
```

```
</xs:element>
```

Przykład XML:

```
<Supplier>
```

```
  <Phone>0123987654</Phone>
```

```
  <Address>przykładowy adres</Address>
```

```
</Supplier>
```



SKŁADNIA XSD – TYPY ZŁOŻONE

Kolejność atrybutów w typie złożonym:

- **<xs:sequence>** – elementy muszą wystąpić w porządku zadeklarowanym w pliku XSD,
- **<xs:choice>** – tylko jeden z wymienionych elementów może pojawić się w dokumencie XML,
- **<xs:all>** – elementy mogą wystąpić w XML w dowolnej kolejności.



SKŁADNIA XSD – TYPY ZŁOŻONE

```
<xs:element name="Supplier">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Phone" type="xs:integer"/>
      <xs:element name="Address" type="AddressType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



```
<xs:complexType name="AddressType">
  <xs:sequence>
    <xs:element name="Line1" type="xs:string"/>
    <xs:element name="Line2" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```



SKŁADNIA XSD – ATRYBUTY

```
<xs:element name="Order">
  <xs:complexType>
    <xs:attribute name="OrderID"
      type="xs:int"
      use="required"/>
  </xs:complexType>
</xs:element>
```

← Deklaracja elementu

← Deklaracja atrybutu

← Typ danych dla atrybutu

← Czy atrybut wymagany?
(required/optional)



SKŁADNIA XSD – ATRYBUTY

```
<xs:element name="Order">
```

```
  <xs:complexType>
```

```
    <xs:attribute name="OrderID"
```

```
      type="xs:int"
```

```
      use="required"/>
```

```
  </xs:complexType>
```

```
</xs:element>
```



```
<Order OrderID="6"/>
```



SKŁADNIA XSD – ANY

Zastosowanie elementu **<xs:any>** umożliwia rozszerzenie dokumentu XML o elementy nie zdefiniowane w schemacie.

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <xs:any minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



```
<person>
  <firstname>Stale</firstname>
  <lastname>Refsnes</lastname>
</person>
```



```
<person>
  <firstname>Hege</firstname>
  <lastname>Refsnes</lastname>
  <children>
    <childname>Cecilie</childname>
  </children>
</person>
```




XSD – TYPY DANYCH: STRING

```
<xs:element name="customer" type="xs:token"/>
```

```
<xs:element name="customer" type="xs:string"/>
```

NAZWA	OPIS
string	Dowolny tekst
language	Tekst zawierający prawidłowy identyfikator języka
normalizedString	Tekst bez nowych linii, enterów oraz tabulatorów
token	Tak bez białych znaków (poza pojedynczymi spacjami)
name	Tekst zawierający prawidłowe XML Name.



XSD – TYPY DANYCH: NUMERIC

```
<xs:element name="prize" type="xs:decimal"/>
```

```
<xs:element name="prize" type="xs:long"/>
```

NAZWA	OPIS
integer	Liczba całkowita
positiveInteger	Tylko dodatnie liczby całkowite
long	Liczba całkowita 64 bitowa
decimal	Liczba zmiennoprzecinkowa
byte	Liczba całkowita z przedziału (-128 , +127)



XSD – TYPY DANYCH: DATY

