

Formaty danych – cz. 2

UWAGA: Te ćwiczenia równoważne są dwóm spotkaniom, wliczając to, które nas ominęło. Termin oddania jest za to o tydzień późniejszy.

Dzisiejsze zajęcia kontynuują zapoznanie się z formatem XML.

Ostatnio poznaliśmy podstawową strukturę XML - jej drzewiasty charakter oraz definicję znaczników.

Atrybuty:

Dziś rozszerzymy to pojęcie poprzez atrybuty. Atrybuty definiowane są w znacznikach, np.

```
<osoba plec="kobieta">
```

lub

```
<osoba plec='kobieta'>
```

Zatem dane osobowe zapisać możemy:

```
<osoba plec="kobieta">  
  <imie> Anna </imie>  
  <nazwisko> Sobowska </nazwisko>  
</osoba>
```

Zwróćcie uwagę, że zapisać możemy to też jako:

```
<osoba>  
  <plec>kobieta</plec>  
  <imie> Anna </imie>  
  <nazwisko> Sobowska </nazwisko>  
</osoba>
```

To my decydujemy (albo ja w treści zadania), jaką formę wybierzemy.

Atrybutów możemy zdefiniować więcej niż jeden:

```
<notka data = "2018-03-21" od = "Kamil">
```

Przestrzenie nazw:

Zbadaj poniższą strukturę:

```
<tabela>  
  <tabela>  
    <wiersz> 120 </wiersz>  
    <wiersz> 123 </wiersz>  
  </tabela>
```

```

    <tabela>
      <kolumna>343</kolumna>
    </tabela>
  <tabele>

```

Wyobraź sobie jak zapisać drzewo takiej struktury.

Na poziomie drugim musielibyśmy zdefiniować dwa węzły <tabela>, prawda (bo jeden miałby podwęzeł wiersz, a drugi kolumna).

Jest to więc konflikt. Takie problemy można rozwiązywać używając przestrzeni nazw. Przykładowo:

```

<tabele>
  <w:tabela>
    <w:wiersz> 120 </wiersz>
    <w:wiersz> 123 </wiersz>
  </w:tabela>
  <k:tabela>
    <k:kolumna>343</kolumna>
  </k:tabela>
</tabele>

```

Walidacja i schematy XSD

Schematy służą do odgórnego narzucenia schematu w jakim ma być napisany plik XML. Oznacza to, że możemy zdefiniować strukturę XML w jakiej mają być produkowane pliki. Jest to przydatne np. podczas odbierania i przetwarzania danych, które mają mieć określoną strukturę. Zanim dokonamy na nich operacji, wykorzystujemy schemat XSD do walidacji struktury otrzymanego pliku XML.

W praktyce, GML - format danych przestrzennych, posiada odgórnie zdefiniowany schemat XSD, dzięki któremu każdy plik GML posiada ustaloną strukturę, którą z kolei systemy GIS są w stanie odczytać i zwizualizować.

Jak definiować schematy XSD

Nagłówek:

```
<?xml version="1.0"?>
```

Który pozostaje zawsze ten sam.

Znaczniki dzielimy na proste i złożone.

Znaczniki proste definiujemy w ten sposób:

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

Oznacza to że plik ma zawierać element o nazwie "do", a jego wartość musi być tekstem (string).

Dopuszczalne wartości to:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date (format YYYY-MM-DD)
- xs:time

Możemy również dodać informację o domyślnej wartości:

```
<xs:element name="kolor" type="xs:string" default="czerwony"/>
```

lub wartość stałą:

```
<xs:element name="kolor" type="xs:string" fixed="czerwony"/>
```

Znaczniki złożone. Tu sprawa jest bardziej złożona. Weźmy na przykład notatkę:

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

```
<notatka>
```

```
    <do>Rygobert</do>
```

```
    <od>Saturnin</od>
```

```
    <naglowek>Wiadomość</naglowek>
```

```
    <tekst>Cześć!</tekst>
```

```
</notatka>
```

Jej schemat będzie następujący:

```
<?xml version="1.0"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

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

```
  <xs:complexType>
```

```
    <xs:sequence>
```

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

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

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

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

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

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

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

```
</xs:schema>
```

Zwróćmy uwagę na dwie rzeczy:

- Główny korzeń <xs:schema> z atrybutem
xmlns:xs="<http://www.w3.org/2001/XMLSchema>"

- Definicja elementu złożonego <xs:complexType>, który zawiera w sobie <xs:sequence>, czyli sekwencję znaczników, które są elementami prostymi o typie string

Atrybuty:

Można również zdefiniować wymóg obecności pewnego atrybutu:

```
<xs:attribute name="nazwa" type="typ"/>
```

Typy są takie same jak w znacznikach prostych.

Ograniczenia:

Możemy również nałożyć ograniczenia na wartości. Na przykład

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

Powyższe ograniczenie wymaga dopisania do znacznika prostego "wiek", znacznika definiującego typ prosty <xs:simpleType> oraz znacznika zawierającego ograniczenia <xs:restriction> o atrybucie mówiącym jakiego typu wartości dotyczy to ograniczenie.

Inny przykład:

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

Można go też zapisać jako:

```
<xs:element name="samochod" type="samochodTyp"/>

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

Jest to przeniesienie definicji złożonych elementów (tak, działa to też z definicją znaczników złożonych) poza strukturę drzewa. Upraszcza to wizualnie zapis.

Dopuszczalne ograniczenia:

- xs:enumeration - dopuszczalne wartości (patrzy powyższy przykład),
- xs:fractionDigits - liczba miejsc po przecinku w liczbie, która jest dopuszczalna
- xs:length - ograniczna długość wartości (w znakach)
- xs:maxInclusive - maksymalna wartość (mniejsza lub równa)
- xs:maxExclusive - maksymalna wartość (mniejsza)
- xs:minInclusive
- xs:minExclusive
- xs:minLength - minimalna długość

Więcej o definicji złożonych elementów:

Pierwszy przykład, podobnie jak w restrykcjach możemy w celu uproszczenia zdefiniować jak poniżej:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
<xs:element name="notatka" type="notatkaTyp"/>

<xs:complexType name="notatkaTyp">
  <xs:sequence>
    <xs:element name="do" type="xs:string"/>
    <xs:element name="od" type="xs:string"/>
    <xs:element name="naglowek" type="xs:string"/>
    <xs:element name="tekst" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

</xs:schema>
```

Ten sam typ danej można wtedy przypisać różnym znacznikom!

Znacznik <xs:sequence> oznacza, że znaczniki muszą występować w podanej w definicji kolejności.

Można również stosować inne znaczniki:

- <xs:all> - muszą występować wszystkie w jakiejkolwiek kolejności
- <xs:choice> - jeden z podanych musi wystąpić

Można również zdefiniować ile znaczników może wystąpić. Na przykład definicja:

```
<xs:element name="osoba">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="imie" type="xs:string"/>
      <xs:element name="imie_dziecka" type="xs:string" maxOccurs="10"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Oznacza, że znacznik osoba, może mieć maksymalnie 10 znaczników imie dziecka. Podobnie zdefiniować można parametr minimalnej liczby wystąpień "minOccurs" (który może być równy 0, to znaczy stanie się wtedy parametrem opcjonalnym).

1. Zgodnie z najnowszą wiedzą popraw zadanie 3 z poprzednich zajęć,
2. Napisz strukturę księgarni, która zawiera:
 - a. Główny korzeń księgarnia, a w nim znacznik "ksiazka" o atrybucie "kategoria" oraz znacznikach podrzędnych:
 - i. tytuł (z atrybutem język tytułu)
 - ii. autor
 - iii. rok
 - iv. cena
 - b. Stwórz 4 książki, niech jedna ma więcej niż jednego autora.
3. Napisz schemat do struktury z zadania nr 2. Zwaliduj je ze sobą za pomocą tej strony: <https://www.xmlvalidation.com/>
4. Zadanie **projekt grupowy**. Proszę dobrać się w pary. Każda osoba napisać ma jeden schemat XSD zawierający minimum:
 - a. 4 znaczniki złożone, wszystkie poprzez przeniesienie definicji typu poza strukturę drzewa.
 - b. 10 znaczniki proste,
 - c. 4 atrybuty,
 - d. minimum 5 ograniczeń,
 - e. minimum 2 ograniczenia liczby wystąpień znacznika
 - f. każdy z ze znaczników <xs:sequence>, <xs:all>, <xs:choice>
 - g. minimum 2 znaczniki opcjonalne

Następnie należy z osobą z grupy wymienić się schematami. Do otrzymanego schematu należy napisać odpowiedni plik XML, który zwaliduje się poprawnie z otrzymanym schematem XSD. Walidację należy przeprowadzić za pomocą strony: <https://www.xmlvalidation.com/>

Zadania proszę wysłać grupami (zadania 1,2,3 osobno, 4 razem) do 07.04.2020. Te ćwiczenia równoważne są dwóm spotkaniom.