

Wrocławski Instytut Zastosowań Informatyki  
Przestrzennej i Sztucznej Inteligencji

# Materiały szkoleniowe

XML – warsztaty praktyczne w zakresie  
niezbędnym do realizacji zadań  
związanych z INSPIRE

2016-10-12

## Spis treści

<b>1</b>	<b>Wstęp .....</b>	<b>3</b>
<b>2</b>	<b>Rozdział I – Podstawy składni XML.....</b>	<b>3</b>
2.1	Ćwiczenia 1 – Tworzenie XML – typy proste.....	3
2.2	Ćwiczenie 2 – Tworzenie XML – typy złożone. ....	5
<b>3</b>	<b>Rozdział II – Język XPATH .....</b>	<b>7</b>
3.1	Ćwiczenie 3 – Ścieżki XPath .....	7
<b>4</b>	<b>Rozdział III – Schematy XSD .....</b>	<b>9</b>
4.1	Ćwiczenie 4 – Prosty schemat XSD.....	9
4.2	Ćwiczenie 5 – Zaawansowany schemat XSD.....	10
<b>5</b>	<b>Rozdział IV – Walidacja .....</b>	<b>14</b>
5.1	Ćwiczenia 6 – Błędy syntaktyczne.....	14
5.2	Ćwiczenie 7 – Sprawdzenie zgodności pliku XML ze schematem XSD.....	16
<b>6</b>	<b>Rozdział V – XML i XML Schema w INSPIRE .....</b>	<b>18</b>
6.1	Ćwiczenia 8 – Analiza fragmentu schematu XSD dla tematu SD .....	18

## 1 Wstęp

Instruktaż ma na celu zapoznanie uczestników z zaprezentowanym materiałem w postaci praktycznej. Trener prezentuje oraz omawia krok po kroku proces tworzenia przykładowych diagramów. Uczestnicy mają obowiązek wykonywać wszystkie prezentowane kroki równoległe z trenerem.

Ćwiczenia mają za zadanie sprawdzić zrozumienie omawianego tematu przez uczestników. Zadania należy wykonywać samodzielnie, na koniec zajęć przekazane zostaną uczestnikom zeszyty ćwiczeń wraz z przykładowymi odpowiedziami. Zadania ćwiczeniowe nie są liniowe, w większości są to zadania otwarte, w związku z czym mogą mieć wiele prawidłowych rozwiązań. Dodatkowo po każdym ćwiczeniu trener zaprezentuje przykładowe rozwiązanie.

## 2 Rozdział I – Podstawy składni XML

### 2.1 Ćwiczenia 1 – Tworzenie XML – typy proste.

#### **Treść ćwiczenia:**

Proszę utworzyć plik XML na podstawie tabeli z danymi dotyczącymi pomników przyrody w województwie dolnośląskim. Plik ten musi być prawidłowym plikiem XML pod względem gramatyki języka. Należy pamiętać o deklaracji pliku XML oraz elemencie głównym. Na potrzeby tego ćwiczenia założymy, że każdy obiekt składa się wyłącznie z typów prostych (tak jak zostało to zaprezentowane w tabeli).

- Nazwa elementów głównego – „zbiór”,
- Nazwa elementu opisującego pomnik przyrody – „obiekt”,
- Nie wolno używać polskich znaków w nazwach znaczników.

Ćwiczenie należy wykonać w oprogramowaniu Notepad++.

id	nazwa	typ	kodinspire	kod-Kraju	województwo	szerokosc	dlugosc	uklad
30478	Chochół	drzewo	PL.ZIPOP.1393.PP.0201011.64	PL	dolnośląskie	51.2647	15.5621	epsg:4326
30479	Wiertaczy	głaz narzutowy	PL.ZIPOP.1393.PP.0201011.81	PL	dolnośląskie	51.2591	15.5762	epsg:4326
30480	Mostowy	drzewo	PL.ZIPOP.1393.PP.0201011.66	PL	dolnośląskie	51.2733	15.5555	epsg:4326