```
<xs:element name="start" type="xs:date"/>
```

```
<xs:element name="start" type="xs:gDay"/>
```

NAZWA	OPIS
date	Data z dokładnością do dnia (YYYY-MM-DD)
dateTime	Data z dokładnością co do sekundy
duration	Do reprezentacji okresu czasu
gDay	Fragment daty – dzień (DD)
gMonth	Fragment daty – miesiąc (MM)



CZEŚĆ PRAKTYCZNA (ĆWICZENIA)

Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział III – Schematy XSD

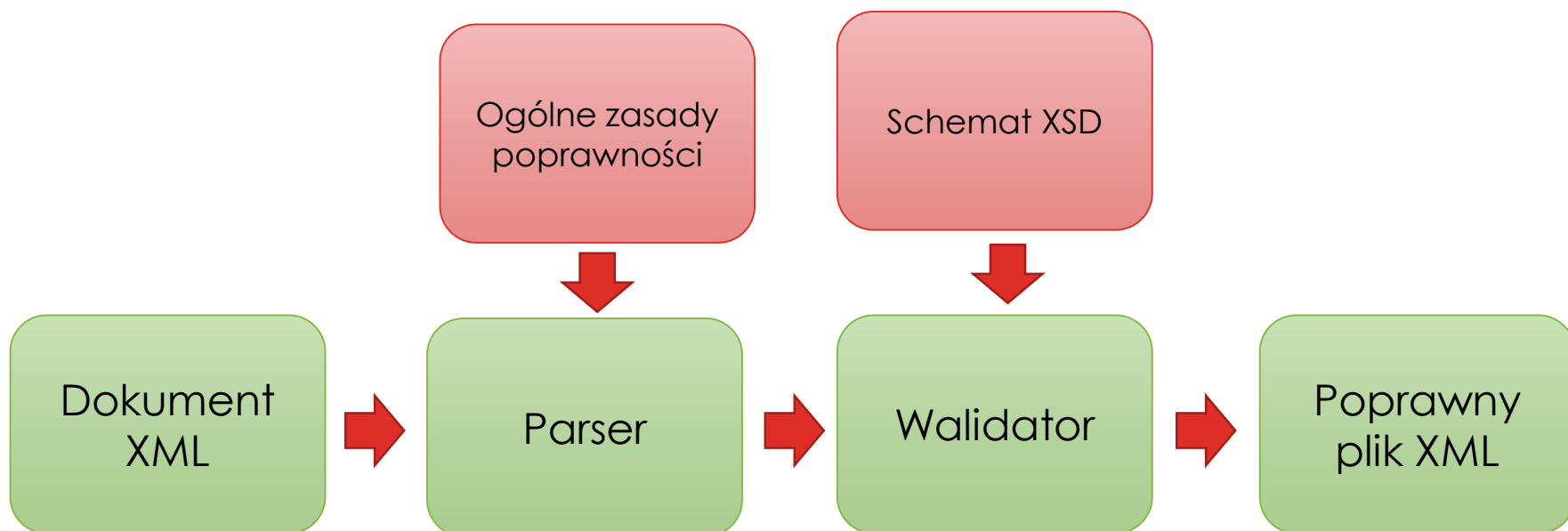
Ćwiczenie 4 – Prosty schemat XSD

Ćwiczenie 5 – Zaawansowany schemat XSD

WALIDACJA DOKUMENTÓW XML



KONTROLA POPRAWNOŚCI XML





KONTROLA POPRAWNOŚCI XML

Ogólne zasady poprawności – kontrola czy dokument jest poprawnie sformułowany (z ang. well-formed):

- umieszczenie deklaracji XML;
- odpowiednie oznaczanie elementów pustych np. `<hr/>`;
- w dokumencie powinien występować element główny;
- odpowiednia kolejność zamykania znaczników;
- umieszczanie wartości atrybutów w cudzysłowie;
- wykorzystanie znaków `<` i `&` tylko do otwierania znaczników i umieszczania odwołań do encji.



KONTROLA POPRAWNOŚCI XML

WALIDACJA — kontrola poprawności dokumentu XML z regułami i ograniczeniami określonymi w schemacie XSD (z którym plik został skojarzony).

Sprawdzenie przeprowadzić można wykonać za pomocą oprogramowania komputerowego przeznaczonego do przetwarzania plików XML, a także za pomocą przeznaczonych do tego serwisów internetowych.



KONTROLA SEMANTYCZNA DANYCH

Poza automatycznymi kontrolami składni dokumentu i jego poprawności względem schematu aplikacyjnego, należy sprawdzić warunki nie dające się zamknąć w zestandaryzowanych ramach:

- poprawne wartości atrybutów – automatyczna kontrola możliwa tylko w typach enumeracyjnych.

INTERPRETACJA BŁĘDÓW WALIDACJI





BŁĘDY SKŁADNIOWE

Niezgodność w nazwach znaczników otwierających i zamykających:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<dane>
```

```
  <klient>
```

```
    <imie>Name1</imie>
```

```
    <nazwisko>Last name1</nazwikso>
```

```
  </klient>
```

