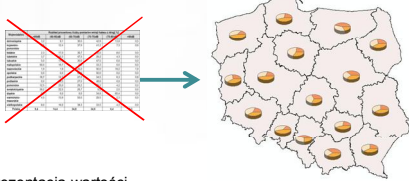


Szkolenia INSPIRE dla administracji

Alternatywna forma przedstawienia danych liczbowych – mapy tematyczne



Prezentacja wartości bezwzględnych w latach

Kartodiagram kołowy strukturalny

---

---

---

---

---

---

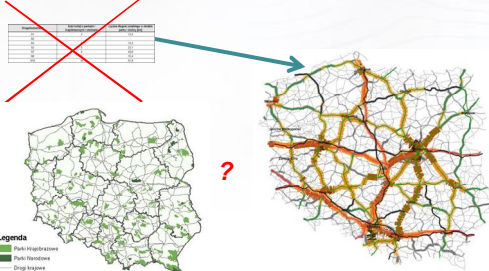
---

---

---

---

Szkolenia INSPIRE dla administracji



Legenda

- Park-Rehabilitacja
- Park-Nowoczesny
- Stary Inne
- Obszar administracyjny

Metoda zasięgu, sygnaturowa liniowa

Kartodiagram wstęgowy

---

---

---

---

---

---

---

---

---

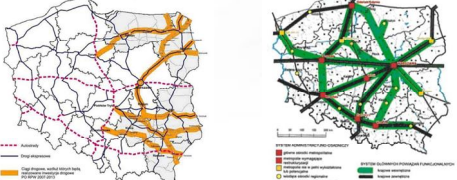
---

Szkolenia INSPIRE dla administracji

Przykłady opracowań mapowych już występujących w raporcie

Ciągi drogowe wyznaczone w PO Rozwój Polski Wschodniej na tle

System osadniczo-administracyjny i główne powiązania funkcjonalne



Sygnatury liniowe

Sygnatury liniowe, punktowe

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**■ Przykłady opracowań mapowych już występujących w raporcie**

Schemat przekształceń struktury przestrzennej Polski w pierwszych latach XXI wieku

Rozmieszczenie złóż kruszyw naturalnych na tle infrastruktury dróg szybkiego ruchu i autostrad w Polsce

Metoda zasięgów planowych, sygnatury punktowe

Sygnatury liniowe, punktowe

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**■ Mapa będąca wynikiem przeprowadzonych analiz danych przestrzennych coraz częściej widziana jest głównie na monitorach, w Internecie**

Stanowi podstawę określenia przestrzennego zróżnicowania prezentowanego zjawiska:

- kształtu, wielkości i wzajemnego położenia obiektów,
- określenia podstawowych cech ilościowych i jakościowych zjawiska,
- określenia relacji przestrzennych występujących między obiektami

Kluczowe znaczenie ma dobór odpowiedniej metody prezentacji kartograficznej do prezentowanego zjawiska

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**■ Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:**

Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA” Spółka z o.o. 82-300 Elbląg, ul. Tysiąclecia 11 [www.opegieka.pl](http://www.opegieka.pl)

Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27 [www.igjk.edu.pl](http://www.igjk.edu.pl)

Centrum UNEP/GRID-Warszawa, 00-764 Warszawa, ul. Sobieszyńska 8 [www.gridw.pl](http://www.gridw.pl)

Podwykonawca – partner technologiczny:

Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52 [www.intergraph.pl](http://www.intergraph.pl)

---

---

---

---

---

---

---

---

---

---

Szkozenia INSPIRE dla administracji

Szkolenie eksperckie

## Środki formalne modelowania informacji geograficznej – UML, XML, GML

wykładowca:  
IMIĘ NAZWISKO

Zajęcia przygotowane na podstawie autorskiego programu opracowanego przez:  
dr inż. Agnieszkę Chojkę z Uniwersytetu Warmińsko-Mazurskiego w Olsztynie

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## 5. Środki formalne modelowania informacji geograficznej – UML, XML, GML

---

---

---

---

---

Szkozenia INSPIRE dla administracji

Plan szkolenia

1. Wprowadzenie
2. UML
  - a. Podstawy języka UML
  - b. Reguły budowy schematów aplikacyjnych w UML
3. XML
  - a. Składania języka XML i XML Schema
  - b. Reguły budowy schematów aplikacyjnych w XML Schema
4. GML
  - a. Schemat aplikacyjny GML
  - b. Reguły kodowania UML-GML

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

Szkozenia INSPIRE dla administracji

Środki formalne modelowania informacji geograficznej – UML, XML, GML

# WPROWADZENIE

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

#### ■ Infrastruktura danych przestrzennych

- **Perspektywa danych** (ang. *data-centric view*):
  - schematy aplikacyjne,
  - metadane.
- **Perspektywa usług** (ang. *service-centric view*):
  - interoperacyjność,
  - architektura zorientowana na usługi.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

#### ■ Perspektywa danych

- **Podejście oparte na modelu** (ang. *Model Driven Approach*):
  - **niezależny od implementacji schemat aplikacyjny** zostaje odwzorowany na **różne specyfikacje** (wykorzystujące różne technologie, np. usługi sieciowe, relacyjne bazy danych, XML),
  - **specyfikacje** mogą zostać zaimplementowane (wdrożone) na **różnych platformach** sprzętowo-programowych.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

#### ■ Podejście oparte na modelu

niezależny od implementacji opis semantyki i logiczna struktura danych

specyfikacje dla różnych technik

implementacje

Podejście oparte na modelu  
Źródło: prCEN/TR 15449, 2006.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

■ **Podejście oparte na modelu**

- **Koncepcja MDA**  
(ang. *Model Driven Architecture*):
  - opracowana przez OMG (ang. *Object Management Group*),
  - **Cel**: rozwiązywanie problemów związanych z integracją systemów informatycznych pochodzących od różnych dostawców oraz działających na różnych platformach informatycznych (wykorzystujących różne technologie, np. różne systemy operacyjne, różne standardy sieciowe, różne języki programowania).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

■ **MDA – modele systemu**

- **CIM** (ang. *Computation Independent Model*)
  - **specyfikacja wymagań**,
- **PIM** (ang. *Platform Independent Model*)
  - **model pojęciowy** (niezależny od platformy),
- **PSM** (ang. *Platform Specific Model*)
  - **model logiczny** (zależny od wybranej platformy),
- **Implementacja** (ang. *Implementation Model*)
  - **model fizyczny** (np. program, struktura bazy danych).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

■ **Technologia MDA**

- Kluczową rolę odgrywa modelowanie systemu w języku **UML** (ang. *Unified Modeling Language*).
- Zalecany przez **normy ISO serii 19100** język schematu pojęciowego (ang. *conceptual schema language*).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### ■ Interoperacyjność

- **Zdolność do współdziałania** – m.in. zdolność do komunikowania, a co za tym idzie dokonywania transferu danych między systemami.
- **Cel norm ISO serii 19100!!!**

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### ■ Interoperacyjność różnych systemów

- Semantyka zawartości i struktur logicznych danych przestrzennych – tzw. **schemat aplikacyjny** (ang. *application schema*).
- Niezależna od platformy sprzętowo-programowej struktura danych, która może reprezentować dane zgodnie ze schematem aplikacyjnym.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### ■ Terminologia

- **Model pojęciowy** (ang. *conceptual model*)
  - model definiujący pojęcia z pewnej przestrzeni rozważań (przedmiotu zainteresowań).
- **Schemat pojęciowy** (ang. *conceptual schema*)
  - formalny opis modelu pojęciowego w określonym języku schematu pojęciowego.
- **Schemat aplikacyjny** (ang. *application schema*)
  - schemat pojęciowy dla danych wykorzystywanych przez jedną lub więcej aplikacji, dla określonego zakresu przedmiotowego.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Schemat aplikacyjny**

- Podstawa pomyślnej wymiany danych:
  - definiuje możliwą zawartość oraz strukturę danych.
- Do wymiany danych:
  - zapisany w języku schematu pojęciowego UML, zgodnie z normami ISO/TS 19103 i ISO 19109 (reguły budowy schematów aplikacyjnych).
- Dostępny dla obu uczestników procesu wymiany danych (nadawcy i odbiorcy):
  - aby zapewnić pomyślny przebieg wymiany danych.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Interoperacyjna wymiana danych**

Ogólna idea wymiany danych między dwoma systemami  
Źródło: prCEN/TR 15449, 2006; ISO/IS 19118:2005

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Interoperacyjna wymiana danych**

- System A chce przesłać zbiór danych do systemu B. W tym celu A i B muszą:
  - uzgodnić wspólny schemat aplikacyjny  $I$ ,
  - zdecydować, jaką zastosować regułę kodowania  $R$ ,
  - jaki wykorzystać rodzaj protokołu transferu.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Interoperacyjna wymiana danych

- Oba systemy, A i B, przechowują dane w wewnętrznych bazach danych zgodnie ze swoimi wewnętrznymi schematami, ale zazwyczaj te schematy są różne.
- Przeniesienie zbioru danych z wewnętrznej bazy danych systemu A do wewnętrznej bazy danych systemu B, musi przebiegać w następujący sposób:

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Interoperacyjna wymiana danych

