

# PRG036 – Technologie XML

---

Přednáší:

Irena Mlýnková ([mlynkova@ksi.mff.cuni.cz](mailto:mlynkova@ksi.mff.cuni.cz))

Martin Nečaský ([necasky@ksi.mff.cuni.cz](mailto:necasky@ksi.mff.cuni.cz))

LS 2011

Stránka přednášky:

<http://www.ksi.mff.cuni.cz/~mlynkova/prg036/>

# Osnova předmětu

---

- ❑ Úvod do principů formátu XML, přehled XML technologií, jaz-yk DTD
  - ❑ Datové modely XML, rozhraní DOM a SAX
  - ❑ Úvod do jazyka XPath
  - ❑ Úvod do jazyka XSLT
  - ❑ XPath 2.0, XSLT 2.0
  - ❑ Úvod do jazyka XML Schema
  - ❑ Pokročilé rysy jazyka XML Schema
  - ❑ Přehled standardních XML formátů
  - ❑ Úvod do jazyka XQuery
  - ❑ Pokročilé rysy jazyka XQuery, XQuery Update
  - ❑ Úvod do XML databází, nativní XML databáze, číslovací schémata, structural join
  - ❑ Relační databáze s XML rozšířením, SQL/XML
-

# Připomenutí z minula (1)

---

- ❑ Správně zformovaný vs. validní XML dokument
  - ❑ XML Schema – jazyk pro definici přípustné struktury XML dokumentů
    - Doporučení konsorcia W3C
    - Výhody i nevýhody
      - ❑ viz minulá přednáška
  - ❑ Prvky jazyka = elementy ze jmenného prostoru jazyka XML Schema
    - Vlastnosti: atributy / podelementy / hierarchie
-

# Připomenutí z minula (2)

---

## ☐ Prvky jazyka:

- Základní – jednoduchý datový typ, složený datový typ, element, atribut, modelová skupina (skupina elementů), skupina atributů

☐ viz minulá přednáška

- Pokročilé – omezení identity, substituční skupiny, zástupci, externí schémata, ...



## ☐ Základní definice XML schématu:

- Definice jednoduchých + složených datových typů
  - Přřazení datových typů atributům / elementům
-

# Připomenutí z minula (3)

---

- Opakovaně využívané prvky definovány globálně + využití referencí
  - Elementy, atributy, modelové skupiny a skupiny atributů

Jednotlivé prvky podrobněji...

---

# Vestavěné jednoduché typy

---

- ☐ Omezení textového řetězce na množinu přípustných hodnot
  - ☐ Přímo součástí jazyka XML Schema
  - ☐ Dělení: ([www](http://www.w3.org/XML/1998/08/xmlschema-2#built-in-simple-types))
    - Základní (20 typů – 19 z verze 1.0)
    - Odvozené
      - ☐ Od typu string – řetězcové typy (12 typů)
      - ☐ Od typu decimal – číselné typy (13 typů)
-

# Základní jednoduché typy (1)

---

- ❑ string – řetězec znaků
  - ❑ boolean – logické hodnoty true, false, popř. 1, 0
  - ❑ decimal – kladné nebo záporné reálné číslo
    - např. -1.23, 1267.5433, 210
  - ❑ float – 32-bitové číslo vyjádřené pomocí mantisy a exponentu
    - Hodnoty:  $m \times 2^e$ , kde  $|m| < 2^{24}$ ,  $-149 \leq e \leq 104$
    - např. -1E4, 1267.43233E12, 12
    - Speciální hodnoty: 0, -0, Inf, -Inf, +Inf a NaN
  - ❑ double – 64-bitové číslo se stejnými vlastnostmi
    - Hodnoty:  $m \times 2^e$ , kde  $|m| < 2^{53}$ ,  $-1075 \leq e \leq 970$
-

# Základní jednoduché typy (2)

---

- ❑ duration – časový úsek ve tvaru **P***nY***nMnD****T***nHnMnS*, kde **P** a **T** jsou oddělovače, *nY* znamená *n* let, *nM* znamená *n* měsíců atd.
    - např. -P13Y7M, P2Y1MT2H
  - ❑ dateTime – datum a čas ve tvaru YYYY-MM-DD**T**hh:mm:ss.ss, kde **T** je oddělovač
  - ❑ time – čas ve tvaru hh:mm:ss.ss
  - ❑ date – datum ve tvaru YYYY-MM-DD
  - ❑ gYearMonth – měsíc v roce ve tvaru YYYY-MM
  - ❑ gYear – rok ve tvaru YYYY
-



# Základní jednoduché typy (3)

---

- ❑ gMonthDay – den v měsíci ve tvaru --MM-DD
  - ❑ gMonth – měsíc ve tvaru --MM
  - ❑ gDay – den ve tvaru ---DD
  - ❑ hexBinary – hexadecimální číslo
  - ❑ base64Binary – binární data s kódováním Base64 („obdoba“ hexadecimálního kódování)
  - ❑ anyURI – absolutní nebo relativní URI
  - ❑ QName – XML Qualified Name, tj. řetězec ve tvaru prefix:místní část
    - viz jmenné prostory
  - ❑ NOTATION – odkaz na notaci
    - viz notace
-

# Základní jednoduché typy (4) - verze 1.1

---

- precisionDecimal - kladné nebo záporné reálné číslo s  
možností definovat přesnost, tj. dolní a horní mez exponentu e
    - viz odvozování restrikcí
-