## Przykładowe rozwiązanie ćwiczenia 1:

```
<?xml version = "1.0" encoding = "utf-8"?>
<zbior>
  <obiekt>
    <id>30478</id>
    <nazwa>Chochoł</nazwa>
    <typ>drzewo</typ>
    <kodInspire>PL.ZIPOP.1393.PP.0201011.64</kodInspire>
    <kodKraju>PL</kodKraju>
    <województwo>dolnośląskie</województwo>
    <szerokosc>51.2647</szerokosc>
    <dlugosc>15.5621</dlugosc>
    <uklad>epsg:4326</uklad>
  </obiekt>
  <obiekt>
    <id>30479</id>
    <nazwa>Wiertaczy</nazwa>
    <typ>głaz narzutowy</typ>
    <kodInspire>PL.ZIPOP.1393.PP.0201011.81</kodInspire>
    <kodKraju>PL</kodKraju>
    <województwo>dolnośląskie</województwo>
    <szerokosc>51.2591</szerokosc>
    <dlugosc>15.5762</dlugosc>
    <uklad>epsg:4326</uklad>
  </obiekt>
  <obiekt>
    <id>30480</id>
    <nazwa>Mostowy</nazwa>
    <typ>drzewo</typ>
    <kodInspire>PL.ZIPOP.1393.PP.0201011.66</kodInspire>
    <kodKraju>PL</kodKraju>
    <województwo>dolnośląskie</województwo>
    <szerokosc>51.2733</szerokosc>
    <dlugosc>15.5555</dlugosc>
    <uklad>epsg:4326</uklad>
  </obiekt>
</zbior>
```

