

XML (Extensible Markup Language)

Początek laboratorium:

- na pulpicie utworzyć folder `lab1`,
- otworzyć go w *Visual Studio Code* lub dowolnym IDE od *JetBrains* (np. *PHPStorm*, *IntelliJ*, *PyCharm* itp.),
- w przypadku wyboru *VSCode* zainstalować w nim rozszerzenie „XML”:

<https://marketplace.visualstudio.com/items?itemName=redhat.vscode-xml>

Zadania (XML):

- specyfikacja XML 1.0: <https://www.w3.org/TR/xml/>
- przykładowy plik XML: <https://www.w3schools.com/xml/simple.xml>
- specyfikacja XML Schema 1.1: <https://www.w3.org/XML/Schema.html#dev>
- poprawność dokumentu (składniowa i strukturalna):

<https://www.mimuw.edu.pl/~czarnik/zajecia/xml03/lab2.html>

<http://edu.pjwstk.edu.pl/wyklady/tin/scb/wyklad07/index.html#typ>

- DTD – nie będzie występował na laboratorium,
- przykładowy plik XML Schema „*shiporder.xsd*” (w połowie strony):

https://www.w3schools.com/xml/schema_example.asp

- przykłady zastosowania XML Schema jako struktury dla dokumentów XML:

<https://www.podatki.gov.pl/e-deklaracje/dokumentacja-it/struktury-dokumentow-xml/>

Zadanie 1.1:

Zapoznać się z następującymi zagadnieniami dotyczącymi języka XML:

- deklaracja XML,
- znaczniki,
- elementy,
- element bazowy,
- atrybuty,
- dozwolone nazwy atrybutów, znaczników...,
- przestrzenie nazw,
- sekcje CDATA.

<http://www.cs.put.poznan.pl/jkobusinski/xml.html>

<https://www.w3schools.com/xml/>

Zadanie 1.2:

Utworzyć przykładowy plik XML o nazwie `rozkład.xml` odnośnie *rozkładu jazdy pociągów*, według wymagań:

Dla rozkładu jazdy pociągów powinna być dostępna informacja o okresie obowiązywania.

Dla każdego pociągu w rozkładzie jazdy powinna być dostępna informacja o:

- numerze pociągu,
- nazwie przewoźnika (maks. 15 znaków),
- rodzaju pociągu (osobowy, pospieszny, ekspres),
- opcjonalnej nazwie pociągu,
- nazwach stacji/przystanków, na których zatrzymuje się pociąg wraz z godzinami przyjazdu i odjazdu,
- zestawieniu 1-5 wagonów (wagony klasy 2, wagony klasy 1, wagony sypialne, wagony z miejscami do leżenia, wagon restauracyjny).

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

<!-- zawartość pliku -->

</rozkład>
```

Zadanie 1.3:

Otworzyć plik `rozkład.xml` w przeglądarce internetowej w celu sprawdzenia czy plik się prawidłowo *parsuje*, więc jest poprawny składniowo (*well-formed*).

https://www.w3schools.com/xml/xml_syntax.asp
<https://www.w3resource.com/xml/well-formed.php>

Zadanie 1.4:

Zapoznać się z podstawowymi pojęciami dotyczącymi XML Schema:

- XML Schema,
 - XML Schema Definition (XSD),
 - XSD Data types,
 - XSD Elements (*element, all, sequence, complexType, simpleType, attribute*),
 - XSD Restrictions,
- oraz wzorcami budowania XML:
- deklarowanie *globalne* vs *lokalne*,
 - wzorzec *Russian Doll*,
 - wzorzec *Salami Slice*,
 - wzorzec *Venetian Blind*,
 - wzorzec *Garden of Eden*.

https://www.w3schools.com/xml/schema_simple.asp
https://www.w3schools.com/xml/schema_elements_ref.asp
https://www.w3schools.com/xml/schema_facets.asp
https://www.w3schools.com/xml/schema_complex_indicators.asp

<http://www.java2s.com/Tutorial/XML/CatalogXML.htm>

<https://www.xfront.com/GlobalVersusLocal.html>
<https://www.oracle.com/technical-resources/articles/java/design-patterns.html>

Zadanie 1.5:

Zaprojektować *definicję struktury* dla plików XML mających przechowywać informacje o rozkładzie jazdy pociągów:

- ustalić wymagania/zasady jak w zadaniu 1.2, ustalić typy danych,
- zdefiniować XML Schema w nowym pliku rz1.xsd (skopiować do niego poniższy fragment),
- napisać *scemę* według wzorca *Russian Doll*.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="unqualified">

<!-- zawartość pliku -->

</xs:schema>
```

Zadanie 1.6:

W pliku rozkład.xml umieścić referencję do pliku rz1.xsd. Następnie sprawdzać poprawność *strukturalną*.

```
<rozklad
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="rz1.xsd">
```

Przykład:

```
<?xml version="1.0" encoding="UTF-8"?>
<rozklad
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="rz1.xsd">
  <od>2023-10-01</od>
  <do>2023-13-31</do>
  <poc>
    Content of type 'date' is expected.
  </poc>
  The following content is not a valid type:
  '2023-13-31'
```

Zadanie 1.7:

Utworzyć plik rz2.xsd w którym napisać definicję struktury jak w zadaniu 1.5, ale tym razem napisać *scemę* według wzorca *Venetian Blind*.

–

Zadanie 1.8:

W pliku rozkład.xml zastąpić obecną referencję do pliku rz1.xsd na do pliku rz2.xsd. Następnie sprawdzać poprawność *strukturalną*.

–

✳ – zadania/podpunkty do samodzielnego dokończenia/wykonania,

✴ – zadania/podpunkty dla zainteresowanych.

Po zakończonym laboratorium należy skasować wszystkie pobrane oraz utworzone przez siebie pliki z komputera w sali laboratoryjnej.

Inne: *

<https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

<https://maven.apache.org/xsd/maven-4.0.0.xsd>

<https://bykowski.pl/hibernate-cfg-xml/>

<http://hibernate.org/dtd/hibernate-configuration-3.0.dtd>