# Základní jednoduché typy (5)



## ☐ Poznámky:

- Časové datové typy lze specifikovat také v UTC (světový čas) nebo s odchylkou od něj
  - ☐ např. 15:30:25Z, 09:30:25+06:00
- Časové datové typy mohou být i záporné
- Vestavěné datové typy pojmenované výhradně velkými písmeny odpovídají stejnojmenným typům z DTD
  - ☐ Platí i dále
- Vestavěné datové typy pojmenované výhradně velkými písmeny (a typy od nich odvozené) mohou být přiřazeny pouze atributům
  - ☐ Platí i dále

# Řetězcové jednoduché typy (1)

---

- ❑ `normalizedString` – string, který neobsahuje znaky CR, LF a tabulátor
  - ❑ `token` – `normalizedString`, který nemá mezery na začátku ani na konci a neobsahuje posloupnost mezer delší než jedna
  - ❑ `language` – identifikátor jazyka
    - Přípustné hodnoty dány speciální normou
    - např. `en`, `en-GB`
  - ❑ `Name` – XML Name, tj. řetězec, který obsahuje písmena, číslice nebo znaky '-', '\_', ':' a '.'
  - ❑ `NCName` – XML Name neobsahující znak ':'
-

# Řetězcové jednoduché typy (2)

---

- ❑ NMTOKEN – jednoslovná hodnota z písmen, číslic a příp. dalších znaků (viz norma)
  - ❑ NMTOKENS – seznam jednoslovných hodnot
  - ❑ ID – hodnota jednoznačná v rámci celého XML dokumentu
  - ❑ IDREF – odkaz na hodnotu typu ID
  - ❑ IDREFS – seznam odkazů na hodnoty typu ID
  - ❑ ENTITY – odkaz na entitu
    - Entity lze definovat pouze v DTD!!
  - ❑ ENTITIES – seznam odkazů na entity
-

# Číselné jednoduché typy (1)

---

- ☐ integer – celé číslo
  - ☐ positiveInteger – kladné celé číslo
  - ☐ negativeInteger – záporné celé číslo
  - ☐ nonPositiveInteger – nekladné celé číslo
  - ☐ nonNegativeInteger – nezáporné celé číslo
  - ☐ long – celé číslo z intervalu  $< -2^{63}, 2^{63}-1 >$
  - ☐ int – celé číslo z intervalu  $< -2^{31}, 2^{31}-1 >$
  - ☐ short – celé číslo z intervalu  $< -2^{15}, 2^{15}-1 >$
  - ☐ byte – celé číslo z intervalu  $< -2^7, 2^7-1 >$
-

# Číselné jednoduché typy (2)

---

- ☐ unsignedLong – nezáporné číslo menší než  $2^{64}$
  - ☐ unsignedInt – nezáporné číslo menší než  $2^{32}$
  - ☐ unsignedShort – nezáporné číslo menší než  $2^{16}$
  - ☐ unsignedByte – nezáporné číslo menší než  $2^8$
-

# Element simpleType

## – jednoduchý typ

---

- ❑ Definice vlastního jednoduchého typu
  - Restrikcí, seznamem, sjednocením
- ❑ Atributy:
  - id – jednoznačný identifikátor elementu simpleType
  - name – jméno jednoduchého typu
    - ❑ Pro globálně definované jednoduché typy
  - final – zákaz dalšího odvozování
    - ❑ restriction, union, list, #all
    - ❑ Implicitní hodnota: atribut **finalDefault** elementu schema
- ❑ Obsah:
  - (annotation?, (restriction | list | union))





# Element restriction

## – odvození restrikcí

---

- ☐ Omezení hodnot původního typu dle některého z možných omezení
  - ☐ Atributy:
    - id – jednoznačný identifikátor elementu restriction
    - base – název původního (omezovaného) typu
      - ☐ Lze také určit podelementem simpleType
  - ☐ Obsah:
    - (annotation?, (**simpleType**?, (minExclusive | minInclusive | maxExclusive | maxInclusive | totalDigits | fractionDigits | maxScale | minScale | length | minLength | maxLength | enumeration | whiteSpace | pattern)\*))
-

# Přípustná omezení (1)

---

- ☐ length – počet jednotek daného typu
  - ☐ minLength – min. počet jednotek daného typu
  - ☐ maxLength – max. počet jednotek daného typu
  - ☐ pattern – regulární výraz, kterému musí hodnoty vyhovovat
    - Operátory: . (lib. znak) \ (escape nebo metaznak) ? \* + | () (skupina) {} (opakování) [] (rozsah)
      - ☐ např. `.*\d*.*` ... `"*1234*"`, `a{2,4}` ... `"aaa"`, `(\d|[A-Z])+` ... `"3"`, `"U2"`
    - `\s` (bílé znaky) `\S` (nebílé znaky) `\d` (cifra) `\n` `\t`
  - ☐ enumeration – vyjmenovaná množina hodnot
-

# Přípustná omezení (2)

---

- ❑ `maxInclusive` – hodnoty  $\leq$  zadaná hodnota
- ❑ `minInclusive` – hodnoty  $\geq$  zadaná hodnota
- ❑ `maxExclusive` – hodnoty  $<$  zadaná hodnota
- ❑ `minExclusive` – hodnoty  $>$  zadaná hodnota
- ❑ `totalDigits` – max. počet cifer
- ❑ `fractionDigits` – max. počet cifer za desetinnou čárkou

```
<xs:simpleType name="NeprázdnýŘetězec" final="#all">  
  <xs:restriction base="xs:string">  
    <xs:minLength value="1"/>  
  </xs:restriction>  
</xs:simpleType>
```

---

# Přípustná omezení (3) - verze 1.1

---

- ❑ minScale – minimální exponent e při vyjádření hodnoty ve formě mantisy a exponentu
- ❑ maxScale – maximální exponent e při vyjádření hodnoty ve formě mantisy a exponentu
  - Oba pouze pro typ precisionDecimal

```
<xs:simpleType name="price">  
  <xs:restriction base="precisionDecimal">  
    <xs:totalDigits value="8"/>  
    <xs:minScale value="2"/>  
    <xs:maxScale value="2"/>  
  </xs:restriction>  
</xs:simpleType>
```

DECIMAL (8, 2)

# Přípustná omezení (4)



- ❑ whitespace – zpracování bílých znaků v řetězci
  - preserve – žádné změny
  - replace – znaky CR, LF a tabulátor jsou nahrazeny mezerou
  - collapse – navíc jsou odstraněny mezery na začátku a na konci a posloupnosti mezer nahrazeny mezerou

```
<xs:simpleType name="JménoSVelkýmiPísmeny">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse"/>
    <xs:pattern value="([A-Z]([a-z])*) ?)/>
  </xs:restriction>
</xs:simpleType>
```

# Přípustná omezení (5)

---

- Atributy:
    - id – jednoznačný identifikátor elementu
    - value – hodnota omezení
      - V závislosti na typu omezení
    - fixed – příznak, zda toto omezení lze dalším odvozováním měnit
      - Pro všechna omezení kromě pattern a enumeration
  - Obsah:
    - (annotation?)
-

# Element list

## – odvození seznamem

---

- Seznam hodnot původního typu oddělených bílými znaky (vícehodnotové typy)
  - Problém: seznam řetězcových typů vs. oddělovač bílý znak
- Atributy:
  - id – jednoznačný identifikátor elementu list
  - itemType – název původního (jednohodnotového) typu
    - Lze také určit podelementem simpleType
- Obsah:
  - (annotation?, simpleType?)

```
<xs:simpleType name="SeznamReálnýchČísel">  
  <xs:list itemType="xs:float"/>  
</xs:simpleType>
```

---

# Element union

## – odvození sjednocením

---

- ❑ Sjednocení přípustných hodnot daných typů
- ❑ Atributy:
  - id – jednoznačný identifikátor elementu union
  - memberTypes – seznam názvů sjednocovaných typů
    - ❑ Lze také určit podelementem simpleType
- ❑ Obsah:
  - (annotation?, [simpleType](#)\*)

```
<xs:simpleType name="NenulováCeláČísla">  
  <xs:union memberTypes=  
    "xs:positiveInteger xs:negativeInteger"/>  
</xs:simpleType>
```



# Element attribute – atributy (1)

---

- ☐ Název + jednoduchý datový typ
  - ☐ Atributy:
    - id – jednoznačný identifikátor elementu attribute
    - name – název atributu
    - type – název (jednoduchého) datového typu atributu
      - ☐ Lze také určit podelementem simpleType
    - ref – odkaz na globálně definovaný atribut
      - ☐ Reference – viz minulá přednáška
    - default – implicitní hodnota atributu není-li v XML dokumentu uveden
    - fixed – konstantní hodnota atributu
-

# Element attribute – atributy (2)

---

- use – povinnost výskytu atributu
  - optional, required, prohibited (viz odvozování složených typů)
- form – příznak, zda je nutné uvádět jméno atributu plně kvalifikované
  - qualified, unqualified
  - Implicitní hodnota: atribut **attributeFormDefault** elementu schema
  - Pouze pro lokálně definované atributy
- Obsah:
  - (annotation?, **simpleType?**)

```
<xs:attribute name="Věk"  
              type="xs:positiveInteger"/>
```

---

# Element element – elementy (1)

---

- ❑ Název + jednoduchý / složený datový typ
- ❑ Atributy:
  - id – jednoznačný identifikátor elementu element
  - name – název elementu
  - type – název datového typu elementu
    - ❑ Lze také určit podelementem simpleType / complexType
  - ref – odkaz na globálně definovaný element
    - ❑ Reference – viz minulá přednáška
  - nillable – příznak možného prázdného obsahu



```
<xs:element name="Prázdný"  
            type="xs:int"  
            nillable="true"/>
```

```
<Prázdný>123</Prázdný>  
<Prázdný></Prázdný>  
<Prázdný/>  
<Prázdný xsi:nil="true"/>
```