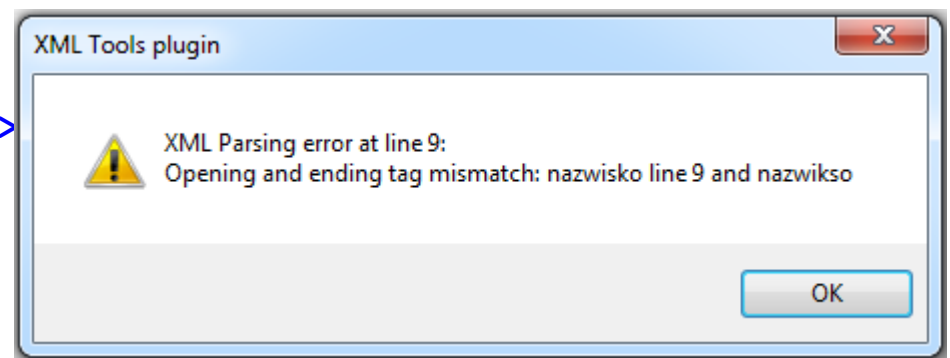
```
  <sprzedawca>
```

```
    <imie>Name2</imie>
```

```
    <nazwisko>Last name2</nazwikso>
```

```
  </sprzedawca>
```

```
</dane>
```





BŁĘDY SKŁADNIOWE

Niedomknięty znacznik:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<dane>
```

```
  <klient>
```

```
    <imie>Name1</imie>
```

```
    <nazwisko>Last name1</nazwisko
```

```
  </klient>
```

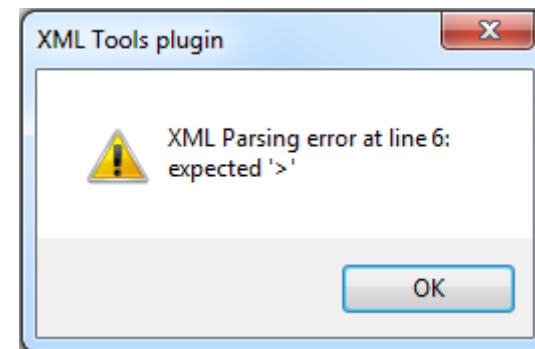
```
  <sprzedawca>
```

```
    <imie>Name2</imie>
```

```
    <nazwisko>Last name2</nazwisko>
```

```
  </sprzedawca>
```

```
</dane>
```





BŁĘDY SKŁADNIOWE

Brak znacznika zamykającego/otwierającego:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<dane>
```

```
  <klient>
```

```
    <imie>Name1</imie>
```

```
    <nazwisko>Last name1
```

```
  </klient>
```

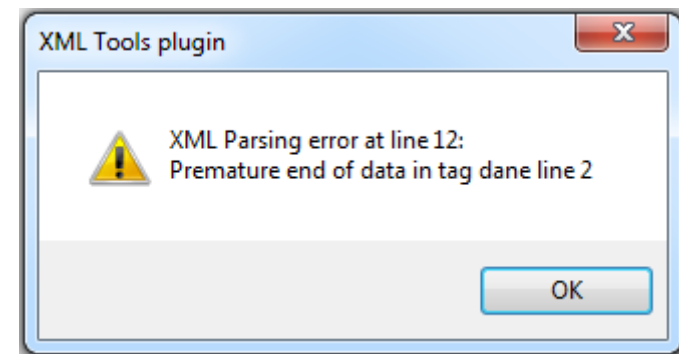
```
  <sprzedawca>
```

```
    <imie>Name2</imie>
```

```
    <nazwisko>Last name2</nazwisko>
```

```
  </sprzedawca>
```

```
</dane>
```