1. System A tłumaczy swoje wewnętrzne dane na strukturę danych zgodną ze wspólnym schematem aplikacyjnym / – powstaje schemat aplikacyjny określonej struktury danych iA.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Interoperacyjna wymiana danych

2. Zostaje wykorzystana usługa kodowania, która stosuje regułę kodowania  $R$  do stworzenia struktury danych  $d$  niezależnej od systemu, a więc odpowiedniej do transferu.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

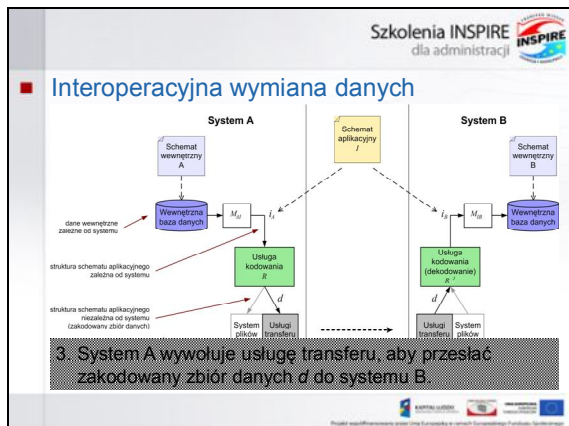
---

---

---

---






---

---

---

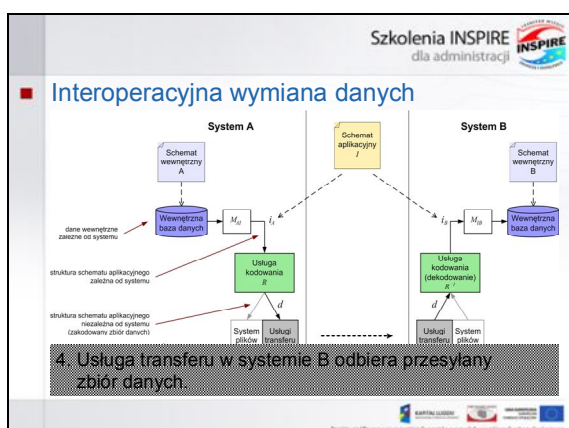
---

---

---

---

---




---

---

---

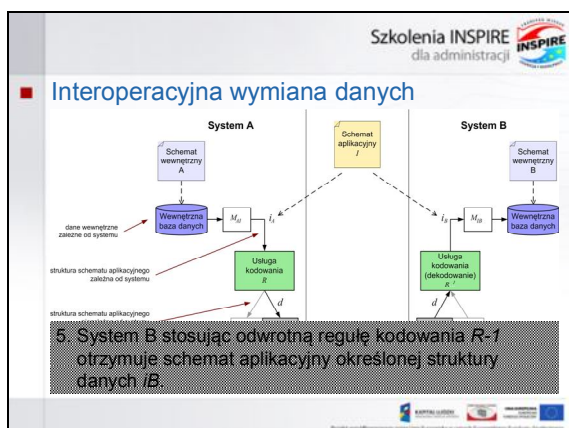
---

---

---

---

---




---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Interoperacyjna wymiana danych

6. System B tłumaczy schemat aplikacyjny określonej struktury danych *iB* na swoją wewnętrzną bazę danych.

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Reguły kodowania

- Normy ISO serii 19100 określają dwie reguły kodowania oparte na języku XML (ang. *Extensible Markup Language*):
  - ISO 19136 (GML), Załącznik E
    - reguła kodowania oparta na XML dla schematów aplikacyjnych zgodnych z normą ISO 19109, które mogą być przedstawione za pomocą ograniczonego profilu UML, pozwalającego na konwersję do języka XML Schema,
  - ISO/TS 19139
    - reguła kodowania oparta na XML dla schematów pojęciowych określających typy, które opisują zasoby geograficzne, np. metadane według ISO 19115 i katalogi obiektów według ISO 19110.

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Reguła kodowania

- Powinna określać:
  - reguły konwersji schematu,
  - reguły konwersji instancji.

Reguła kodowania oparta na XML  
Źródło: ISO/DIS 19118:2005

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

### Reguły konwersji

- Oparte na języku XML dla neutralnej wymiany danych zakładają, że:
  - definicje klas w schemacie aplikacyjnym są odwzorowywane na deklaracje typu w schemacie XML (XML Schema),
  - obiekty w modelu instancji są odwzorowywane na odpowiadające im struktury elementu w dokumencie XML.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

### Materiały uzupełniające

- MDA Guide Version 1.0, 2003. OMG. <http://www.omg.org/mda>
- prCENTR 15449, 2006. *Geographic information – Standards, specifications, technical reports and guidelines, required to implement Spatial Data Infrastructure.*
- ISO/TS 19103:2005 *Geographic information – Conceptual schema language*
- ISO 19109:2009 *Geographic information – Rules for application schema*
- ISO/DIS 19118:2005 *Geographic information – Encoding*
- ISO 19136:2007 *Geographic information – Geography Markup Language (GML)*
- ISO/TS 19139:2007 *Geographic information – Metadata – XML schema implementation*

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---


---

---




---

---


---

Szkolenia INSPIRE dla administracji 

**Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:**

-  Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA” Spółka z o.o. 82-300 Elbląg, ul. Tysiąclecia 11 [www.opegieka.pl](http://www.opegieka.pl)
-  Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27 [www.igik.edu.pl](http://www.igik.edu.pl)
-  Centrum UNEP/GRID-Warszawa, 00-764 Warszawa, ul. Sobieszyńska 8 [www.gridw.pl](http://www.gridw.pl)

Podwykonawca – partner technologiczny:

-  Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52 [www.intergraph.pl](http://www.intergraph.pl)

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

Szkolenie eksperckie

## Środki formalne modelowania informacji geograficznej – UML, XML, GML

wykładowca:  
IMIĘ NAZWISKO

Zajęcia przygotowane na podstawie autorskiego programu opracowanego przez:  
dr inż. Agnieszkę Chojkę z Uniwersytetu Warmińsko-Mazurskiego w Olsztynie

## 6. Środki formalne modelowania informacji geograficznej – UML, XML, GML

---



---



---



---

Szkolenia INSPIRE dla administracji

Środki formalne modelowania informacji geograficznej – UML, XML, GML

# UML

## Reguły budowy schematów aplikacyjnych w UML

---



---



---



---



---

Szkolenia INSPIRE dla administracji

Schemat aplikacyjny

- Powinien być zapisany w języku schematu pojęciowego UML według zasad określonych w normach ISO/TS 19103 i ISO 19109.
- Składa się z pojęć określonych przez aplikację wyrażonych jako klasy i powiązania.
- Niektóre z klas mogą być zaimportowane ze schematów znormalizowanych z innych standardów.

---



---



---



---



---




---

Szkolenia INSPIRE dla administracji

## ISO 19109

- Ogólne reguły budowy i dokumentowania schematów aplikacyjnych:
  - zasady modelowania pojęciowego obiektów oraz ich właściwości,
  - reguły definiowania schematu aplikacyjnego za pomocą języka schematu pojęciowego,
  - wyrażanie pojęć z modelu pojęciowego w postaci typów danych w schemacie aplikacyjnym,
  - zasady integracji schematu aplikacyjnego ze znormalizowanymi schematami pojęciowymi informacji geograficznej.



---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

## Istota ISO 19109

- Definicja obiektu geograficznego (ang. *geographic feature*), który reprezentuje dowolny element ze świata rzeczywistego, np. budynek, drzewo, obraz satelitarny.
- Integracja obiektu z modelem informacji geograficznej w jednorodny (homogeniczny) sposób:
  - General Feature Model (GFM)* – abstrakcyjny obiekt z atrybutami (właściwościami obiektu) i operacjami (zachowanie obiektu).



---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

## ISO/TS 19103

- Profil UML w dziedzinie informacji geograficznej (geoinformatyki/geomatyki) dostosowany do norm ISO serii 19100 (przyjęta konwencja nazywania i modelowania pozostaje niezmienna dla całej serii norm).
  - m.in. zasady definiowania klas, atrybutów, typów danych, operacji, związków i stereotypów.
  - modele normatywne wykorzystują diagramy klas i diagramy pakietów. Inne diagramy UML mogą być stosowane informacyjnie.



---

---

---

---

---

---

---

---

---

---



Szkolenia INSPIRE dla administracji

### UML

- ang. *Unified Modeling Language*
- pol. *zunifikowany (ujednolicony) język modelowania*
- Język formalny służący do opisu świata obiektów w analizie obiektowej oraz programowaniu obiektowym.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Diagram klas w UML

- Opis:
  - zbioru danych,
  - zależności pomiędzy nimi.
- Podstawowe elementy:
  - oznaczenia klas,
  - związków pomiędzy nimi,
  - oznaczenia pomocnicze (np. stereotypy, metki, ograniczenia itp.).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Diagram klas – przykład

Przykład diagramu klas w UML

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

## ■ Diagram pakietów w UML

- Zawiera m.in. klasy i diagramy klas.
- Porządkuje strukturę zależności w systemie, w który można wyróżnić bardzo wiele klas, przypadków użycia itp.
- Przedstawia podział systemu z logicznego punktu widzenia.

