

Y36XML – Technologie XML

Přednáší:

Irena Mlýnková (mlynkova@ksi.mff.cuni.cz)

Martin Nečaský (necasky@ksi.mff.cuni.cz)

ZS 2009

Stránka přednášky:

<http://www.ksi.mff.cuni.cz/~mlynkova/Y36XML/>

Osnova předmětu

- ❑ Úvod do principů formátu XML, přehled XML technologií, jazyk 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
-

Dotazovací jazyky nad XML daty

- Cíle: dotazování, pohledy, transformace, případně aktualizace XML dat
 - Od r. 1998 XML-QL, XQL, ...
 - Vývoj v konsorciu W3C se ustálil/pokračuje v jazycích XSLT 1.0, XSLT 2.0, XPath 1.0, XPath 2.0, XQuery 1.0
 - XSLT je jazyk pro transformace, využívá XPath, zápis transformací hodně využívá XML
 - XQuery vhodnější pro dotazování – uživatelsky orientovaná syntaxe
 - Pz.: XPath 2.0 \subset XQuery
-

Co je XPath?

- ❑ XPath je základní jazyk pro dotazování nad XML dokumenty
 - Výběr částí XML dokumentů
 - ❑ Základní myšlenka se podobá navigaci v systému souborů
 - ❑ Notace XPath není vyjádřena v XML
 - ❑ Na XPath je založen XSLT, XPointer, XQuery
-

Model XML dat v XPath

```
<?xml version="1.0"?>
<!DOCTYPE objednavka SYSTEM "objednavka.dtd">
<objednavka datum="10/10/2008" stav="expedovana">
  <zakaznik cislo="C992">Martin Nečaský</zakaznik>
  <polozky>
    <polozka kod="48282811">
      <mnozstvi>5</mnozstvi>
      <cena>22</cena>
    </polozka>
    <polozka kod="929118813">
      <mnozstvi>1</mnozstvi>
      <cena>91934</cena>
      <barva>modra</barva>
    </polozka>
  </polozky>
</objednavka>
```

Model XML dat v XPath

```
<?xml version="1.0"?>
<!DOCTYPE objednavka SYSTEM "objednavka.dtd">
<objednavka datum="10/10/2008" stav="expedovana">
  <zakaznik cislo="C992">Martin Nečaský</zakaznik>
  <polozky>
    <polozka kod="48282811">
      <mnozstvi>5</mnozstvi>
      <cena>22</cena>
    </polozka>
    <polozka kod="929118813">
      <mnozstvi>1</mnozstvi>
      <cena>91934</cena>
      <barva>modra</barva>
    </polozka>
  </polozky>
</objednavka>
```

document

Model XML dat v XPath

```
<?xml version="1.0"?>
<!DOCTYPE objednavka SYSTEM "objednavka.dtd">
<objednavka datum="10/10/2008" stav="expedovana">
  <zakaznik cislo="C992">Martin Nečaský</zakaznik>
  <polozky>
    <polozka kod="48282811">
      <mnozstvi>5</mnozstvi>
      <cena>22</cena>
    </polozka>
    <polozka kod="929118813">
      <mnozstvi>1</mnozstvi>
      <cena>91934</cena>
      <barva>modra</barva>
    </polozka>
  </polozky>
</objednavka>
```

document

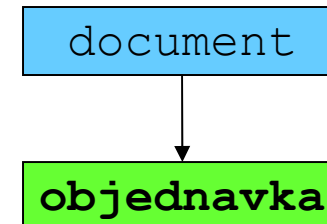
Model XML dat v XPath

```
<?xml version="1.0"?>
<!DOCTYPE objednavka SYSTEM "objednavka.dtd">
<objednavka datum="10/10/2008" stav="expedovana">
  <zakaznik cislo="C992">Martin Nečaský</zakaznik>
  <polozky>
    <polozka kod="48282811">
      <mnozstvi>5</mnozstvi>
      <cena>22</cena>
    </polozka>
    <polozka kod="929118813">
      <mnozstvi>1</mnozstvi>
      <cena>91934</cena>
      <barva>modra</barva>
    </polozka>
  </polozky>
</objednavka>
```