# Element element – elementy (2)

---

- default – implicitní hodnota elementu, je-li prázdný
  - Pouze má-li jednoduchý obsah
- fixed – konstantní hodnota elementu
  - Pouze má-li jednoduchý obsah
- minOccurs – minimální počet výskytů elementu
- maxOccurs – maximální počet výskytů elementu

```
<xs:element name="Příjmení">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

# Element element – elementy (3)

---

- form – příznak, zda je nutné uvádět jméno elementu plně kvalifikované
    - qualified, unqualified
    - Implicitní hodnota: atribut **elementFormDefault** elementu schema
    - Pouze pro lokálně definované elementy
  - abstract, substitutionGroup, final, block
    - viz substituční skupiny
  - Obsah:
    - (annotation?, ((**simpleType** | **complexType**)?, (**unique** | **key** | **keyref**)\*))
-

# Element complexType

## – složený typ (1)

---

- ❑ Definice složitějších datových typů
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu complexType
    - name – název složeného typu
      - ❑ Pro globálně definované složené typy
    - mixed – příznak smíšeného obsahu
    - abstract – příznak abstraktního datového typu
      - ❑ Nelze jej přiřadit žádnému elementu
    - final – pro daný typ zakazuje další odvozování
      - ❑ restriction, extension, #all
      - ❑ Implicitní hodnota: atribut **finalDefault** elementu schema
    - block – viz implicitní substituovatelnost
-

# Element complexType

## – složený typ (2)

---

- Obsah:
    - (annotation?, (simpleContent | complexContent | ((group | all | choice | sequence)?, ((attribute | attributeGroup)\*, anyAttribute?), (assert | report)\* )))
  - Typy obsahu složeného typu:
    - S jednoduchým obsahem (simpleContent)
    - Posloupnost elementů (sequence)
    - Výběr z elementů (choice)
    - Množina elementů (all)
    - Modelová skupina (group)
    - Se složeným obsahem (complexContent)
-

# Element simpleContent (1)

---

- ❑ Jednoduchý obsah elementu + atributy
- ❑ Atributy:
  - id - jednoznačný identifikátor elementu simpleContent
- ❑ Obsah: (annotation?, (restriction | extension))
- ❑ Element restriction
  - Atributy:
    - ❑ id – jednoznačný identifikátor elementu extension
    - ❑ base – název omezovaného jednoduchého typu / složeného typu s jednoduchým obsahem
  - Obsah:
    - ❑ (annotation?, (**simpleType?**, (minExclusive | minInclusive | maxExclusive | maxInclusive | totalDigits | fractionDigits | **maxScale** | **minScale** | length | minLength | maxLength | enumeration | whiteSpace | pattern)\*)?, ((attribute | attributeGroup)\*, anyAttribute?), (**assert** | **report**)\*)





# Element simpleContent (2)

- Element extension
  - Atributy:
    - id – jednoznačný identifikátor elementu extension
    - base – název rozširovaného jednoduchého typu / složeného typu s jednoduchým obsahem
  - Obsah:
    - (annotation?, ((attribute | attributeGroup)\*, anyAttribute?), (**assert** | **report**)\*)



```
<xs:complexType name="Typ">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="Podtyp" type="xs:string"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

# Element sequence (1)

---

- ❑ Posloupnost prvků
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu sequence
    - minOccurs – minimální počet výskytů posloupnosti
    - maxOccurs – maximální počet výskytů posloupnosti
  - ❑ Obsah:
    - (annotation?, (**element** | **group** | **choice** | **sequence** | **any**)\*)
      - ❑ Žádný all!!
-

# Element sequence (2)

---

```
<xs:complexType name="Osoba">
  <xs:sequence>
    <xs:element name="Jméno" type="xs:string"
      maxOccurs="5"/>
    <xs:element name="Příjmení" type="xs:string"/>
    <xs:element name="DatumNar" type="xs:date"/>
    <xs:element name="Poznámka" type="xs:string"
      minOccurs="0"/>
    <xs:sequence minOccurs="0" maxOccurs="3">
      <xs:element name="Adresa" type="TypAdresa"/>
      <xs:element name="Telefon" type="xs:string"/>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="Id" type="xs:ID"/>
</xs:complexType>
```

# Element choice (1)

---

- ❑ Výběr z prvků
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu choice
    - minOccurs – minimální počet výskytů výběru
    - maxOccurs – maximální počet výskytů výběru
  - ❑ Obsah:
    - (annotation?, (**element** | **group** | **choice** | **sequence** | **any**)\*)
    - ❑ Žádný all!!
-

# Element choice (2)

---

```
<xs:complexType name="TypCeník">
  <xs:choice>
    <xs:sequence>
      <xs:element name="CenaBezDPH" type="xs:string"/>
      <xs:element name="PlnáCena" type="xs:string"/>
    </xs:sequence>
    <xs:element name="SníženáCena" type="xs:string"/>
    <xs:element name="StudentskáCena" type="xs:string"/>
  </xs:choice>
</xs:complexType>
```

# Element all (1)

---

- ❑ Množina elementů
- ❑ Atributy:
  - id – jednoznačný identifikátor elementu all
  - minOccurs – minimální počet výskytů množiny
  - maxOccurs – maximální počet výskytů množiny
- ❑ Obsah:
  - (annotation?, **element**\*) ... verze 1.0
  - (annotation?, (**element** | **any**\*) ... verze 1.1
- ❑ Poznámky:
  - verze 1.0: maxOccurs elementů v množině i celé množiny max. 1
  - verze 1.1: maxOccurs celé množiny max. 1



# Element all (2)

---

```
<xs:complexType name="Kniha">
  <xs:all>
    <xs:element name="Název" type="xs:string"/>
    <xs:element name="Autor" type="xs:string"/>
    <xs:element name="Vydání" type="xs:date"/>
    <xs:element name="ISBN" type="xs:string"/>
    <xs:element name="Poznámka" type="xs:string"
      minOccurs="0"/>
  </xs:all>
</xs:complexType>
```

# Element group (1)

---

- Modelová skupina – opakované využití typů obsahů
  - Atributy:
    - id – jednoznačný identifikátor elementu group
    - name – název modelové skupiny
    - ref – odkaz na modelovou skupinu
      - Reference – viz minulá přednáška
    - minOccurs – minimální počet výskytů skupiny
    - maxOccurs – maximální počet výskytů skupiny
  - Obsah:
    - (annotation?, (**all** | **choice** | **sequence**)?)
-



# Element group (2)

---

```
<xs:group name="SpolečnéElementy">
```

```
  <xs:sequence>
```

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

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

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

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

```
</xs:group>
```

```
<xs:complexType name="Kniha">
```

```
  <xs:sequence>
```

```
    <xs:group ref="SpolečnéElementy" minOccurs="0"/>
```