Przykład współfinansowanego projektu Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

## ■ Diagram pakietów – przykład

Przykład diagramu klas w UML

Przykład współfinansowanego projektu Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

## ■ Klasa

- Opis zbioru obiektów, które mają takie same atrybuty, operacje, związki i znaczenie (wzorzec dla tworzonego obiektu).
- Według ISO/TS 19103 każda klasa powinna być zdefiniowana wewnątrz określonego pakietu.
- **Symbol graficzny:**
  - Prostokąt podzielony na trzy pola:

pole nazwy
pole atrybutów
pole operacji

    - **pole nazwy** – nazwa klasy, stereotyp, lista wartości etykietowanych,
    - **pole atrybutów** – lista atrybutów klasy,
    - **pole operacji** – lista operacji (metod) zdefiniowanych dla klasy.

Klasa w UML

Przykład współfinansowanego projektu Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Nazwa

- Każda klasa musi mieć przypisaną nazwę, która wyróżnia ją spośród innych klas.
- Napis, pisany pogrubioną czcionką.
- Na ogół w formie krótkiego rzeczownika lub wyrażenia rzeczownikowego (mianownik, l. poj.), pochodzącego ze słownictwa modelowanego systemu.
- Każdy wyraz w nazwie zaczyna się wielką literą.
- Jeżeli nazwa klasy jest wieloczłonowa, każdy kolejny człon zaczyna się wielką literą, a człony są ze sobą „sklejone” (brak spacji, podkreślenia), np. Budynek, PunktAdresowy.

Przykład współfinansowania projektu Europejski Fundusz Społeczny

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Nazwa

- Zgodnie z ISO/TS 19103, nazwa klasy powinna być poprzedzona przedrostkiem składającym się z dwóch wielkich liter, który identyfikuje pakiet lub normę, z której taka klasa pochodzi, np. **GM** – geometria, **TP** – topologia.

AD	Abstract address	ISO 19133	LI	Lineage	ISO 19115
CC	Changing coordinates	ISO 19111	LR	Linear reference	ISO 19133
CI	Citation	ISO 19115	MD	Metadata	ISO 19115
CV	Coverages	ISO 19123	NT	Network position	ISO 19133
DQ	Data quality	ISO 19115	PF	Feature Portrayal	ISO 19117
DS	Dataset	ISO 19115	PS	Positioning Services	ISO 19116
EX	Externet	ISO 19115	RS	Reference System	ISO 19111, ISO 19115
FC	Feature Catalogue	ISO 19110	SC	Spatial coordinates	ISO 19111
FD	Feature ID	ISO 19133	SD	Sensor and Data Models	ISO 19130
FE	Feature	ISO 19109	SI	Spatial Identification	ISO 19112
FI	Feature topology	ISO 19107	SV	Services	ISO 19119
GF	Geobneral feature model	ISO 19109	TM	Temporal	ISO 19108
GM	Geometry model	ISO 19107	TT	Topology	ISO 19107
GR	Graph	ISO 19107	TS	Simple Topology	ISO 19107

Przykłady skrótów stosowanych w nazwach klas

Przykład współfinansowania projektu Europejski Fundusz Społeczny

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Atrybut

- Nazwana właściwość klasy.
- Klasa może mieć dowolną liczbę atrybutów albo nie mieć ich wcale.
- Nazwa atrybutu może być dowolnym tekstem (jak nazwa klasy), na ogół w formie rzeczownika lub wyrażenia opisującego właściwość danej klasy (mianownik, l. poj.).
  - Każdy wyraz w nazwie (poza pierwszym) zaczyna się wielką literą, kolejne wyrazy są ze sobą „sklejone”, np. nazwisko, dataUrodzenia.
- Musi mieć przypisany typ danych, wybrany spośród obowiązujących typów podstawowych.

Przykład współfinansowania projektu Europejski Fundusz Społeczny

---

---

---

---

---

---

---

---

---

---

---

---



Szkolenia INSPIRE dla administracji

## Operacja

- Abstrakcja czegoś, co można zrobić z każdym obiektem danej klasy.
- Klasa może mieć dowolną liczbę operacji albo nie mieć ich wcale.
- Nazwa operacji może być dowolnym tekstem, na ogół w formie czasownika lub wyrażenia opisującego pewne zachowanie danej klasy.
  - Każdy wyraz w nazwie (poza pierwszym) zaczyna się wielką literą, kolejne wyrazy są ze sobą „sklejone”, np. usuń, zatwierdźOperację.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

## Widoczność atrybutów i/lub operacji

- Dodatkowo określana na diagramach UML:
  - + oznacza, że atrybut (operacja) jest publiczny (ang. *public*), czyli dostępny spoza klasy,
  - oznacza, że atrybut (operacja) jest prywatny (ang. *private*), czyli niedostępny spoza swojej klasy,
  - # oznacza, że atrybut (operacja) jest chroniony (ang. *protected*), czyli dostępny dla klas pochodnych (dziedziczących) od danej klasy.
- ! oznacza, że wartość danego atrybutu jest wyliczana (ang. *derived*) na podstawie wartości innych atrybutów.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

## Typy danych

- ISO/TS 19103 grupuje podstawowe typy danych w 3 kategorie:
  - Elementarne** (ang. *primitives types*),
  - Implementacyjne** (ang. *implementation types*),
  - Pochodne** (ang. *derived types*).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Związki**

- **Zależność** (ang. *dependency*)
- **Powiązanie** (ang. *association*)
- **Uogólnienie** (ang. *generalization*)

Przykład współfinansowania projektu Europejski w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Zależność**

- Związek użycia między dwoma elementami.
- Zmiany dokonane w specyfikacji jednego elementu (np. klasy *Zamówienie*) mogą mieć wpływ na inny element, który używa (korzysta) tego pierwszego (np. na klasę *ZamówienieSpecjalne*), ale niekoniecznie na odwrót.
- **Graficznie** – linia przerywana z grotem skierowanym na element, od którego coś zależy:

ZamówienieSpecjalne -----> Zamówienie

Przykład zależności w UML

Przykład współfinansowania projektu Europejski w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Powiązanie (asocjacja)**

- Związek strukturalny równorzędnych partnerów, tzn. żadna klasa nie jest ważniejsza.
- Wskazuje, iż obiekty jednego elementu są połączone z obiektami innego.
- Dla każdej pary klas/obiektów można zdefiniować dowolnie wiele powiązań, z których każde reprezentuje inną zależność.
- **Graficznie** – linia ciągła łącząca klasy:

Klasa1 nazwaRoli krotność nazwaPowiązania nazwaRoli krotność Klasa2

Powiązanie w UML

Przykład współfinansowania projektu Europejski w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Powiązanie

- Można opisać za pomocą:
  - nazwy,
  - roli,
  - krotności.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Nazwa powiązania

- Powiązanie może mieć przypisaną nazwę, która określa istotę danego związku.
- Aby uniknąć niejednoznaczności, można podać kierunek odczytu (trójkątny znacznik przy nazwie).
- Nie jest konieczna, gdy określone są role.
- Zwykle jest to czasownik w czasie teraźniejszym, w 3os. l. poj.

```

    graph LR
      Osoba[Osoba] -- pracujeDla --> Firma[Firma]
  
```

Przykład nazwy powiązania

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Rola

- Klasa biorąca udział w powiązaniu ogrywa w nim określoną rolę.
- „Oblicze”, które klasa przy jednym końcu powiązania prezentuje klasie przy drugim końcu.
- Dana klasa może odgrywać tę samą albo inną rolę w różnych powiązaniach.

```

    graph LR
      Osoba[Osoba] -- pracownik --> Firma[Firma]
      Osoba -- pracodawca --> Firma[Firma]
      Osoba --- K1[1..*]
      Firma --- K2[*]
  
```

Przykład roli i krotności w powiązaniu

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

Szkolenia INSPIRE  
dla administracji

**■ Krotność (liczność)**

- Oznacza ile obiektów może być połączonych przez jeden egzemplarz powiązania.
- Przykłady możliwych krotności dla powiązania w UML:

Krotność	Znaczenie	Np.
–	1 lub nie określono krotności	–
N	dokładnie N	5
*	dowolnie wiele	–
m..n	od m do n	1..5
m..*	od m do dowolnie wiele	2..*

Przykłady możliwych krotności w powiązaniu

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE  
dla administracji

**■ Powiązania w UML**

- Nawigacja** (ang. *navigation*)
- Agregacja zwykła** (ang. *aggregation*)
- Agregacja całkowita, kompozycja** (ang. *composition*)

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE  
dla administracji

**■ Nawigacja**

- Powiązanie jednokierunkowe, tzn. tylko jedna z klas biorących udział w powiązaniu „wie” o drugiej klasie.
- Graficznie** – strzałka dołączona do zakończenia linii powiązania i wskazująca na daną klasę. Brak strzałki oznacza, że powiązaniem można „poruszać się” w obu kierunkach (obie klasy wiedzą o sobie).

```

classDiagram
    class Klasa1
    class Klasa2
    Klasa1 --> Klasa2 : nazwaPowiazania, nazwaRoli, krotnosc
    
```