## 2.2 Ćwiczenie 2 – Tworzenie XML – typy złożone.

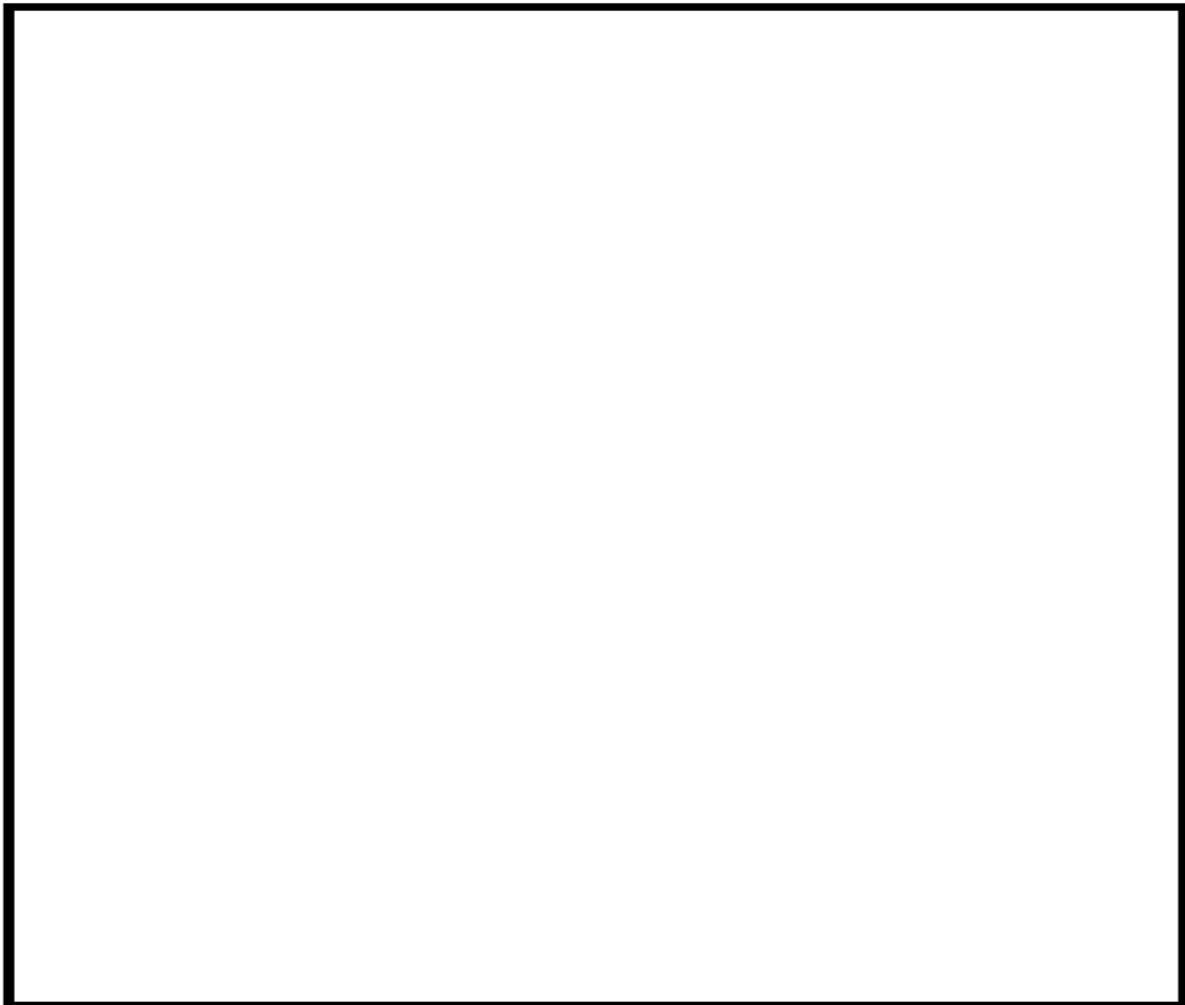
### **Treść ćwiczenia**

Celem ćwiczenia jest zaznajomienie się ze złożonymi typami danych oraz zastosowanie ich w praktyce. Proszę zmodyfikować opracowany w ćwiczeniu 1 plik XML zgodnie z następującymi wytycznymi:

- „id” oraz „układ” powinny zostać użyte jako atrybuty, a nie jako elementy, miejsce w drzewie obiektów należy dobrać tak aby miało to sens,
- Zdefiniowany zostanie typ złożony „wspolrzedne” zawierający w sobie „szerokosc” oraz „dlugosc”,
- Zdefiniowany zostanie typ złożony „lokalizacja” zawierający w sobie „kod\_kraju”, „województwo” oraz „wspolrzedne”.

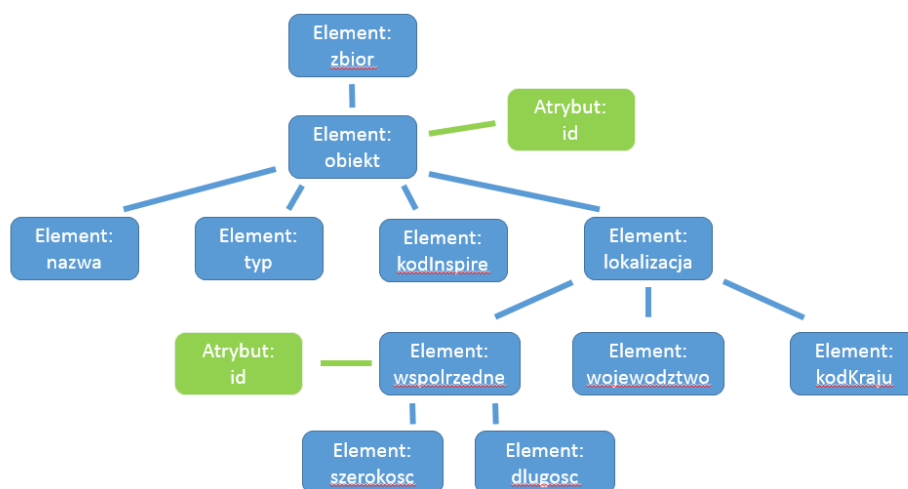
Dodatkowo proszę narysować drzewo DOM dla zmodyfikowanego dokumentu. Proszę w tym celu wykorzystać poniższą przewidzianą do tego przestrzeń.