BŁĘDY SKŁADNIOWE

Niepoprawna kolejność znaczników:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<dane>
```

```
  <klient>
```

```
    <imie>Name1</imie>
```

```
    <nazwisko>Last name1
```

```
  </klient></nazwisko>
```

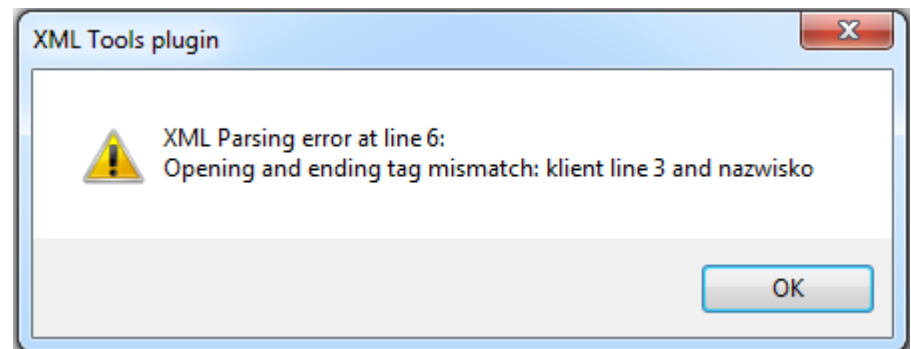
```
  <sprzedawca>
```

```
    <imie>Name2</imie>
```

```
    <nazwisko>Last name2</nazwisko>
```

```
  </sprzedawca>
```

```
</dane>
```



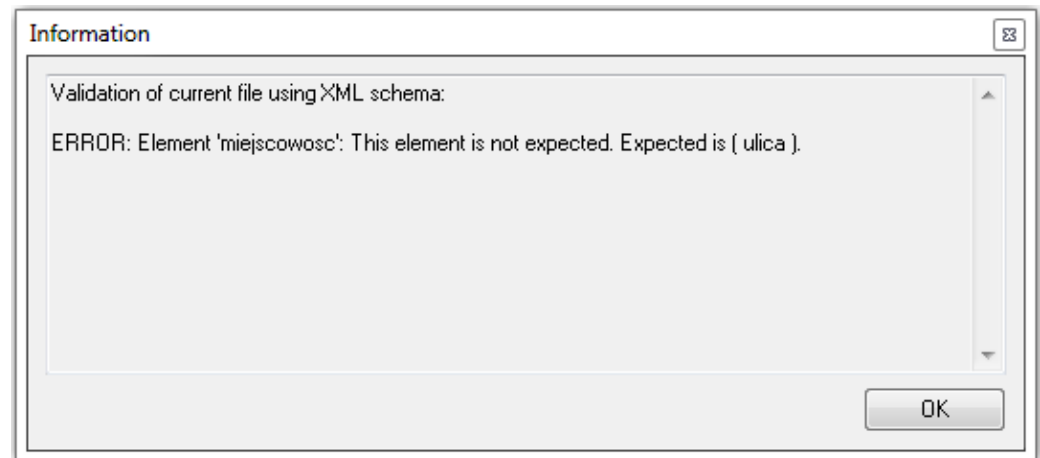


NIEZGODNOŚĆ ZE SCHEMATEM XSD

Nieprawidłowa kolejność elementów:

```
<xs:complexType name="adres">
  <xs:sequence>
    <xs:element name="ulica" type="xs:string"/>
    <xs:element name="miestowosc" type="xs:string"/>
    <xs:element name="kod-pocztowy" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>
```