Nawigacja w UML

---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

### ■ Agregacja zwykła

- Związek „**posiadania**” między klasami, klasa-całość składa się z zestawu innych klas, które stanowią jej składniki (klasy-części); całość „**ma/posiada**” części.
- Klasy-części mogą być składnikami innych agregacji, zaś ich czas życia nie jest ograniczony do czasu życia klasy-całości.
- **Graficznie** – pusty romb umieszczonego na końcu linii powiązania od strony klasy-całości:



Agregacja zwykła w UML

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

### ■ Agregacja całkowita (kompozycja)

- Związek „**zawierania**” między klasami, gdzie klasa-całość zawiera inne klasy, które stanowią jej składniki (klasy-części); całość „**zawiera**” części.
- Klasy-części nie mogą być składnikami innych klas-całości, zaś ich czas życia jest ograniczony do czasu życia klasy-całości.
- **Graficznie** – wypełniony romb umieszczony na końcu linii powiązania od strony klasy-całości:



Agregacja całkowita w UML

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### ■ Uogólnienie (dziedziczenie)

- Związek między elementem ogólnym (generalizacją, nadklasą, przodkiem) a jego pewnym specyficznym rodzajem (specjalizacją, podklasą, potomkiem).
- Potomek może wystąpić wszędzie tam, gdzie jest spodziewany przodek, ale nie na odwrót (potomek może zawsze zastąpić przodka).
- Potomek dziedziczy wszystkie właściwości przodka:
  - atrybuty, - powiązania,
  - operacje, - ograniczenia.
- Najczęściej potomek ma jeszcze własne cechy, poza odziedziczonymi.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

### Uogólnienie (dziedziczenie)

- Klasa może nie mieć przodka, a może mieć jednego lub więcej.
- Może mieć nazwę, ale zwykle nie jest ona potrzebna.
- **Graficznie** – linia ciągła zakończona niewypełnionym trójkątem wskazującym przodka:



Dziedziczenie w UML

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Mechanizmy rozszerzenia w UML

- Pozwalają na rozszerzenie semantyki (słownictwa) UML:
  - umożliwiają dostosowanie UML do potrzeb konkretnego zadania i pozwalają na przystosowanie go do nowych technologii.
  - pozwalają na uszczegółowienie modelu, zdefiniowanie nowych elementów, przystosowanie notacji do specyficznej dziedziny lub preferencji użytkownika.
- Są to:
  - **stereotypy** (ang. *stereotypes*),
  - **metki** (ang. *tagged values*),
  - **ograniczenia** (ang. *constraints*).

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Materiały uzupełniające

- Booch G., Rumbaugh J., Jacobson I. 2002. *UML. Przewodnik użytkownika*. WNT, Warszawa.
- Fowler M., Scott K. 2002. *UML w kropelce*. Oficyna wydawnicza LTP, Warszawa.
- Schmuller J. 2003. *UML dla każdego*. Helion, Gliwice.
- Subieta K. 1999. *Wprowadzenie do obiektowych metody projektowania i notacji UML. Jedenasta Górská Szkoła PTI Szczyrk*.
- Object Management Group (OMG), UML. <http://www.uml.org>
- Portal UML. Projekty systemów informatycznych z UML. <http://www.uml.com.pl>
- ISO/TS 19103:2005 *Geographic information – Conceptual schema language*
- ISO 19109:2009 *Geographic information – Rules for application schema*

---

---

---

---

---

---

---

---


---


---


---

Szkolenia INSPIRE dla administracji


**Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:**

 Opejeka Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA” Spółka z o.o. 82-300 Elbląg, ul. Tysiąclecia 11 [www.opejeka.pl](http://www.opejeka.pl)

 IGiK Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27 [www.igik.edu.pl](http://www.igik.edu.pl)

 GRID Centrum UNEP/GRID-Warszawa, 00-764 Warszawa, ul. Sobieszyńska 8 [www.gridw.pl](http://www.gridw.pl)

Podwykonawca – partner technologiczny:

 Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52 [www.intergraph.pl](http://www.intergraph.pl)

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

■ Szkolenie eksperckie

## Środki formalne modelowania informacji geograficznej – UML, XML, GML

wykładowca:  
IMIĘ NAZWISKO

Zajęcia przygotowane na podstawie autorskiego programu opracowanego przez:  
dr inż. Agnieszkę Chojkę z Uniwersytetu Warmińsko-Mazurskiego w Olsztynie

### 7. Środki formalne modelowania informacji geograficznej – UML, XML, GML

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

■ Środki formalne modelowania informacji geograficznej – UML, XML, GML

## XML

### Reguły budowy schematów aplikacyjnych w XML Schema

Szkolenia INSPIRE dla administracji

■ ISO/DIS 19118

- Określa **reguły kodowania** wykorzystywane podczas wymiany danych przestrzennych.
- Reguła kodowania pozwala na zakodowanie informacji geograficznej, zdefiniowanej przez schematy aplikacyjne i schematy znormalizowane, na strukturę danych niezależną od systemu – odpowiednią do przesyłania i przechowywania danych.
- Na potrzeby neutralnej wymiany danych normy ISO serii 19100 proponują reguły kodowania oparte na języku **XML**, który jest niezależny od platformy informatycznej a ponadto wykazuje interoperacyjność z siecią www.



Szkolenia INSPIRE dla administracji

### Reguła kodowania oparta na XML

- reguły konwersji **schematu aplikacyjnego UML** na **schemat struktur danych zapisany w XML Schema**,
- reguły konwersji **instancji na struktury danych zapisane w dokumencie XML**.

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Schemat XML (XSD)

- Reguły konwersji schematu definiują jak utworzyć **dokument schematu XML (XSD, ang. XML Schema Definition)** zgodnie ze schematem aplikacyjnym wyrażonym w UML.
- Główny cel – zapewnienie, że dokumenty XML wytworzone przy użyciu reguł konwersji danych będą poprawne.
- Powinny zawierać definicje typów, deklaracje atrybutów i elementów, które odpowiadają klasom zdefiniowanym w schemacie aplikacyjnym.
- Fizycznie może być pojedynczym dokumentem schematu lub może być podzielony na kilka oddzielnych.

Program współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### XML

- Standard opracowany przez W3C (ang. *World Wide Web Consortium*).
- Rozszerzalny język znaczników.
- Zaprojektowany do przesyłania i przechowywania danych.
- Język znaczników, podobnie jak język HTML (ang. *HyperText Markup Language*).
- Programowo i sprzętowo niezależne narzędzie przenoszenia informacji (HTML odpowiada jedynie za jej wyświetlanie).

Program współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

■ XML Schema

- Standard opracowany przez W3C (ang. *World Wide Web Consortium*).
- Schemat rozszerzalnego języka znaczników.
- Opisuje strukturę dokumentu XML.
- Wykorzystuje składnię języka XML.
- Dokumenty zawierające definicje XML Schema zapisywane w plikach z rozszerzeniem *.XSD*.

Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

■ Zasady składni dokumentu XML

- musi zaczynać się od deklaracji XML,
- musi mieć jeden unikalny element główny („korzeń”, ang. *root*) – „rodzic” dla pozostałych elementów „dzieci”,
- znaczniki otwierające/początkowe (np. `<notatka>`) muszą posiadać odpowiadające im znaczniki zamykające/końcowe (np. `</notatka>`),
- znaczniki rozróżniają małe i duże litery,
- wszystkie elementy muszą być zamknięte,
- wszystkie elementy muszą być odpowiednio zagnieżdżone,
- wszystkie wartości atrybutów muszą być ujęte w cudzysłów.

Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

■ Dokument XML – przykład

```
<?xml version="1.0" encoding="ISO-8859-2"?>  
<notatka>  
<dla>Tomka</dla>  
<od>Ania</od>  
<tytuł>Przypomnienie</tytuł>  
<treść>Nie zapomnij o mnie podczas weekendu!</treść>  
</notatka>
```

- Pierwszy wiersz – **deklaracja XML** – definiuje wersję XML (1.0) i stosowane kodowanie (ISO-8859-2, zbiór znaków dla Europy Środkowej i Wschodniej).
- Następna linia opisuje **element główny** („korzeń”) dokumentu: `<notatka>`.
- Kolejne 4 wiersze opisują 4 **elementy „dzieci”** elementu głównego: `<dla>`, `<od>`, `<tytuł>`, `<treść>`.
- Ostatni wiersz definiuje **koniec elementu głównego**: `</notatka>`.

Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---



Szkolenia INSPIRE dla administracji

### Elementy XML

- Muszą stosować się do poniższych zasad nazywania:
  - nazwy mogą zawierać litery, liczby i inne znaki,
  - nazwy nie mogą rozpoczynać się od liczby lub znaku przestankowego (np. . , ; : ' ? - ),
  - nazwy nie mogą rozpoczynać się od liter xml (lub XML, lub Xml, itp.),
  - nazwy nie mogą zawierać spacji.

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Atrybut a element w XML – przykład

– Oba przykłady dotyczą tych samych informacji.

<pre>&lt;osoba plec="kobieta"&gt; &lt;imię&gt;Anna&lt;/imię&gt; &lt;nazwisko&gt;Nowak&lt;/nazwisko&gt; &lt;/osoba&gt;</pre>	<pre>&lt;osoba&gt; &lt;plec&gt;kobieta&lt;/plec&gt; &lt;imię&gt;Anna&lt;/imię&gt; &lt;nazwisko&gt;Nowak&lt;/nazwisko&gt; &lt;/osoba&gt;</pre>
---	---

– *plec* jest atrybutem    – *plec* jest elementem

- Dla **danych** najlepiej używać **elementów**.
- Dla **informacji** (które nie dotyczą danych) należy używać **atrybutów**.