Miejsce na wyrysowanie drzewa DOM:



## Przykładowe rozwiązanie ćwiczenia 2:

```
<?xml version = "1.0" encoding = "utf-8"?>
<zbior>
  <obiekt id="30478" >
    <nazwa>Chochoł</nazwa>
    <typ>drzewo</typ>
    <kodInspire>PL.ZIPOP.1393.PP.0201011.64</kodInspire>
    <lokalizacja>
      <kodKraju>PL</kodKraju>
      <województwo>dolnośląskie</województwo>
      <wspolrzedne układ="epsg:4326" >
        <szerokosc>51.2647</szerokosc>
        <dlugosc>15.5621</dlugosc>
      </wspolrzedne>
    </lokalizacja>
  </obiekt>
  <obiekt id="30479" >
    <nazwa>Wiertaczy</nazwa>
    <typ>głaz narzutowy</typ>
    <kodInspire>PL.ZIPOP.1393.PP.0201011.81</kodInspire>
    <lokalizacja>
      <kodKraju>PL</kodKraju>
      <województwo>dolnośląskie</województwo>
      <wspolrzedne układ="epsg:4326" >
        <szerokosc>51.2591</szerokosc>
        <dlugosc>15.5762</dlugosc>
      </wspolrzedne>
    </lokalizacja>
  </obiekt>
  <obiekt id="30480" >
    <nazwa>Mostowy</nazwa>
    <typ>drzewo</typ>
    <kodInspire>PL.ZIPOP.1393.PP.0201011.66</kodInspire>
    <lokalizacja>
      <kodKraju>PL</kodKraju>
      <województwo>dolnośląskie</województwo>
      <wspolrzedne układ="epsg:4326">
        <szerokosc>51.2733</szerokosc>
        <dlugosc>15.5555</dlugosc>
      </wspolrzedne>
    </lokalizacja>
  </obiekt>
</zbior>
```



## 3 Rozdział II – Język XPATH

### 3.1 Ćwiczenie 3 – Ścieżki XPath

#### **Treść ćwiczenia**

Za zbiór źródłowy posłuży utworzony w ćwiczeniu poprzednim plik XML z pomnikami przyrody. Należy zmodyfikować wartości dla elementu „województwo” w sposób następujący:

- Dla obiektu o id = „30478”, województwo „dolnośląskie”,
- Dla obiektu o id = „30479”, województwo „mazowieckie”,
- Dla obiektu o id = „30480”, województwo „mazowieckie”.

Ćwiczenie należy przeprowadzić w aplikacji Notepad++ z wykorzystaniem rozszerzenia XMLtools. Są tam dwie przydatne funkcje (z punktu widzenia tego ćwiczenia):

- „Current XML path” – wskazuje ścieżkę do miejsca w pliku XML, gdzie obecnie znajduje się kursor,
- „Evaluate XML expression” - moduł do wykonywania zapytań w języku XPath.

Proszę opracować ścieżki, dzięki którym w efekcie uzyskamy:

- Wszystkie nazwy pomników przyrody:

Odp:

- Wszystkie nazwy pomników przyrody oraz ich typ:

Odp:

- Identyfikatory wszystkich obiektów (atrybut):

Odp:

- Kody Inspire obiektów leżących w województwie mazowieckim:

Odp:

- Liczbę obiektów zlokalizowanych w województwie mazowieckim (count):

Odp:

Przykładowe rozwiązania ćwiczenia 3:

a) Wszystkie nazwy pomników przyrody:

Odp: //nazwa

Type	Name	Value
Node	nazwa	Chochół
Node	nazwa	Wiertaczy
Node	nazwa	Mostowy

b) Wszystkie nazwy pomników przyrody oraz ich typ:

Odp: //nazwa | //typ

Type	Name	Value
Node	nazwa	Chochół
Node	typ	drzewo
Node	nazwa	Wiertaczy
Node	typ	głaz narzutowy
Node	nazwa	Mostowy
Node	typ	drzewo

c) Identyfikatory wszystkich obiektów (atrybut):

Odp: /zbior/obiekt/@id

Type	Name	Value
Attr	id	30478
Attr	id	30479
Attr	id	30480

d) Kody Inspire obiektów leżących w województwie mazowieckim:

Odp: /zbior/obiekt/kodInspire[..]/lokalizacja/wojewodztwo='mazowieckie']

Type	Name	Value
Node	kodInspire	PL.ZIPOP.1393.PP.0201011.81
Node	kodInspire	PL.ZIPOP.1393.PP.0201011.66

e) Liczbę obiektów zlokalizowanych w województwie mazowieckim:

Odp: count(/zbior/obiekt/lokalizacja[wojewodztwo='mazowieckie'])

Type	Name	Value
Num		2.000000



## 4 Rozdział III – Schematy XSD

### 4.1 Ćwiczenie 4 – Prosty schemat XSD

#### **Treść zadania**

Dla dokumentu XML opracowanego w podpunkcie 2.1 proszę opracować schemat XSD. Schemat ten powinien zawierać deklaracje dotyczące kolejności elementów, ich nazw oraz typów danych.

Należy zwrócić uwagę na fakt, iż element „obiekt” występuje trzykrotnie, w związku z czym wymagane jest dodanie informacji o dopuszczalnej krotności dla tego elementu.

Plik XSD powinien zostać opracowany w aplikacji Notepad++, która również posłuży do testów zgodności dokumentu XML ze schematem.

Przypomnienie sposobu deklaracji krotności:

```
<xs:element name="text" minOccurs="0" maxOccurs="1" >
```

Przypomnienie deklaracji typu złożonego:

```
<xs:element name="text">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="text" type="typ_danych"/>  
      <xs:element name="text" type="typ_danych"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

Budowę schematu powinniśmy rozpocząć od deklaracji pliku, określeniu „root element” oraz wskazaniu przestrzeni nazw dla XSD (URI => <http://www.w3.org/2001/XMLSchema>).

## 4.2 Ćwiczenie 5 – Zaawansowany schemat XSD

### Treść zadania

Celem ćwiczenia jest przebudowanie schematu XSD tak aby był zgodny z plikiem opracowanym w punkcie 2.2. Wyjątkowo nie będziemy tworzyć dokumentu XML na podstawie XSD, lecz odwrotnie.

- a) W tym celu należy wprowadzić następujące korekty:
- Wprowadzony zostanie złożony typ danych „wspolzedne” o typie „wspolzedne-Type”, który będzie zawierać szerokość oraz długość,
  - Wprowadzony zostanie złożony element „lokalizacja” o typie „lokalizacjaType”, który będzie zawierać kod kraju, województwo oraz współrzędne,
  - Id stanie się atrybutem elementu „zbior”, natomiast układ stanie się atrybutem dla elementu „wspolzedne”.

Dla przypomnienia poniżej zamieszczona została deklaracja złożonego typu danych z XSD:

```
<xs:complexType name="nazwa typu danych">  
  <xs:sequence>  
    <xs:element name="text" type="typ danych"/>  
    <xs:element name="text" type="typ danych"/>  
  </xs:sequence>  
</xs:complexType>
```

- b) Dodatkowo proszę o rozbudowanie schematu o następujące ograniczenia:
- Szerokość oraz długość będą typu zmiennoprzecinkowego,
  - Szerokość geograficzna może przyjmować wartości od -90 do +90,
  - Długość geograficzna musi przyjmować wartości od -180 do +180,
  - Dla szerokości oraz długości dopuszczalna liczba miejsc po przecinku to 4.

Dla przypomnienia sposób deklarowania ograniczeń (restrykcji):

```
<xs:element name="nazwa elementu">  
  <xs:simpleType>  
    <xs:restriction base="typ danych">  
      <xs:ograniczenie value="wartosc"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

Wykaz dopuszczalnych 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 ilości cyfr
whiteSpace	Deklaracja jak traktować białe znaki
maxLength	Maksymalna dopuszczalna ilość znaków
minLength	Minimalna dopuszczalna ilość znaków
minInclusive	Minimalna dopuszczalna wartość
maxInclusive	Maksymalna dopuszczalna wartość