```
<adres>
  <miestowosc>Wrocław</miestowosc>
  <ulica>Grabiszyńska 241a</ulica>
  <kod-pocztowy>53-234</kod-pocztowy>
</adres>
```



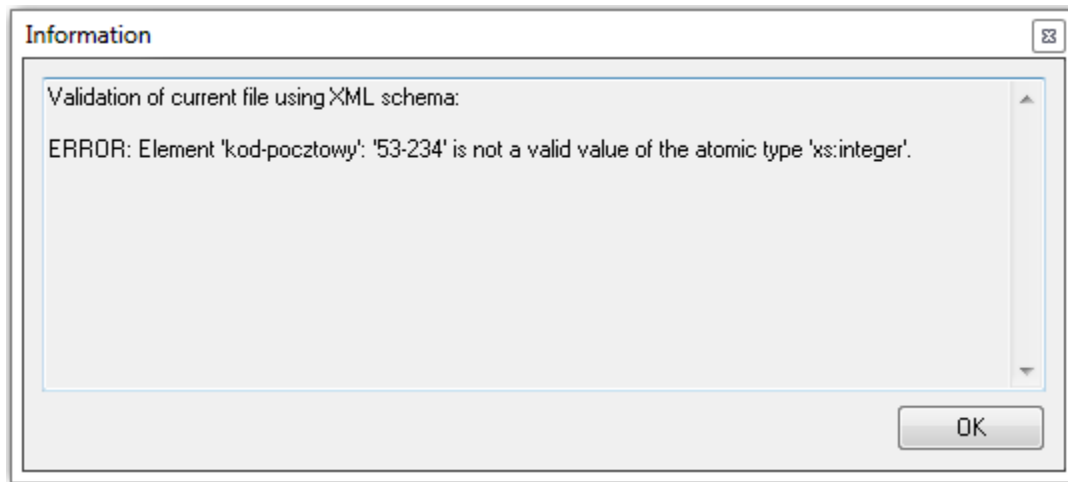


NIEZGODNOŚĆ ZE SCHEMATEM XSD

Nieprawidłowy zakres dziedzinowy wartości atrybutu:

```
<xs:complexType name="adres">
  <xs:sequence>
    <xs:element name="ulica" type="xs:string"/>
    <xs:element name="miestowosc" type="xs:string"/>
    <xs:element name="kod-pocztowy" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>
```

```
<adres>
  <miestowosc>Wrocław</miestowosc>
  <ulica>Grabiszyńska 241a</ulica>
  <kod-pocztowy>53-234</kod-pocztowy>
</adres>
```





CZEŚĆ PRAKTYCZNA (ĆWICZENIA)

Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział IV – Walidacja

Ćwiczenie 6 – Błędy syntaktyczne
Ćwiczenie 7 – Błędy walidacji

XML A INSPIRE ROLA ORAZ ZASTOSOWANIE



XML A INSPIRE – SCHEMATY APLIKACYJNE DANYCH

Język XML ma zastosowanie w INSPIRE dla przedstawienia ich schematów aplikacyjnych w formie plików XSD, które:

- powinny być odzwierciedleniem zapisów specyfikacji w zakresie klas, atrybutów i ról asocjacji zbioru danych INSPIRE w danym temacie;
- umożliwiają wymianę danych pomiędzy schematem źródłowym i INSPIRE (np. w aplikacji HALE);
- umożliwiają walidację zharmonizowanego zbioru danych.



XML A INSPIRE – SCHEMATY APLIKACYJNE DANYCH

(...)

```
<element name="SpeciesDistributionDataSet" substitutionGroup="gml:AbstractFeature" type="sd:SpeciesDistributionDataSetType">
  <annotation>
    <documentation>
  </annotation>
</element>
<complexType name="SpeciesDistributionDataSetType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element minOccurs="0" name="inspireId" type="base:IdentifierPropertyType">
          <annotation>
            <documentation>
          </annotation>
        </element>
        <element name="name" nillable="true">
          <annotation>
            <documentation>
          </annotation>
        </element>
        <complexType>
          <simpleContent>
            <extension base="string">
              <attribute name="nilReason" type="gml:nilReasonType"/>
            </extension>
          </simpleContent>
        </complexType>
      </sequence>
    </extension>
  </complexContent>
</complexType>
</element>
```

ltd...



XML A INSPIRE – ZBIORY DANYCH INSPIRE

GML, który jest podstawowym formatem danych INSPIRE jest odmianą języka XML stworzoną dla przechowywania danych przestrzennych. Reprezentacja danych w ten sposób pozwala na:

- łatwą wymianę danych pomiędzy różnymi systemami – osiągnięcie interoperacyjności;
- automatyzację kontroli zgodności zbioru INSPIRE ze schematem aplikacyjnym.



XML A INSPIRE – ZBIORY DANYCH INSPIRE

(...)