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Przestrzeń nazw

- W XML nazwy elementów definiowane są przez użytkownika – może prowadzić do konfliktów nazw elementów pochodzących z różnych dokumentów XML.
- Aby temu zapobiec stosuje się przedrostek nazwy, dla którego musi być zdefiniowana tzw. **przestrzeń nazw** (ang. *namespace*).
  - Określana przez atrybut **xmlns** w znaczniku rozpoczynającym element: `xmlns:prefix="URI"` (URI, ang. *Uniform Resource Identifier*).
  - **Przestrzeń nazw URI** nie jest używana do szukania informacji. Jej zadaniem jest dawanie unikalnych nazw przestrzeni nazw.

---

---

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

### ■ Przestrzeń nazw – przykład

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<domp>
<stół xmlns:p="http://www.sklep.meblowy.com/jadania">
<jazwa>Olgiend</jazwa>
<jszerokość>90</jszerokość>
<jdługość>120</jdługość>
</stół>
<nstół xmlns:n="http://www.warsztat.com/narzędzia">
<n.garaz>
<n.narzędzia>imadło<n.narzędzia>
<n.narzędzia>młotek<n.narzędzia>
</n.garaz>
</nstół>
</domp>
```

- Atrybut `xmlns` w znaczniku `<stół>` otrzymał przedrostki `j` i `n`: określające przestrzeń nazw.
- Kiedy przestrzeń nazw jest zdefiniowana dla elementu, wszystkie elementy „dzieci” z tym samym przedrostkiem są powiązane z tą samą przestrzenią nazw.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

### ■ XML Schema

- Definiuje:
  - elementy, które mogą pojawić się w dokumencie,
  - atrybuty, które mogą pojawić się w dokumencie,
  - które elementy są elementami „dziećmi”,
  - kolejność (porządek) elementów „dzieci”,
  - liczbę elementów „dzieci”,
  - czy element jest pusty, czy może zawierać tekst,
  - typy danych dla elementów i atrybutów,
  - wartości domyślne i stałe dla elementów i atrybutów.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

### ■ XML Schema

- Elementem głównym („korzeniem”) każdego dokumentu XML Schema jest element `<schema>`.
  - Może on zawierać atrybuty.
  - **Elementy globalne** – elementy będące bezpośrednio „dziećmi” elementu `<schema>`.
  - **Elementy lokalne** – elementy zagnieżdżone wewnątrz innych elementów.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---


---

Szkolenia INSPIRE dla administracji

### ■ Deklaracja schematu

```
<?xml version="1.0"?>  
<x:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema"  
  targetNamespace="http://www.w3schools.com"  
  xmlns="http://www.w3schools.com"  
  elementFormDefault="qualified">  
  ...  
</x:schema>
```

- **xmlns:xs="http://www.w3.org/2001/XMLSchema"** – elementy i typy danych użyte w schemacie pochodzą z przestrzeni nazw <http://www.w3.org/2001/XMLSchema>. Określa również, że elementy i typy danych, które pochodzą z tej przestrzeni nazw powinny być poprzedzone przedrostkiem xs.
- **targetNamespace="http://www.w3schools.com"** – elementy zdefiniowane przez schemat (np. <książka>, <tytuł>, <autor>) pochodzą z przestrzeni nazw <http://www.w3schools.com>.
- **xmlns="http://www.w3schools.com"** – domyślną przestrzeń nazw jest <http://www.w3schools.com>.
- **elementFormDefault="qualified"** – każdy element użyty w instancji (egzemplarz) dokumentu XML, który został zadeklarowany w schemacie musi mieć określoną przestrzeń nazw.



---

---

---

---

---

---

---

---

---

---

---


---

Szkolenia INSPIRE dla administracji

### ■ Dokument XML

```
<?xml version="1.0" encoding="ISO-8859-2"?>  
<księgamia xmlns="http://www.w3schools.com"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.w3schools.com księgamia.xsd">  
  <książka księganame="DZIECI">  
    <tytuł>Harry Potter i czarna ognia</tytuł>  
    <autor>J.K. Rowling</autor>  
    <rok>2001</rok>  
    <cenaa49.99</cena>  
  </książka>  
</księgamia>
```

- **xmlns="http://www.w3schools.com"** określa domyślną deklarację przestrzeni nazw, która oznacza, że wszystkie elementy użyte w tym dokumencie XML są zadeklarowane w przestrzeni nazw <http://www.w3schools.com>.
- Jeżeli przestrzeń nazw instancji XML Schema jest dostępna: **xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"** można zastosować atrybut **schemaLocation**. Ma on dwie wartości:
  - pierwszą jest przestrzeń nazw,
  - drugą lokalizacja schematu XML dla tej przestrzeni nazw: **xsi:schemaLocation="http://www.w3schools.com księgamia.xsd"**



---

---

---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

### ■ Typy proste (element simpleType)

- **Element prosty** – element XML, który:
  - zawiera tylko tekst,
  - nie może zawierać żadnych innych elementów i atrybutów.
- Składania definicji elementu prostego:  
**<xs:element name="xxx" type="yyy"/>**
  - "xxx" nazwa elementu,
  - "yyy" typ danych tego elementu.



---

---

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

■ **Typy proste (element simpleType)**

- XML Schema posiada wiele wbudowanych typów danych. Najbardziej popularne to:
  - xs:string
  - xs:decimal
  - xs:integer
  - xs:boolean
  - xs:date
  - xs:time
- Element prosty może mieć również określoną wartość domyślną (**default**) lub stałą (**fixed**).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

■ **Typy proste (element simpleType)**

- W XML Schema wszystkie atrybuty są zadeklarowane jako typy proste.
- Elementy proste nie mogą mieć atrybutów.
- Element, który posiada atrybuty jest typu złożonego (**complexType**).
- Atrybut (sam w sobie) jest zawsze zadeklarowany jako typ prosty (**simpleType**).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---


---

---

---

---

---

Szkolenia INSPIRE dla administracji 

■ **Typy proste (element simpleType)**

- Składania definiowania atrybutu:  
`<xs:attribute name="xxx" type="yyy"/>`
  - "xxx" nazwa atrybutu,
  - "yyy" typ danych atrybutu.
- Atrybut może:
  - mieć określoną wartość domyślną lub stałą,
  - być opcjonalny lub wymagany (**use="required"**). Domyślnie atrybut jest opcjonalny.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Typy proste (element **simpleType**)

- W XML Schema można zdefiniować ograniczenie (**restriction**) zawartości elementu lub atrybutu, które jest stosowane do określania akceptowalnych wartości elementów i atrybutów XML.

Ograniczenie	Opis
enumeration	Definiuje listę dopuszczalnych wartości.
fractionDigits	Określa max liczbę dziesiętnych miejsc dziesiętnych. Musi być większe lub równe 0.
length	Określa dokładną liczbę znaków lub listę elementów dozwolonych. Musi być większe lub równe 0.
maxExclusive	Określa górną granicę dla wartości numerycznych (jej wartość musi być mniejsza niż ta wartość).
maxInclusive	Określa górną granicę dla wartości numerycznych (jej wartość musi być mniejsza lub równa tej wartości).
maxLength	Określa max liczbę znaków lub listę elementów dozwolonych. Musi być większe lub równe 0.
minExclusive	Określa dolną granicę dla wartości numerycznych (jej wartość musi być większa niż ta wartość).
minInclusive	Określa dolną granicę dla wartości numerycznych (jej wartość musi być większa lub równa tej wartości).
minLength	Określa min liczbę znaków lub listę elementów dozwolonych. Musi być większe lub równe 0.
pattern	Definiuje dokładną sekwencję dopuszczalnych znaków.
totalDigits	Określa całkowitą liczbę cyfr w wartości (jej wartość musi być większa lub równa 0).
whiteSpace	Określa jak „białe” znaki (whitespace – przesunięcia o wiersz, tabulatory, spacje i powroty karek) są obsługiwane.

Ograniczenia dla typów danych  
Źródło: W3Schools Online  
Web Tutorials

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Ograniczenie – przykład

```
<xs:element name="haslo">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="5"/>
<xs:maxLength value="8"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
```

- Element prosty "haslo" z ograniczeniem zawartości.
- Długość "hasła" musi wynosić co najmniej 5 i co najwyżej 8 znaków.

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Typy złożone (element **complexType**)

- Element złożony** – element XML, który:
  - zawiera inne elementy i/lub
  - atrybuty.
- Wyróżnia się 4 rodzaje elementów złożonych:
  - elementy puste,
  - elementy, które zawierają tylko inne elementy,
  - elementy, które zawierają tylko tekst,
  - elementy, które zawierają zarówno inne elementy jak i tekst.

Każdy z tych elementów może również zawierać atrybuty.

---

---

---

---

---

---

---

---

---

---

---

---



Szkolenia INSPIRE dla administracji

**Deklaracja elementu złożonego**

- W XML Schema można go zdefiniować na dwa sposoby:
  - bezpośrednio zadeklarować element złożony przez nazwanie go lub
  - przypisać elementowi złożonemu nazwę i atrybut **type**, który odnosi się do nazwy.
    - Wtedy kilka elementów w schemacie może odwoływać się do tego samego typu złożonego.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Deklaracja elementu złożonego – przykład**

- **Element złożony z mieszaną zawartością**
  - może zawierać atrybuty, elementy i tekst.

Element XML "list", który zawiera tekst i inne elementy:

```

<list>
  Szanowny Panie<nazwisko>Kowalski</nazwisko>.
  Pana zamówienie<id_zamowienia>1032</id_zamowienia>
  zostanie zrealizowane<data_dostawy>2011-06-25</data_dostawy>.
</list>
          
```

Definicja "list" elementu w XML Schema:

```

<xsd:element name="list">
  <xsd:complexType mixed="true">
    <xsd:sequence base="xsd:string" minOccurs="1" maxOccurs="1" use="required"/>
    <xsd:element name="nazwisko" type="xsd:string" minOccurs="1" maxOccurs="1" use="required"/>
    <xsd:element name="id_zamowienia" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1" use="required"/>
    <xsd:element name="data_dostawy" type="xsd:date" minOccurs="1" maxOccurs="1" use="required"/>
    <xsd:sequence base="xsd:string" minOccurs="1" maxOccurs="1" use="required"/>
  </xsd:complexType>
</xsd:element>
          
```

- Aby w elemencie "list" mogły pojawić się dane typu znakowego między elementami „dzieci”, atrybut **mixed** musi być ustawiony na "true".
- **<xsd:sequence>** – zdefiniowane elementy ("nazwisko", "id\_zamowienia" i "data\_dostawy") muszą pojawić się dokładnie w takiej kolejności wewnątrz elementu "list".
- Elementowi **complexType** można przypisać nazwę a elementowi "list" atrybut **type**, który odnosi się do nazwy **complexType** (kilkę elementów w schemacie może odwoływać się do tego samego typu złożonego).

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Wskaźniki**

- Pozwalają kontrolować sposób używania elementów w dokumencie:
  - **Wskaźniki porządkowe** – definiują kolejność elementów:
    - **all** – elementy „dzieci” mogą pojawić się w dowolnej kolejności, i każdy element „dziecko” musi pojawić się tylko raz,
    - **choice** – jeden albo więcej elementów „dziecko” może wystąpić,
    - **sequence** – elementy „dzieci” muszą pojawić się w określonej kolejności.
  - **Wskaźniki występowania** – definiują częstość występowania elementów:
    - **maxOccurs** – max ilość wystąpień elementu,
    - **minOccurs** – min ilość wystąpień elementu.
  - **Wskaźniki grupy** – definiują powiązane zbiory elementów:
    - **group** – nazwana grupa elementów,
    - **attributeGroup** – nazwana grupa atrybutów.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Wskazniki – przykład

```
<xs:group name="podstawowe_dane_osobowe">
  <xs:sequence>
    <xs:element name="imię" type="xs:string" maxOccurs="3"/>
    <xs:element name="nazwisko" type="xs:string"/>
    <xs:element name="data_urodzenia" type="xs:date" minOccurs="0"/>
  </xs:sequence>
</xs:group>

<xs:element name="osoba" type="dane_osobowe"/>

<xs:complexType name="dane_osobowe">
  <xs:sequence>
    <xs:group ref="podstawowe_dane_osobowe"/>
    <xs:element name="narodowosc" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

- Zdefiniowano grupę elementów "podstawowe\_dane\_osobowe" (wskaznik `group`).
- Elementy w grupie muszą pojawić się dokładnie w podanej kolejności (wskaznik `sequence`).
- Element "imię" może pojawić się maksymalnie 3 razy (wskaznik `maxOccurs`), a element "data\_urodzenia" jest opcjonalny (wskaznik `minOccurs`).
- Po zdefiniowaniu grupy, można się do niej odwołać w innej definicji (`group ref="podstawowe_dane_osobowe"`).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Element zastępowania

- W XML Schema jeden element można zastąpić innym elementem – elementem zastępowania (`substitutionGroup`).
- Najpierw należy zadeklarować element główny (ang. `head`), a następnie pozostałe elementy, które stanowią zastępstwo dla elementu głównego.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Element zastępowania – przykład

```
<xs:element name="nazwisko" type="xs:string"/>
<xs:element name="pseudonim" substitutionGroup="nazwisko"/>

<xs:complexType name="muzyk">
  <xs:sequence>
    <xs:element ref="nazwisko"/>
  </xs:sequence>
</xs:complexType>
```

- Element "nazwisko" może zostać zastąpiony elementem "pseudonim".
- Poprawne dokumenty XML według powyższego XML Schema:

<pre>&lt;muzyk&gt;   &lt;nazwisko&gt;Smiolik&lt;/nazwisko&gt; &lt;/muzyk&gt;</pre>	lub	<pre>&lt;muzyk&gt;   &lt;pseudonim&gt;Smiolik&lt;/pseudonim&gt; &lt;/muzyk&gt;</pre>
--	-----	--

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---



Szkolenia INSPIRE dla administracji

**Klasa <<BasicType>>**

- Należy przekształcić na deklarację **simpleType** w XML Schema.
- Każdy z typów danych zdefiniowany w XML Schema może zostać użyty jako moduł konstrukcyjny do określenia typów podstawowych zdefiniowanych przez użytkownika.
- Podstawowe typy danych zdefiniowane przez ISO/TS 19103 należy przekształcić na odpowiadające im typy danych w XML Schema.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Klasa <<Enumeration>>**

- Należy przekształcić na typ prosty, który ogranicza łańcuch tekstowy do kilku wartości wyliczeniowych.

«Enumeration» Płeć +kobieta +mężczyzna	<pre>&lt;xs:simpleType name="Plec"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:enumeration value="kobieta"&gt;     &lt;xs:enumeration value="męczyzna"&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>
---	--

Przekształcenia klasy <<Enumeration>> Płeć na odpowiednią strukturę w XML Schema

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

**Klasa <<CodeList>>**

- Nie powinna być przekształcana.
- Może być odwzorowana na słownik, który przechowuje pary: *kod* i *wartość* zdefiniowane w liście kodowej.
- Słownik powinien być dostępny publicznie a jego adres *www* należy podać jako URI.
- Atrybut typu lista kodowa należy kodować jako wartość łańcucha znaków (*string*).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Atrybut

- Należy przekształcić na deklarację elementu lub deklarację atrybutu w deklaracji typu złożonego klasy.
  - Jeżeli atrybut jest typu podstawowego z krotnością 0..1, może zostać przekształcony na **deklarację atrybutu**.
    - Należy również określić krotność atrybutu (1, wtedy **optional="false"**, 0..1 – ustawienie domyślne).
  - Wszystkie inne atrybuty należy przekształcić na **deklarację elementu**.

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Atrybut – przykład

**DataTyp=>Osoba**  
+imię[1..3]: CharacterString  
+nazwisko: CharacterString  
+wiek[0..1]: Integer  
+dziecko[0..\*]: Osoba

Domyślna deklaracja elementu:

```
<xs:complexType name="Osoba">  
<xs:sequence>  
<xs:element name="imię" type="CharacterString" minOccurs="1" maxOccurs="3"/>  
<xs:element name="nazwisko" type="CharacterString"/>  
<xs:element name="wiek" type="Integer" minOccurs="0" maxOccurs="1"/>  
<xs:element name="dziecko" type="Osoba" minOccurs="0" maxOccurs="unbounded"/>  
</xs:sequence>  
</xs:complexType>
```

Atrybuty "nazwisko" i "wiek" przekształcono zgodnie z regułą deklaracji atrybutu:

```
<xs:complexType name="Osoba">  
<xs:sequence>  
<xs:element name="imię" type="CharacterString" minOccurs="1" maxOccurs="3"/>  
<xs:element name="dziecko" type="Osoba" minOccurs="0" maxOccurs="unbounded"/>  
</xs:sequence>  
<xs:attribute name="nazwisko" type="CharacterString" optional="false"/>  
<xs:attribute name="wiek" type="Integer"/>  
</xs:complexType>
```

Typ danych Osoba w UML posiada 4 atrybuty: *imię, nazwisko, wiek i dziecko*.  
Atrybuty *nazwisko* i *wiek* można przekształcić na deklarację atrybutu, natomiast atrybut *imię* ma krotność 1..3, a *dziecko* jest typu złożonego, więc muszą zostać odwzorowane na deklaracje elementów.

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Powiązanie (asocjacja)

- Ogólny związek między dwiema klasami (źródłową i docelową) – obiekty źródłowe przechowują odwołania do obiektów docelowych i na odwrót.
  - Typ złożony odpowiadający klasie źródłowej powinien zawierać deklarację elementu (jeśli asocjacja jest nawigowalna i klasa docelowa jest identyfikowana przez nazwę roli).
  - Typ złożony odpowiadający klasie docelowej powinien zawierać deklarację elementu (jeśli asocjacja jest nawigowalna i klasa źródłowa jest identyfikowana przez nazwę roli).

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

## Powiązanie (asocjacja) – przykład

• Powiązanie między klasą *Firma* i *Osoba*, gdzie tylko klasa *Firma* wie o klasie *Osoba* – zdefiniowano tylko jedną nazwę roli.