### c) Przykładowe rozwiązanie zadania 4:

```

<?xml version = "1.0" encoding = "utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="zbior">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="obiekt" minOccurs="0" maxOccurs="unbounded" >
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id" type="xs:string"/>
              <xs:element name="nazwa" type="xs:string"/>
              <xs:element name="typ" type="xs:string"/>
              <xs:element name="kodInspire" type="xs:string"/>
              <xs:element name="kodKraju" type="xs:string"/>
              <xs:element name="województwo" type="xs:string"/>
              <xs:element name="szerokosc" type="xs:string"/>
              <xs:element name="dlugosc" type="xs:string"/>
              <xs:element name="uklad" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
    
```

### Przykładowe rozwiązanie ćwiczenia 4 - A:

```

<?xml version = "1.0" encoding = "utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="zbior">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="obiekt" minOccurs="0" maxOccurs="unbounded" >
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nazwa" type="xs:string"/>
              <xs:element name="typ" type="xs:string"/>
              <xs:element name="kodInspire" type="xs:string"/>
              <xs:element name="lokalizacja" type="lokalizacjaType"/>
            </xs:sequence>
            <xs:attribute name="id"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="lokalizacjaType">
    <xs:sequence>
      <xs:element name="kodKraju" type="xs:string"/>
      <xs:element name="województwo" type="xs:string"/>
      <xs:element name="wspolrzedne" type="wspolrzedneType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="wspolrzedneType">
    <xs:sequence>
      <xs:element name="szerokosc" type="xs:string"/>
      <xs:element name="dlugosc" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="uklad"/>
  </xs:complexType>

</xs:schema>
    
```

Przykładowe rozwiązanie ćwiczenia 5-B:

Elementy nie modyfikowane w stosunku do ćwiczenia poprzedniego, zostały zwinięte w poniższym listingu.

```

<?xml version = "1.0" encoding = "utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="zbior">
    <xs:complexType name="lokalizacjaType">
      <xs:complexType name="wspolrzedneType">
        <xs:sequence>
          <xs:element name="szerokosc">
            <xs:simpleType>
              <xs:restriction base="xs:decimal">
                <xs:minInclusive value="-90"/>
                <xs:maxInclusive value="90"/>
                <xs:fractionDigits value="4"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="dlugosc">
            <xs:simpleType>
              <xs:restriction base="xs:decimal">
                <xs:minInclusive value="-180"/>
                <xs:maxInclusive value="180"/>
                <xs:fractionDigits value="4"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="uklad"/>
      </xs:complexType>
    </xs:element>
  </xs:schema>
    
```

## 5 Rozdział IV – Walidacja

### 5.1 Ćwiczenia 6 – Błędy syntaktyczne.

Celem ćwiczenia jest poprawa dostarczonego przez trenera pliku XML zawierającym błędy składniowe. Dokument zawiera metadane dotyczące Wielkopolskiego Parku Narodowego.

Plik należy otworzyć w aplikacji Notepad++, a składnie sprawdzać za pomocą narzędzia XML-tools=>”Check XML syntax now”.

W poniższej tabeli należy udokumentować znalezione błędy, podając numer linii oraz powód wystąpienia błędu.

LINIA WYSTĄPIENIA BŁĘDU	POWÓD WYSTĄPIENIA BŁĘDU

Przykładowe rozwiązanie dotyczące ćwiczenia 6:

LINIA WYSTĄPIENIA BŁĘDU	POWÓD WYSTĄPIENIA BŁĘDU
19	<p>Niezgodność nazwy tagu otwierającego oraz zamykającego. Wielkość liter w nazwach elementów odgrywa znaczącą rolę.</p> <p>Jest: <code>&lt;/gmd:OrganisationName&gt;</code>,</p> <p>Powinno być: <code>&lt;/gmd:organisationName&gt;</code></p>
40-41	<p>Znaczniki zamykające element: <code>&lt;/gmd:CI_Contact&gt;</code> oraz <code>&lt;/gmd:address&gt;</code> zostały zamienione miejscami.</p>
78	<p>Atrybut „<code>xmlns:type</code>” w elemencie „<code>gmd:title</code>” powinien zostać zapisany w cudzysłowie.</p>
100	<p>Brak znaczniki zamykającego dla elementu <code>&lt;gco:CharacterString&gt;</code>, zamiast tego są dwa otwierające.</p>