```
<gml:featureMember>
  <sd:SpeciesDistributionDataSet gml:id="Article17SpeciesDistribution_Year2013">
    <gml:identifier codeSpace="http://inspire.ec.europa.eu/ids">
      https://msdi.data.gov.mt/so/sd/SpeciesDistributionDataSet/Article17SpeciesDistribution/Year2013
    </gml:identifier>
    <sd:inspireId>
      <base:Identifier>
        <base:localId>Year2013</base:localId>
        <base:namespace>Article17SpeciesDistribution</base:namespace>
      </base:Identifier>
    </sd:inspireId>
    <sd:name>Article17SpeciesDistribution</sd:name>
    <sd:domainExtent>
      <gml:MultiSurface gml:id="_8747e9ea-436a-4dbc-85d0-69b16ce3a4a2" srsName="urn:ogc:def:crs:EPSG::3045" srsDimension="2">
        <gml:surfaceMember>
          <gml:Polygon gml:id="_87dc4639-ce3c-4626-be57-aa37e23fb6e4">
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList>3991510.03 439138.98 3991510.04 439138.98</gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:MultiSurface>
    </sd:domainExtent>
  </sd:SpeciesDistributionDataSet>
</gml:featureMember>
```

ltd...



XML A INSPIRE – METADANE INSPIRE

XML jest także formatem właściwym dla przechowywania metadanych – także w profilu INSPIRE. Umożliwia to między innymi:

- łatwość edycji i tworzenia metadanych zarówno na poziomie pliku, jak i dedykowanych aplikacji;
- umieszczenie w pliku metadanych dowolnych informacji o zbiorze;
- łatwy eksport do przyjaznych dla użytkownika form, np. dokumentu PDF lub strony internetowej.



XML A INSPIRE – METADANE INSPIRE

(...)

```
<gmd:contact>
  <gmd:CI_ResponsibleParty>
    <gmd:organisationName>
      <gco:CharacterString>Roztoczański Park Narodowy</gco:CharacterString>
    </gmd:organisationName>
    <gmd:contactInfo>
      <gmd:CI_Contact>
        <gmd:address>
          <gmd:CI_Address>
            <gmd:deliveryPoint>
              <gco:CharacterString>Plażowa 2</gco:CharacterString>
            </gmd:deliveryPoint>
            <gmd:city>
              <gco:CharacterString>Zwierzyniec</gco:CharacterString>
            </gmd:city>
            <gmd:postalCode>
              <gco:CharacterString>22-470</gco:CharacterString>
            </gmd:postalCode>
            <gmd:country>
              <gco:CharacterString>Polska</gco:CharacterString>
            </gmd:country>
            <gmd:electronicMailAddress>
              <gco:CharacterString>sekretariat@roztoczanskipn.pl</gco:CharacterString>
            </gmd:electronicMailAddress>
          </gmd:CI_Address>
        </gmd:address>
      </gmd:CI_Contact>
    </gmd:contactInfo>
    <gmd:role>
      <gmd:CI_RoleCode
codeList="http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas/resources/codelist/gmxCodelists.xml#CI_RoleCode"
codeListValue="pointOfContact">punktKontaktowy</gmd:CI_RoleCode>
      </gmd:role>
    </gmd:CI_ResponsibleParty>
  </gmd:contact>
  itd...
```




CZEŚĆ PRAKTYCZNA (ĆWICZENIA)

Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział V – XML i XML Schema w INSPIRE

Ćwiczenie 8 – Analiza fragmentu schematu XSD dla tematu SD