Przekształcenie asocjacji na odpowiednią deklarację w XML Schema:

```
<xs:complexType name="Firma">
  <xs:sequence>
    <xs:element name="pracownik" type="OdwolanieDoOsoba" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="OdwolanieDoOsoba">
  <xs:attributeGroup ref="TM_ObjectReference"/>
</xs:complexType>
```

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

## Agregacja zwykła

- Slaby związek typu „całość-część” między klasą-całość a klasą-część („części” mogą być członkami więcej niż jednej „całości” w tym samym czasie).
- Ogólnie „całość” może tylko przechowywać odniesienia do swoich „części”, ale w przypadku całkowitego posiadania może zawierać odpowiednie „części”.
  - Typ złożony odpowiadający klasie „całość” powinien zawierać deklarację elementu, gdzie nazwa odpowiada nazwie roli identyfikującej klasę „część”.
  - Typ złożony odpowiadający klasie „część” powinien zawierać deklarację elementu (jeśli powiązanie jest nawigowalne i klasa docelowa jest identyfikowana przez nazwę roli).

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

## Agregacja zwykła – przykład

• Agregacja zwykła między „całością” *Zespół* i „częścią” *Muzyk*.  
 • *Zespół* identyfikuje *Muzyka* przez nazwę roli *członek* oraz *Muzyk* identyfikuje *Zespół* przez nazwę roli *kapela*.

Przekształcenie agregacji zwykłej na odpowiednią deklarację w XML Schema:

```
<xs:complexType name="Zespół">
  <xs:sequence>
    <xs:element name="członek" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Muzyk">
  <xs:sequence>
    <xs:element name="Zespół" type="Zespół" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attributeGroup ref="TM_ObjectReference"/>
</xs:complexType>
<xs:complexType name="Muzyk">
  <xs:sequence>
    <xs:element name="kapela" type="Zespół" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

---

---

---

---

---

---

---

---

---

---

---

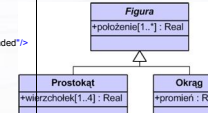
---



Szkolenia INSPIRE dla administracji

### Dziedziczenie pojedyncze – przykład

```
<xs:complexType name="Figura" abstract="true">  
  <xs:complexContent base="TM_Object">  
    <xs:sequence>  
      <xs:element name="poloznienie" type="Real" maxOccurs="unbounded"/>  
    </xs:sequence>  
  </xs:complexContent>  
</xs:complexType>  
  
<xs:complexType name="Prostokat">  
  <xs:complexContent base="Figura">  
    <xs:sequence>  
      <xs:element name="wierzcholek" type="Real" maxOccurs="4"/>  
    </xs:sequence>  
  </xs:complexContent>  
</xs:complexType>  
  
<xs:complexType name="Okrag">  
  <xs:complexContent base="Figura">  
    <xs:sequence>  
      <xs:element name="promien" type="Real"/>  
    </xs:sequence>  
  </xs:complexContent>  
</xs:complexType>
```



- Nadtyp Figura z podtypami Prostokat i Okrag.
- Klasa Figura jest klasą abstrakcyjną.

Przekształcenie dziedziczenia pojedynczego na odpowiednią strukturę w XML Schema

---

---

---

---

---

---

---

---

---

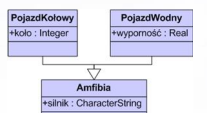
---

---

---

Szkolenia INSPIRE dla administracji

### Dziedziczenie wielokrotne – przykład



```
<xs:complexType name="PojazdKolowy">  
  <xs:sequence>  
    <xs:element name="kolo" type="Integer"/>  
  </xs:sequence>  
  <xs:attributeGroup ref="TM_ObjectIdentification"/>  
</xs:complexType>  
  
<xs:complexType name="PojazdWodny">  
  <xs:sequence>  
    <xs:element name="wypornosc" type="Real"/>  
  </xs:sequence>  
  <xs:attributeGroup ref="TM_ObjectIdentification"/>  
</xs:complexType>  
  
<xs:complexType name="Amfibia">  
  <xs:sequence>  
    <xs:element name="kolo" type="Integer"/>  
    <xs:element name="wypornosc" type="Real"/>  
    <xs:element name="silnik" type="CharacterString"/>  
  </xs:sequence>  
  <xs:attributeGroup ref="TM_ObjectIdentification"/>  
</xs:complexType>
```

- Typy PojazdKolowy, PojazdWodny i Amfibia.
- Amfibia jest podtypem dwóch pozostałych.

Przekształcenie dziedziczenia wielokrotnego na odpowiednią strukturę w XML Schema

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Materiały uzupełniające

- W3Schools Online Web Tutorials. <http://www.w3schools.com>
- ISO/TS 19103:2005 Geographic information – Conceptual schema language
- ISO 19109:2009 Geographic information – Rules for application schema
- ISO/DIS 19118:2005 Geographic information – Encoding

---

---

---

---

---

---

---

---

---

---


---


---




Szkolenia INSPIRE  
dla administracji


**Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:**

 Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA” Spółka z o.o. 62-300 Elbląg, ul. Tysiąclecia 11  
[www.opegielka.pl](http://www.opegielka.pl)

 Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27  
[www.igik.edu.pl](http://www.igik.edu.pl)

 Centrum LINFP/GRID-Warszawa 00-764 Warszawa ul. Śnieżczyńska 8  
[www.gridw.pl](http://www.gridw.pl)

Podwykonawca – partner technologiczny:

 Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52  
[www.intergraph.pl](http://www.intergraph.pl)

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

■ Szkolenie eksperckie

### Środki formalne modelowania informacji geograficznej – UML, XML, GML

wykładowca:  
IMIĘ NAZWISKO

Zajęcia przygotowane na podstawie autorskiego programu opracowanego przez:  
dr inż. Agnieszkę Chojkę z Uniwersytetu Warmińsko-Mazurskiego w Olsztynie

## 8. Środki formalne modelowania informacji geograficznej – UML, XML, GML

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

■ Środki formalne modelowania informacji geograficznej – UML, XML, GML

### G M L

#### Reguły budowy schematów aplikacyjnych w GML (XML Schema)

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

■ ISO 19136

- Specyfikacja implementacyjna języka GML (ang. *Geography Markup Language*).
- Określa **zasady przekształcania schematów aplikacyjnych** zapisanych w UML zgodnie z ISO 19109 na schematy aplikacyjne GML zapisane w XML Schema.

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

**GML**

- **Język znaczników geograficznych** – format wymiany danych przestrzennych pomiędzy różnymi systemami geoinformacyjnymi.
- **Aplikacja XML** – określa regułę kodowania dla schematów aplikacyjnych zgodnych z normami ISO serii 19100, opartą na XML zgodnie z ISO 19118.
- **Definiuje sposób zapisu w XML Schema** określonych właściwości przestrzennych i nieprzestrzennych (zdefiniowanych w normach ISO serii 19100) obiektów geograficznych, np. jednostki miary, geometria i topologia, systemy odniesienia.




---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

**Schemat GML**

- Składa się z komponentów w przestrzeni nazw XML <http://www.opengis.net/gml/3.2>, które przeznaczone są do zapisu określonych właściwości przestrzennych i nieprzestrzennych obiektów geograficznych.




---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

**Schemat aplikacyjny GML**

- Schemat aplikacyjny zapisany w języku **XML Schema** zgodnie z regułami określonymi w ISO 19136.
- Dla określonej dziedziny zastosowań może wymagać rozszerzenia lub ograniczenia typów zdefiniowanych przez schemat GML.
- Powinien:
  - być **określony w XML Schema**,
  - **importować schemat GML**.




---

---

---

---

---

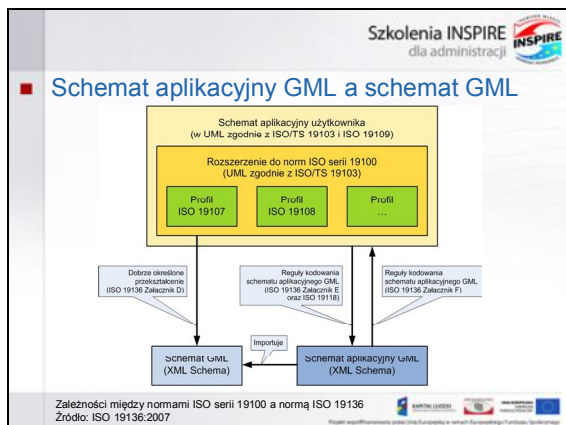
---

---

---

---

---




---

---

---

---

---

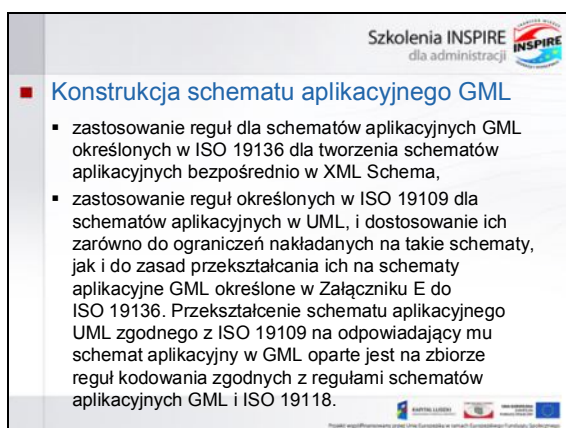
---

---

---

---

---




---

---

---

---

---

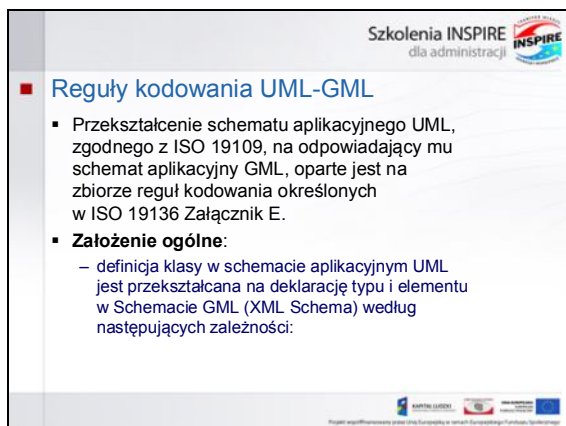
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---



Szkolenia INSPIRE dla administracji

### Klasa

- Na schemat aplikacyjny GML przekształcane są jedynie klasy UML:
  - bez stereotypu,
  - ze stereotypem:
    - <<FeatureType>>,
    - <<Type>>,
    - <<DataType>>,
    - <<Union>>,
    - <<CodeList>>,
    - <<Enumeration>>.
- Wszystkie klasy UML powinny mieć zero lub jeden nadtyp.
- Każda klasa UML przekształcana jest na typ nazwany i otrzymuje przyrostek „Type”.

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Typy danych

- Zdefiniowane przez normy ISO serii 19100 przekształcane są na odpowiadające im typy danych w XML Schema.

Klasa UML	Element obiektu GML	Typ GML	Typ właściwości GML
GM_Object	gml:AbstractGeometry	gml:AbstractGeometryType	gml:GeometryPropertyType
GM_Point	gml:Point	gml:PointType	gml:PointPropertyType
TP_Edge	gml:Edge	gml:EdgeType	gml:DirectedEdgePropertyType
SC_CRS	gml:AbstractCRS	gml:AbstractCRSType	gml:CRSPropertyType
CharacterString	-	-	xsd:string
Integer	-	-	xsd:integer, xsd:nonNegativeInteger, xsd:negativeInteger, xsd:nonNegativeInteger, xsd:positiveInteger
Length, Distance	-	-	gml:lengthType

Przykłady implementacji typów danych z norm ISO serii 19100 w schemacie aplikacyjnym GML  
Źródło: ISO 19136:2007

---

---

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Klasa <<DataType>>

- Przekształcana na typ złożony (element **complexType**) w XML Schema.
- Należy zdefiniować elementy globalne XML z odpowiednimi ustawieniami dla nazwy (nazwa klasy UML), typu (nazwa klasy z przyrostkiem „Type”), abstrakcyjności (jeśli klasa jest abstrakcyjna) i grupą zastępowania (nazwa określająca element nadtypu lub *gml:AbstractObject*, jeśli klasa nie ma nadtypu).
- Należy utworzyć nazwany typ złożony, którego nazwa zawiera nazwę klasy UML z przyrostkiem „PropertyType”, jeśli dla klasy nie ustawiono metki „noPropertyType” z wartością „true”.

---

---

---

---

---

---

---

---

---

---

---

---








Szkolenia INSPIRE dla administracji

### Klasa <<CodeList>>

- Bez ustawionej metki „asDictionary” na „true” powinna zostać przekształcona tak jak klasa ze stereotypem <<Enumeration>>, ale z tą różnicą, że:
  - Należy dodać wzorzec „<pattern value=’other: lw{2,}’>”, który dopuszcza inne wartości tekstowe poza zdefiniowanymi (wartości te są poprzedzone przedrostkiem „other:”).




---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Klasa <<CodeList>> – przykład