## 5.2 Ćwiczenie 7 – Sprawdzenie zgodności pliku XML ze schematem XSD.

Podstawowym celem ćwiczenia jest zapoznanie uczestników ćwiczenia z techniką kontroli poprawności utworzonego pliku XML ze schematem XSD. Plik XML zawierać będzie kilka błędów, stąd drugi cel ćwiczenia to ich sukcesywna eliminacja aż do osiągnięcia zgodności ze schematem.

Plik XML należy otworzyć w aplikacji Notepad++, a następnie wybrać z menu kontekstowego

Wtyczki ->XML Tools -> Validate now, a następnie w otwartym oknie podać ścieżkę do pliku ze schematem XSD oraz zatwierdzić OK.

W poniższej tabeli należy udokumentować znalezione błędy, podając numer linii oraz powód wystąpienia błędu.

LINIA WYSTĄPIENIA BŁĘDU	POWÓD WYSTĄPIENIA BŁĘDU



Przykładowe rozwiązanie dotyczące ćwiczenia 7:

LINIA WYSTĄPIENIA BŁĘDU	POWÓD WYSTĄPIENIA BŁĘDU
2	Błąd nazwy elementu głównego.  Jest: zbiór  Powinno być: zbior
24	Wartość atrybutu <i>szerokosc</i> spoza zakresu. Jest: 251.2591  Powinno być: Wartość z przedziału -180 do +180
37-38	Niepoprawna kolejność elementów.  Jest: <i>długosc, szerokosc</i>  Powinno być: <i>szerokosc, długosc</i>

## 6 Rozdział V – XML i XML Schema w INSPIRE

### 6.1 Ćwiczenia 8 – Analiza fragmentu schematu XSD dla tematu SD

Celem ćwiczenia jest analiza fragmentu pliku XSD poświęconego klasie obiektów „Species DistributionDataSet” z tematu Rozmieszczenie Gatunków.

W oparciu o treść pliku zamieszczonego poniżej, proszę uzupełnić tabelę.

Tabela składa się z następujących pól:

- „Nazwa elementu”,
- „Typ danych”,
- „Wymagana krotność” -czy została sprecyzowana minimalna oraz maksymalna ilość wystąpień),
- „Numer linii” – w której linii pliku XSD zostały zamieszczone informacje dotyczące powyższych elementów.

Nazwa elementu	Typ danych	Wymagana krotność	Numer linii
inspireId			
beginLifeSpanVersion			
domainExtent			
endLifeSpanVersion			
name			
member			

Treść pliku:

```

1  <element name="SpeciesDistributionDataSet" substitutionGroup="gml:AbstractFeature" type=
2  "sd:SpeciesDistributionDataSetType">
3  <annotation>
4  <documentation>-- Name --
5  species distribution data set
6
7  -- Definition --
8  This data set is a collection of individual spatial objects (units) in a distribution of species.
9
10 -- Description --
11 Collection of individual spatial objects of allowed geometry where a species occurs.</documentation>
12 </annotation>
13 </element>
14 <complexType name="SpeciesDistributionDataSetType">
15 <complexContent>
16 <extension base="gml:AbstractFeatureType">
17 <sequence>
18 <element minOccurs="0" name="inspireId" type="base:IdentifierPropertyType">
19 <annotation>
20 <documentation>-- Name --
    inspire id
    
```

```

21
22 -- Definition --
23 External object identifier of the spatial object.
24
25 -- Description --
26 An external object identifier is a unique object identifier published by the responsible body, which may be used by
27 external applications to reference the spatial object. The identifier is an identifier of the spatial object, not an
28 identifier of the real-world phenomenon.</documentation>
29
30 </annotation>
31 </element>
32 <element name="name" nillable="true">
33 <annotation>
34 <documentation>-- Name --
35
36 name
37
38 -- Definition --
39 Name of a specific data set provided for Species Distribution.
40
41 -- Description --
42 A short descriptive identification name for a specific dataset provided by an institution.
43 EXAMPLE 1 Redlisted_species_Norway
44 EXAMPLE 2 Invasive_alien_species_Ireland
45 EXAMPLE 3 Articles17_report_Germany</documentation>
46
47 </annotation>
48 <complexType>
49 <simpleContent>
50 <extension base="string">
51 <attribute name="nilReason" type="gml:NilReasonType"/>
52 </extension>
53 </simpleContent>
54 </complexType>
55 </element>
56 <element name="domainExtent" nillable="true" type="gml:MultiSurfacePropertyType">
57 <annotation>
58 <documentation>-- Name --
59
60 domain extent
61
62 -- Definition --
63 The geographic extent of the domain of the feature collection.</documentation>
64
65 </annotation>
66 </element>
67 <element name="beginLifespanVersion" nillable="true">
68 <annotation>
69 <documentation>-- Name --
70
71 begin lifespan version
72
73 -- Definition --
74 Date and time at which this version of the spatial object was inserted or changed in the spatial data set.
75
76 -- Description --
77 NOTE This date is recorded to enable the generation of change only update files.</documentation>
78
79 </annotation>
80 <complexType>
81 <simpleContent>
82 <extension base="dateTime">
83
84 <attribute name="nilReason" type="gml:NilReasonType"/>
85 </extension>
86 </simpleContent>
87 </complexType>
88 </element>
89 <element minOccurs="0" name="endLifespanVersion" nillable="true">
90 <annotation>
91 <documentation>-- Name --
92
93 begin lifespan version
94

```

```

84 -- Definition --
85 Date and time at which this version of the spatial object was superseded or retired in the spatial data set.
86
87 -- Description --
88 NOTE This date is recorded to enable the generation of change only update files.</documentation>
89     </annotation>
90     <complexType>
91     <simpleContent>
92         <extension base="dateTime">
93             <attribute name="nilReason" type="gml:NilReasonType"/>
94         </extension>
95     </simpleContent>
96 </complexType>
97 </element>
98 <element maxOccurs="unbounded" minOccurs="0" name="documentBasis" nillable="true">
99     <annotation>
100         <documentation>-- Name --
101     </annotation>
102     documentBasis
103
104 -- Definition --
105 Reference to or citation of a document describing a campaign or a legal act which is the basis for the data set.
106 </documentation>
107     </annotation>
108     <complexType>
109     <sequence minOccurs="0">
110         <element ref="base2:DocumentCitation"/>
111     </sequence>
112     <attributeGroup ref="gml:AssociationAttributeGroup"/>
113     <attributeGroup ref="gml:OwnershipAttributeGroup"/>
114 </complexType>
115 </element>
116 <element maxOccurs="unbounded" name="member" type="sd:SpeciesDistributionUnitPropertyType">
117     <annotation>
118         <documentation>-- Name --
119     </annotation>
120     member
121
122 -- Definition --
123 Individual spatial objects in a collection of spatial objects.
124
125 -- Description --
126 Individual spatial objects of allowed geometry where a species occurs.</documentation>
127     </annotation>
128     </element>
129 </sequence>
130 </extension>
131 </complexContent>
132 </complexType>
    
```

Rozwiązanie ćwiczenia 8:

Nazwa elementu	Typ danych	Wymagana krotność	Numer linii
inspireId	base:IdentifierPropertyType	MIN = 0, MAX = 1	17
beginLifeSpanVersion	dateTime	MIN = 1, MAX = 1	60; 73
domainExtent	gml:MultiSurfacePropertyType	MIN = 1, MAX = 1	51
endLifeSpanVersion	dateTime	MIN = 0, MAX = 1	79; 92
name	STRING	MIN = 1, MAX = 1	29; 45
member	sd:SpeciesDistributionUnitPropertyType	MIN = 1, MAX = unbounded	114