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

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

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

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

# Element complexContent

---

- ❑ Odvozování nových typů z již existujících
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu complexContent
    - mixed – příznak smíšeného obsahu
  - ❑ Obsah:
    - (annotation?, (restriction | extension))
  - ❑ Typy odvození:
    - Restrikcí (restriction)
    - Rozšířením (extension)
-

# Element restriction (1)

---

- Omezení elementů / atributů původního typu
    - Omezení počtu výskytů / odstranění elementů
      - Atribut maxOccurs
    - Odstranění atributu
      - Atribut use="prohibited" elementu attribute
    - Omezení datového typu elementu / atributu
  - Atributy:
    - id – jednoznačný identifikátor elementu restriction
    - base – název omezovaného globálně definovaného složeného datového typu
  - Obsah:
    - (annotation?, (**group** | **all** | **choice** | **sequence**)?, ((attribute | attributeGroup)\*, anyAttribute?), (**assert** | **report**)\*)
-

# Element restriction (2)

---

```
<xs:complexType name="Publikace">
  <xs:sequence>
    <xs:element name="Název" type="xs:string"/>
    <xs:element name="Autor" type="xs:string" maxOccurs="10"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PublikaceSJednímAutorem">
  <xs:complexContent>
    <xs:restriction base="Publikace">
      <xs:sequence>
        <xs:element name="Název" type="xs:string"/>
        <xs:element name="Autor" type="xs:string"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

# Element extension (1)

---

- ❑ Doplnění nových atributů nebo elementů k původnímu typu
    - Nové elementy jsou přidány „za“ původní
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu extension
    - base – název rozšiřovaného globálně definovaného složeného datového typu
  - ❑ Obsah:
    - (annotation?, ((group | all | choice | sequence)?, ((attribute | attributeGroup)\*, anyAttribute?)), (assert | report)\*)
-

# Element extension (2)

---

```
<xs:complexType name="Publikace">
  <xs:sequence>
    <xs:element name="Název" type="xs:string"/>
    <xs:element name="Autor" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Kniha">
  <xs:complexContent>
    <xs:extension base="Publikace">
      <xs:sequence>
        <xs:element name="ISBN" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

# Elementy assert a report (1)

## - invarianty

---

- ❑ Verze 1.1: Možnost specifikace podmínek na existenci nebo hodnoty podelementů / atributů (jazyk XPath)
  - ❑ Obdoba integritního omezení CHECK v databázích
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu assert / report
    - test – XPath výraz, který musí platit pro specifikované elementy / atributy
  - ❑ Obsah:
    - (annotation?)
  - ❑ Různý význam:
    - assert - chyba, pokud zadaný výraz neplatí
    - report - chyba, pokud zadaný výraz platí
-

# Elementy assert a report (2)

## - invarianty

---

```
<xs:complexType name="Rozsah">
  <xs:attribute name="min" type="xs:int"/>
  <xs:attribute name="max" type="xs:int"/>
  <xs:assert test="@min le @max"/>
</xs:complexType>
```

```
<xs:complexType name="Pole">
  <xs:sequence>
    <xs:element name="Položka" minOccurs="0"
                maxOccurs="unbounded"
                type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="Délka" type="xs:int"/>
  <xs:assert test="@Délka eq fn:count(./Položka)"/>
</xs:complexType>
```



# Element attributeGroup (1)

---

- ❑ Skupina atributů – opakované využití skupiny atributů
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu attributeGroup
    - name – název skupiny atributů
    - ref – odkaz na skupinu atributů
      - ❑ Reference – viz minulá přednáška
  - ❑ Obsah:
    - (annotation?, ((attribute | attributeGroup)\*, anyAttribute?))
-

# Element attributeGroup (2) – příklad

---

```
<xs:attributeGroup name="SpolečnéAtributy">
  <xs:attribute name="Vypůjčena" type="xs:boolean"/>
  <xs:attribute name="Id" type="xs:ID"/>
</xs:attributeGroup>

<xs:complexType name="Kniha">
  <xs:sequence>
    <xs:element name="Název" type="xs:string"/>
    <xs:element name="Vydavatel" type="xs:string"/>
  </xs:sequence>
  <xs:attributeGroup ref="SpolečnéAtributy"/>
</xs:complexType>
```

# Element schema (1)

---

- ❑ Kořenový element schématu
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu schema
    - version – verze vytvářeného schématu
    - xml:lang – jazyk textů ve schématu (např. en, en-GB)
      - ❑ Přípustné hodnoty určuje norma
      - ❑ Atribut ze jmenného prostoru XML 1.0 (vždy k dispozici)
    - targetNamespace – URI cílového jmenného prostoru
    - finalDefault – implicitní hodnota atributu final všech elementů, jednoduchých i složených datových typů
      - ❑ extension, restriction, list, union, #all
-

# Element schema (2)

---

- blockDefault – implicitní hodnota atributu block všech elementů a složených datových typů
    - restriction, extension, substitution, #all
  - attributeFormDefault – implicitní hodnota atributu form všech atributů
  - elementFormDefault – implicitní hodnota atributu form všech elementů
  - xmlns – jmenné prostory schématu
  - Obsah:
    - ((include | import | redefine | annotation)\*, (((simpleType | complexType | group | attributeGroup) | element | attribute | notation), annotation\*)\*)
-

# Implicitní substituovatelnost

---

- ❑ Prostřednictvím atributu `xsi:type`
- ❑ V instanci dokumentu explicitně specifikujeme datový typ elementu
  - Odvozený restrikcí nebo rozšířením

```
<xs:element  
  name="Publikace"  
  type="TypPublikace"/>
```

- ❑ Atribut block elementu `complexType`
  - restriction, extension, #all

```
<Publikace>  
  <Název>Babička</Název>  
  <Autor>B. Němcová</Autor>  
</Publikace>  
<Publikace xsi:type="TypKniha">  
  <Název>Dědeček</Název>  
  <Autor>J. Cimrman</Autor>  
  <ISBN>123-456-789</ISBN>  
</Publikace>
```

# Substituční skupiny (1)

---

- ❑ Rozšíření principu dědičnosti
  - ❑ Mechanismus explicitního povolení / zakázání vzájemné substituce elementů
  - ❑ Myšlenka: Elementy jsou přiřazeny do substituční skupiny vedoucího elementu určené jeho názvem
    - Elementy lze substituovat za vedoucí element
  - ❑ Podmínky:
    - Všechny elementy musí být globálně definované
    - Elementy musí mít stejný datový typ jako vedoucí element nebo typy z něj odvozené
    - Relace „být v substituční skupině“ je tranzitivní
-

# Substituční skupiny (2) – příklad

---

```
<xs:element name="Publikace" type="TypPublikace"/>
<xs:element name="Kniha" type="TypKniha"
    substitutionGroup="Publikace"/>
<xs:element name="Časopis" type="TypČasopis"
    substitutionGroup="Publikace"/>

<xs:element name="Knihovna">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Publikace" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

# Substituční skupiny (3) – příklad

---

```
<Knihovna>
  <Publikace>
    <Název>Babička</Název>
    <Autor>B. Němcová</Autor>
  </Publikace>
  <Kniha>
    <Název>Dědeček</Název>
    <Autor>J. Cimbura</Autor>
    <ISBN>123-456-789</ISBN>
  </Kniha>
  <!-- další elementy Publikace, Kniha nebo Časopis -->
</Knihovna>
```

---



# Substituční skupiny (4)

---

- Vlastnosti určují atributy elementu element
    - substitutionGroup – název globálně definovaného elementu
    - abstract – příznak abstraktního elementu
      - Za element je vždy nutné substituovat
    - final – blokáce umístění prvků do substituční skupiny vedoucího elementu
      - extension, restriction, #all
      - Implicitní hodnota: atribut **finalDefault** elementu schema
    - block – blokáce substituovatelnosti za vedoucí element
      - substitution, extension, restriction, #all
      - Implicitní hodnota: atribut **blockDefault** elementu schema
-

# Externí schémata

## – element include (1)

---

- ❑ Vložení schématu se stejným / žádným cílovým jmenným prostorem
    - Prvky vkládaného schématu se stávají součástí aktuálního (vytvářeného) cílového jmenného prostoru
  - ❑ Atributy:
    - id – jednoznačný identifikátor elementu include
    - schemaLocation – URI vkládaného schématu
  - ❑ Obsah:
    - (annotation?)
-

# Externí schémata

## – element include (2)

---

```
<xs:schema ...>  
  <!-- vložení prvků externího schématu -->  
  <xs:include schemaLocation="mojeSchema1.xsd"/>  
  
  <!-- definice dalších prvků schématu -->  
</xs:schema>
```

# Externí schémata

## – element import (1)

---

- Import prvků schématu s lib. jmenným prostorem
    - Globálně definované prvky lze využívat při definici prvků aktuálního schématu
  - Atributy:
    - id – jednoznačný identifikátor elementu import
    - schemaLocation – URI importovaného schématu
    - namespace – identifikátor cílového jmenného prostoru importovaného schématu
      - Je-li specifikován, odkazujeme se na prvky kvalifikovaným jménem
  - Obsah:
    - (annotation?)
-

# Externí schémata

## – element import (2)

---

```
<xs:schema ...  
    xmlns:typy="http://www.priklady.cz/xml/mojeSchema2">  
    <!-- import prvků externího schématu -->  
    <xs:import  
        namespace="http://www.priklady.cz/xml/mojeSchema2"/>  
  
    <element name="ElementSEexternímTypem"  
        type="typy:ExterníTyp"/>  
</xs:schema>
```

---

# Externí schémata

## – element redefine (1)

---

- Předefinování existujícího prvku
    - Jednoduchý typ – restrikce původního typu
    - Složený typ – restrikce / rozšíření původního typu
    - Skupina elementů
      - Nadmnožina – obsahuje původní prostřednictvím ref atributu
      - Podmnožina – omezení minOccurs a maxOccurs prvků
    - Skupina atributů
      - Nadmnožina – obsahuje původní prostřednictvím ref atributu
      - Podmnožina – modifikace atributu use prvků skupiny
-

# Externí schémata

## – element redefine (2)

---

- Atributy:
    - id – jednoznačný identifikátor elementu redefine
    - schemaLocation – URI schématu z něhož prvek pochází
  - Obsah:
    - (annotation | (simpleType | complexType | group | attributeGroup))\*
-

# Externí schémata

## – element redefine (3)

---

```
<xs:redefine schemaLocation="mojeSchema2.xsd">
  <xs:complexType name="ExterníTyp">
    <xs:complexContent>
      <xs:extension base="ExterníTyp">
        <xs:sequence>
          <xs:element name="NovýElement"
                      type="xs:string"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:redefine>
```

---



# Omezení identity (1)

---

- ☐ ID, IDREF, IDREFS – převzato z DTD
    - Unikátnost hodnoty atributu v rámci celého dokumentu
    - Odkaz(y) na unikátní hodnoty
  - ☐ Typy omezení identity
    - Klíč – povinná, nenulová, jednoznačná hodnota (key)
    - Unikátnost – nenulová, jednoznačná hodnota, je-li v dokumentu uvedena (unique)
    - Odkaz na klíčovou / unikátní hodnotu (keyref)
      - ☐ Obdoba cizích klíčů z databází
  - ☐ Založeny na jazyku XPath (podmnožina)
-

# Omezení identity (2)

---

## ☐ Atributy:

- id – jednoznačný identifikátor elementu unique / key / keyref
- name – název omezení identity
- refer – odkaz na existující omezení unique / key
  - ☐ Pouze pro element keyref

## ☐ Obsah:

- (annotation?, (selector, field+))
-

# Omezení identity (3)

---

- ☐ Element selector – množina elementů v rámci níž omezení platí
  - ☐ Element(y) field – množina podelementů nebo atributů (relativně k množině selector) splňující omezení
    - Množiny vybíráme pomocí XPath dotazů
    - Atributy:
      - ☐ id – jednoznačný identifikátor elementu selector / field
      - ☐ xpath – výraz jazyka XPath
    - Obsah:
      - ☐ (annotation?)
-

# XPath

---

- ❑ Dotazy připomínají cestu k souboru
- ❑ Jednotlivé kroky = elementy / atributy (příznak @)
- ❑ Speciální prvky:
  - . – aktuální element
  - / – přímý podelement nebo atribut
  - // – potomek v libovolné hloubce
  - \* – potomek o úroveň níž

```
Selector ::= PathS ( '|' PathS ) *  
Field    ::= PathF ( '|' PathF ) *  
PathS    ::= ('.//')? Step ( '/' Step ) *  
PathF    ::= ('.//')? ( Step '/' ) * ( Step | '@' NameTest )  
Step     ::= '.' | NameTest  
NameTest ::= QName | '*' | NCName ':' '*'
```

# Element unique

---

```
<xs:element name="Knihovna">
  ...
  <xs:element name="Kniha" maxOccurs="unbounded">
    ...
    <xs:element name="ISBN" type="xs:string"
                  minOccurs="0"/>
    ...
  </xs:element>

  <xs:unique name="UnikátníISBN">
    <xs:selector xpath="./Kniha"/>
    <xs:field    xpath="./ISBN"/>
  </xs:unique>
</xs:element>
```

# Element key

---

```
<xs:element name="Knihovna">
  ...
  <xs:element name="Kniha" maxOccurs="unbounded">
    ...
    <xs:element name="ISBN" type="xs:string"/>
    ...
  </xs:element>

  <xs:key name="PrimárníKlíč">
    <xs:selector xpath="./Kniha"/>
    <xs:field xpath="./ISBN"/>
  </xs:key>
</xs:element>
```

---

# Element keyref (1)

---

```
<xs:element name="Knihovna">
  ... <!-- Předchozí definice elementu a klíče --> ...
  <xs:element name="Autor" maxOccurs="unbounded">
    ...
    <xs:element name="NejlepšíKniha">
      ...
      <xs:element name="ISBN" type="xs:string"/>
      ...
    </xs:element>
    ...
  </xs:element>

  <xs:keyref name="CizíKlíč" refer="PrimárníKlíč">
    <xs:selector xpath="./Autor/NejlepšíKniha"/>
    <xs:field xpath="./ISBN"/>
  </xs:keyref>
</xs:element>
```

# Element keyref (2)

---

```
<Knihovna>
```

```
<!-- knihy v knihovně -->
```

```
<Kniha>
```

```
<ISBN>111-222-333</ISBN>
```

```
<Název>B. Němcová - Babička</Název>
```

```
</Kniha>
```

```
<Kniha>
```

```
<ISBN>444-555-666</ISBN>
```

```
<Název>B. Němcová - Divá Bára</Název>
```

```
</Kniha>
```

```
<Kniha>
```

```
<ISBN>123-456-789</ISBN>
```

```
<Název>J. Cimrman - Dědeček</Název>
```

```
</Kniha>
```

```
...
```



# Element keyref (3)

---

...

```
<!-- informace o autorech knih -->
```

```
<Autor>
```

```
  <Jméno>Božena Němcová</Jméno>
```

```
  <NejlepšíKniha>
```

```
    <ISBN>111-222-333</ISBN>
```

```
    <PočetVydání>123<PočetVydání>
```

```
  </NejlepšíKniha>
```

```
</Autor>
```

```
<Autor>
```

```
  <Jméno>Jára Cimrman</Jméno>
```

```
  <NejlepšíKniha>
```

```
    <ISBN>123-456-789</ISBN>
```

```
    <PočetVydání>0<PočetVydání>
```

```
  </NejlepšíKniha>
```

```
</Autor>
```

```
</Knihovna>
```

# Zástupci (1) – wildcards

---

- Možnost vložit na dané místo „libovolný“ prvek
  - Element anyAttribute
    - Atributy:
      - id – jednoznačný identifikátor elementu anyAttribute
      - namespace – jmenný prostor přípustných prvků
        - seznam URI jmenných prostorů
        - ##any – libovolný známý jmenný prostor
        - ##targetNamespace – cílový jmenný prostor
        - ##other – jiný než cílový jmenný prostor
        - ##local – žádný specifický jmenný prostor
-

# Zástupci (2)

---

- ☐ processContents – způsob validace obsahu
    - strict – povinná validace
    - lax – validace pouze pokud je nalezen odpovídající prvek
    - skip – žádná validace
  - ☐ notNamespace – seznam jmenných prostorů, jejichž prvky se nesmí na daném místě vyskytovat
    - ##targetNamespace, ##local
    - verze 1.1
  - ☐ notQName – seznam názvů elementů resp. atributů, jejichž prvky se nesmí na daném místě vyskytovat
    - verze 1.1
  - Obsah:
    - ☐ (annotation?)
-

# Zástupci (3)

---

## ☐ Element any

### ■ Atributy:

- ☐ id, nameSpace, processContents, notNamespace, notQName – stejné

- ☐ minOccurs – minimální počet výskytů elementu any

- ☐ maxOccurs – maximální počet výskytů elementu any

### ■ Obsah:

- ☐ (annotation?)

---

# Zástupci (3)

---

```
<xs:complexType name="LibovolnýHtmlText">
  <xs:sequence>
    <xs:any namespace="http://www.w3.org/1999/xhtml"
      minOccurs="1"
      maxOccurs="unbounded"
      processContents="strict"/>
  </xs:sequence>
</xs:complexType>
```

---

# Notace (1)

---

- ☐ Odkaz na externí program
    - Obdoba z DTD
    - Významu rozumí až program, který schéma zpracovává
  - ☐ Element notation
    - Atributy:
      - ☐ id – jednoznačný identifikátor elementu notation
      - ☐ name – název notace
      - ☐ system – URI spustitelného programu
      - ☐ public – libovolný identifikátor spustitelného programu
    - Obsah:
      - ☐ (annotation?)
-

# Notace (2)

---

- ❑ Odkazy na notace – datový typ NOTATION
  - Nelze používat přímo instance typu, pouze typy odvozené restrikcí enumeration
  - Pro každou hodnotu musí existovat element notation

```
<xs:notation name="EPS" system="C:\eps\eps.exe"/>

<xs:attribute name="Program">
  <xs:simpleType>
    <xs:restriction base="xs:NOTATION">
      <xs:enumeration value="EPS">
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
```

# Anotace (1)

---

- ❑ Slouží pro dokumentaci / komentování schématu
    - Komentáře lze využít také
    - Součást libovolného prvku schématu
  - ❑ Element annotation
    - Atributy:
      - ❑ id – jednoznačný identifikátor elementu annotation
    - Obsah:
      - ❑ (appinfo | documentation)\*
  - ❑ Element appinfo
    - Informace určená pro nadřazený program
    - Atributy:
      - ❑ source – URI externího souboru v němž je informace uložena
-



# Anotace (2)

---

- Element documentation
  - Informace určená pro člověka
  - Atributy:
    - source – URI externího souboru v němž je informace uložena
    - xml:lang – Jazyk textu, je-li uveden přímo v dokumentu

```
<xs:annotation>  
  <xs:documentation xml:lang="cs">  
    Toto je příklad anotace určené pro člověka.  
  </xs:documentation>  
</xs:annotation>
```

---

# XML dokument vs. XML schéma

---

XML dokument	XML schéma
Elementy	Elementy
Obsah elementů	Jednoduché / složené datové typy
Atributy	Atributy
Hodnoty atributů	Jednoduché datové typy
Entity	Nelze omezovat
Notace	Stejný princip v syntaxi XML Schema
Komentáře	Nelze omezovat
PI	Nelze omezovat
XML deklarace	Nelze omezovat
Jmenné prostory	Nelze omezovat

# DTD vs. XML schéma (1)

---

DTD	XML schéma
DOCTYPE	Jiný přístup – jmenné prostory
Externí / interní	Jiný přístup – elementy include, import, redefine
ELEMENT	Element element
#PCDATA	Jednoduchý datový typ (textový) / složený datový typ se smíšeným obsahem (atribut mixed="true")
ANY	Silnější nástroj – element any
EMPTY	Složený datový typ bez podelementů a smíšeného obsahu
Elementový obsah	Element complexType

# DTD vs. XML schéma (2)

---

DTD	XML schéma
,	Element sequence
	Element choice
?	Atributy minOccurs a maxOccurs
+	Atributy minOccurs a maxOccurs
*	Atributy minOccurs a maxOccurs
( )	Element group
ATTLIST	Element attribute / attributeGroup
CDATA	Jednoduchý datový typ (textový)
ID	Jednoduchý datový typ ID
IDREF, IDREFS	Jednoduché datové typy IDREF, IDREFS

# DTD vs. XML schéma (3)

---

DTD	XML schéma
NOTATION	Jednoduchý datový typ NOTATION
NMTOKEN, NMTOKENS	Jednoduché datové typy NMTOKEN, NMTOKENS
ENTITY, ENTITIES	Jednoduché datové typy ENTITY, ENTITIES
Seznam hodnot	Element enumeration
Implicitní hodnoty atributů	Atribut default elementu attribute
#FIXED hodnoty	Atribut fixed elementu attribute
#REQUIRED a #IMPLIED hodnoty	Atribut use (required, optional) elementu attribute
ENTITY	Nelze deklarovat

# DTD vs. XML schéma (4)

---

DTD	XML schéma
Parametrické entity	Nepodporuje – jiné mechanismy
IGNORE, INCLUDE	Nejsou podporovány
NOTATION	Lze deklarovat
Komentáře	Lze použít + další mechanismy (podelement documentation elementu annotation)
PI	Lze použít + další mechanismy (podelement appinfo elementu annotation)

---

# Konec...

---