document

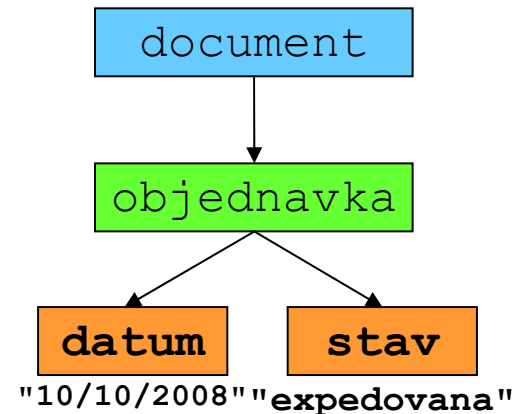
Model XML dat v XPath

```
<?xml version="1.0"?>
<!DOCTYPE objednavka SYSTEM "objednavka.dtd">
<objednavka datum="10/10/2008" stav="expedovana">
  <zakaznik cislo="C992">Martin Nečaský</zakaznik>
  <polozky>
    <polozka kod="48282811">
      <mnozstvi>5</mnozstvi>
      <cena>22</cena>
    </polozka>
    <polozka kod="929118813">
      <mnozstvi>1</mnozstvi>
      <cena>91934</cena>
      <barva>modra</barva>
    </polozka>
  </polozky>
</objednavka>
```



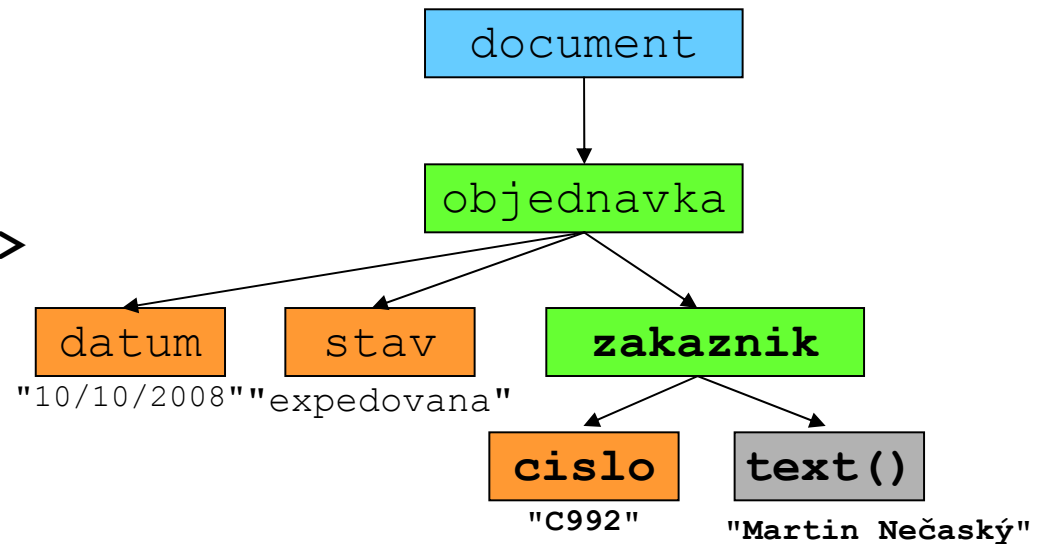
Model XML dat v XPath

```
<?xml version="1.0"?>
<!DOCTYPE objednavka SYSTEM "objednavka.dtd">
<objednavka datum="10/10/2008" stav="expedovana">
  <zakaznik cislo="C992">Martin Nečaský</zakaznik>
  <polozky>
    <polozka kod="48282811">
      <mnozstvi>5</mnozstvi>
      <cena>22</cena>
    </polozka>
    <polozka kod="929118813">
      <mnozstvi>1</mnozstvi>
      <cena>91934</cena>
      <barva>modra</barva>
    </polozka>
  </polozky>
</objednavka>
```