```


<simpleType name="PrzeznaczenieTerenuType">
  <union memberTypes="np:PrzeznaczenieTerenuEnumerationType np:PrzeznaczenieTerenuOtherType"/>
</simpleType>

<simpleType name="PrzeznaczenieTerenuEnumerationType">
  <restriction base="string">
    <enumeration value="1">
      <annotation>
        <appinfo><gml:description>zabudowa Mieszkaniowa</gml:description></appinfo>
      </annotation>
    </enumeration>
    <enumeration value="2">
      <annotation>
        <appinfo><gml:description>droga</gml:description></appinfo>
      </annotation>
    </enumeration>
    <enumeration value="3">
      <annotation>
        <appinfo><gml:description>zbiornikWodny</gml:description></appinfo>
      </annotation>
    </enumeration>
  </restriction>
</simpleType>

<simpleType name="PrzeznaczenieTerenuOtherType">
  <restriction base="string">
    <pattern value="other: lw{2,}">
  </restriction>
</simpleType>
    
```

<<CodeList>>  
PrzeznaczenieTerenu  
#zabudowaMieszkaniowa = 1  
#droga = 2  
#zbiornikWodny = 3  
+...

Przekształcenia klasy <<CodeList>> PrzeznaczenieTerenu na odpowiednią strukturę w GML




---

---

---

---

---

---

---

---


---

---

Szkolenia INSPIRE dla administracji

### Atrybut i nazwa roli

- Przekształcane na element lokalny z tą samą nazwą, co typ złożony (**complexType**) definiujący model zawartości typu obiektu.
- Ograniczenia atrybutów **minOccurs** i **maxOccurs** są ustawiane zgodnie z ich definicjami w modelu UML.
- Jeżeli kodowana właściwość jest końcem powiązania i drugi koniec powiązania jest również kodowany na schemat aplikacyjny GML, nazwa właściwości drugiego końca powiązania powinna zostać przekształcona na element *gml:reversePropertyName* w elementach **annotation** i **appinfo** elementu właściwości.




---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Atrybut i nazwa roli

- Typ zależy od typu wartości właściwości w UML. Jeżeli typ wartości właściwości jest zawartością:
  - prosta – typ stosowany jest bezpośrednio (np. *integer*),
  - złożoną – zostanie użyta właściwość.
    - Domyślne kodowanie właściwości typu pozwala na reprezentację:
      - wbudowaną (*inline*) i przez referencję (*by-reference*) dla typów obiektów.
      - wbudowaną (*inline*) dla typów danych i unii.

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Atrybut i nazwa roli – przykład

```
<element name="wasciel" type="np:OsobaPropertyType" maxOccurs="unbounded">
  <annotation>
    <appinfo>
      <gmml:reversePropertyName>np:wlasnosc</gmml:reversePropertyName>
    </appinfo>
  </annotation>
</element>
...
<complexType name="OsobaPropertyType">
  <sequence minOccurs="0">
    <element ref="np:Osoba"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
...
<element name="wasciel" maxOccurs="unbounded">
  <annotation>
    <appinfo>
      <gmml:reversePropertyName>np:wlasnosc</gmml:reversePropertyName>
    </appinfo>
  </annotation>
  <complexType>
    <sequence minOccurs="0">
      <element ref="np:Osoba"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  </complexType>
</element>
```

+wasciel      +wlasnosc  
Osoba      Osoba  
1..\*      0..\*

Dwa alternatywne przekształcenia właściwości typu (rola w powiązaniu) na reprezentację wbudowaną (*inline*) i przez referencję (*by-reference*) w GML.

---

---

---

---

---

---

---

---

---

---

Szkolenia INSPIRE dla administracji

### Dziedziczenie

- XML Schema obsługuje tylko **dziedziczenie pojedyncze**.
- Realizowane poprzez:
  - mechanizm rozszerzenia** (*extension*) lub
  - mechanizm ograniczenia** (*restriction*) oraz
  - wykorzystanie **elementu zastępowania** (*substitutionGroup*).

```
<element name="Elipsa" type="np:ElipsaType" substitutionGroup="gml:AbstractCurveSegment"/>
<complexType name="ElipsaType">
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <element name="srodek" type="gml:DirectPositionType"/>
        <element name="małaPółś" type="gml:VectorType"/>
        <element name="wielkaPółś" type="gml:VectorType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Przekształcenie klasy **Elipsa**, która dziedziczy z klasy abstrakcyjnej **GM\_CurveSegment**, na odpowiednią strukturę w GML.

---

---

---

---

---

---

---

---


---


---




Szkolenia INSPIRE  
dla administracji


■ *Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:*

 Opegielka Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA” Spółka z o.o. 62-300 Elbląg, ul. Tysiąclecia 11 [www.opegielka.pl](http://www.opegielka.pl)

 Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27 [www.igik.edu.pl](http://www.igik.edu.pl)

 Centrum I INF/GRID-Warszawa 00-764 Warszawa ul. Śnieżczyńska 8 [www.gridw.pl](http://www.gridw.pl)

Podwykonawca – partner technologiczny:

 Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52 [www.intergraph.pl](http://www.intergraph.pl)

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

---

---

---

---

---

---

---

---

---

---



**Edukacyjne wsparcie procesu wdrażania dyrektywy INSPIRE  
w administracji samorządowej w kontekście podniesienia jakości usług i efektywności działania**



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI



**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY