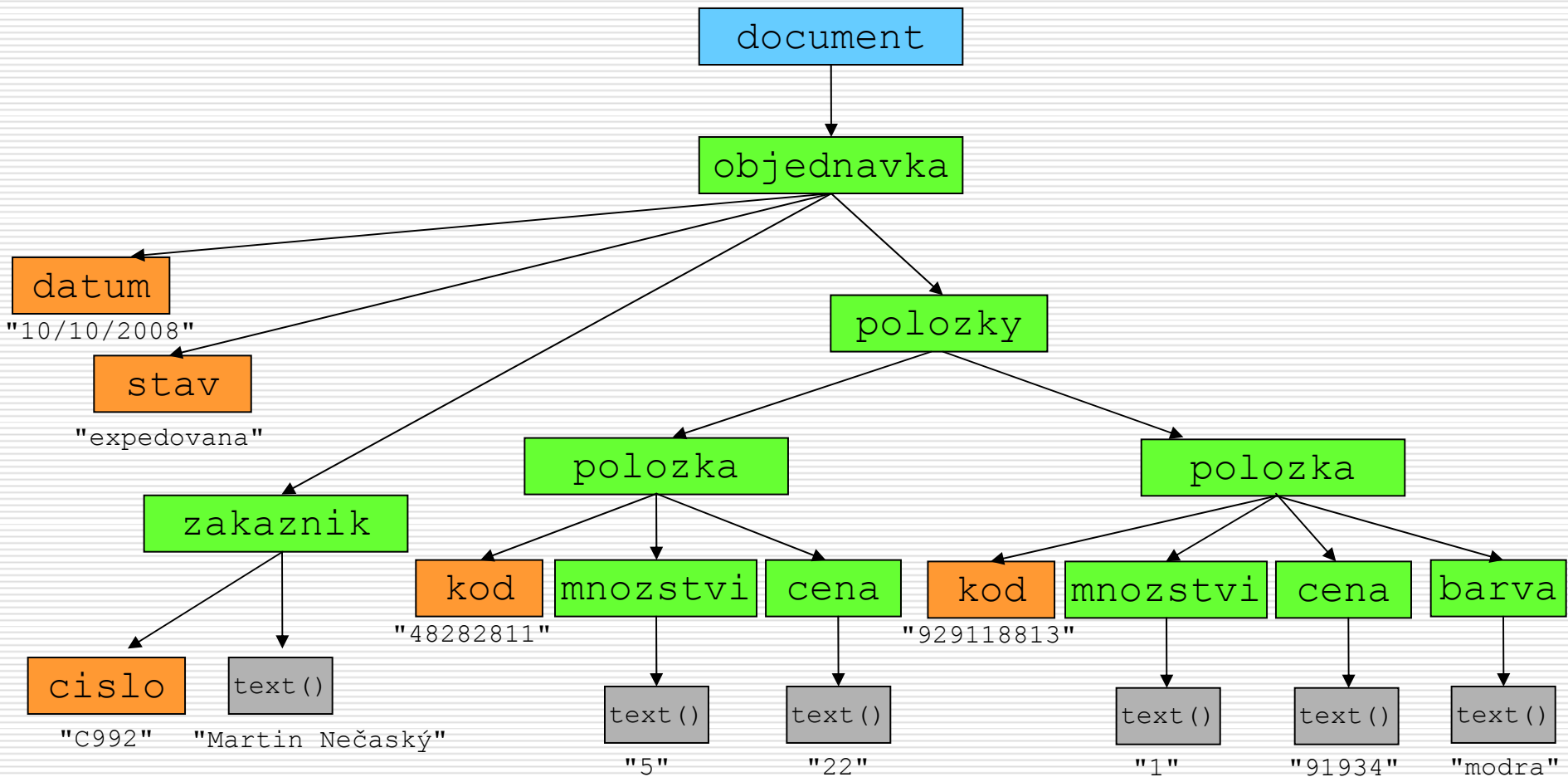


Model XML dat v XPath

```
<?xml version="1.0"?>
<!DOCTYPE objednavka SYSTEM "objednavka.dtd">
<objednavka datum="10/10/2008" stav="expedovana">
  <zakaznik cislo="C992">Martin Nečaský</zakaznik>
  <polozky>
    <polozka kod="48282811">
      <mnozstvi>5</mnozstvi>
      <cena>22</cena>
    </polozka>
    <polozka kod="929118813">
      <mnozstvi>1</mnozstvi>
      <cena>91934</cena>
      <barva>modra</barva>
    </polozka>
  </polozky>
</objednavka>
```



Model XML dat v XPath



Model XML dat v XPath

□ Typy uzlů v modelu

- kořenový uzel
- uzly elementů
- textové uzly
- atributové uzly
- uzly pro komentáře
- uzly proveditelných instrukcí
- uzly jmenných prostorů

□ Co tam není: sekce CDATA, odkazy na entity a DTD

Model XML dat v XPath

☐ Specifika

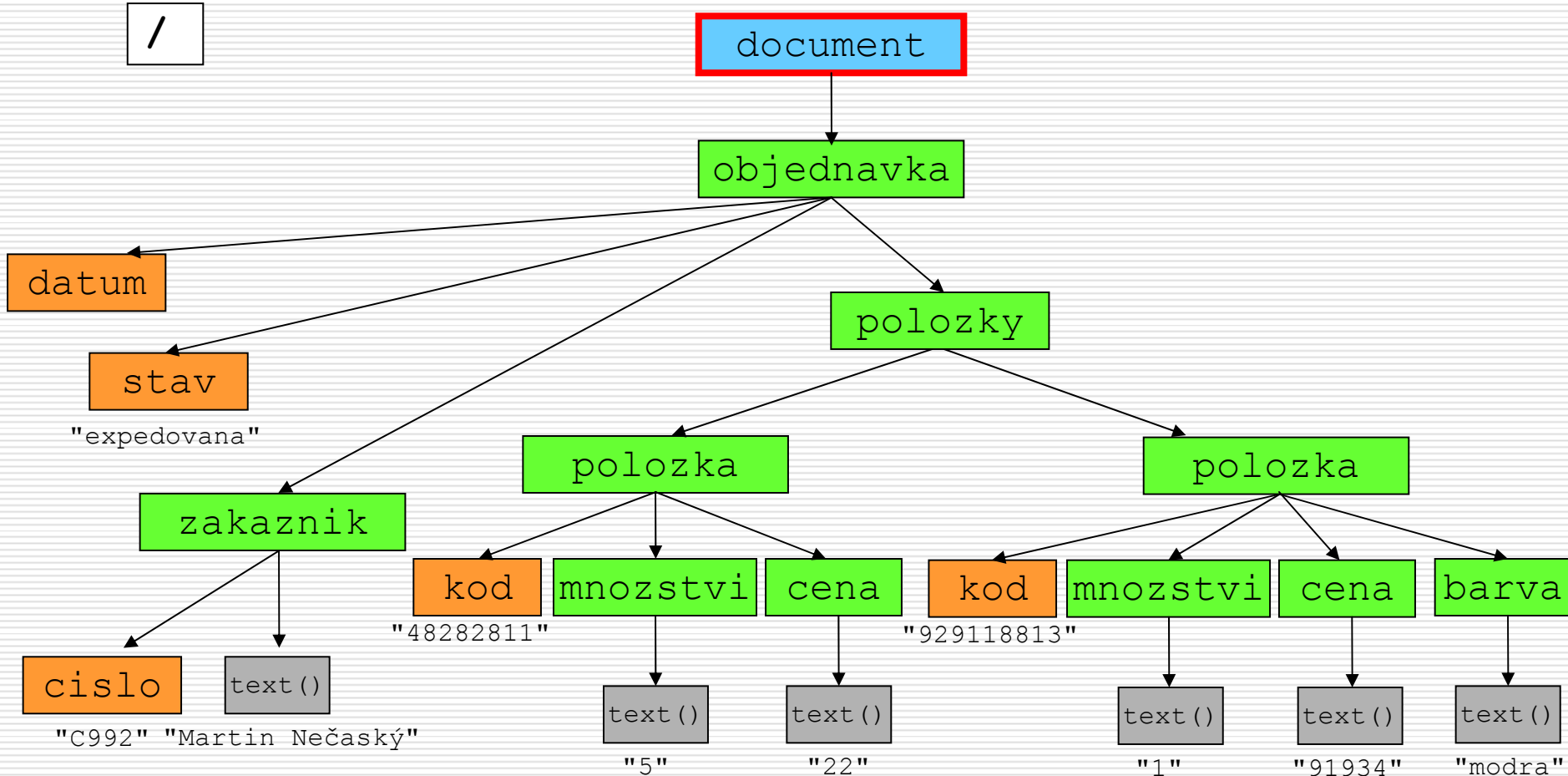
- Atributy nejsou zařazeny do seznamu dětí a potomků daného uzlu
 - Atributy ze jmenného prostoru xmlns jsou chápány jako speciální uzly
 - Kořenový uzel nereprezentuje kořenový element ale celý XML dokument
 - ☐ Kořenový element je reprezentován jako dítě kořenového uzlu
-

Výrazy v XPath

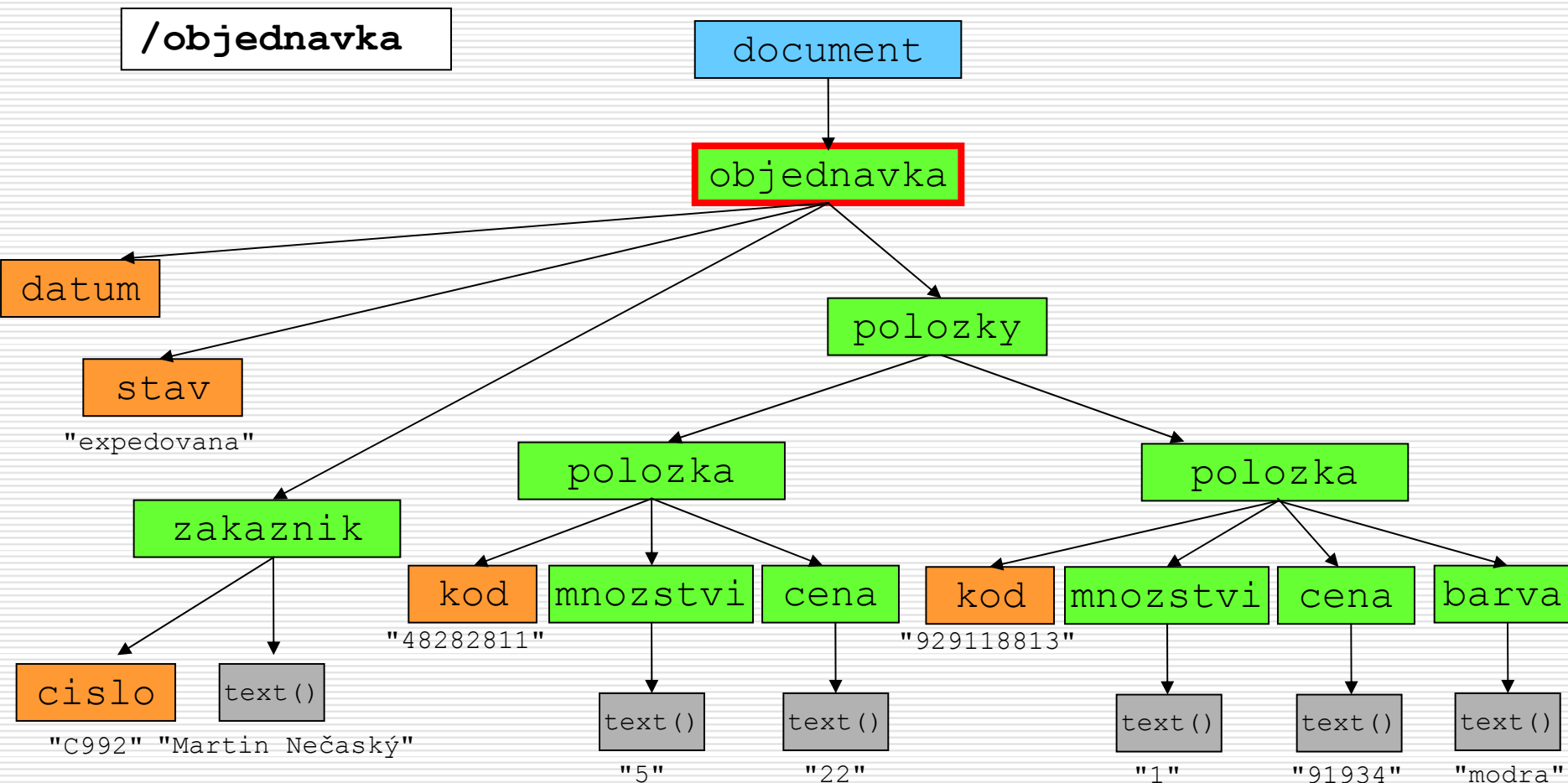
- ❑ Výrazem v jazyce XPath je cesta
 - ❑ Cesta se skládá z kroků
 - Absolutní cesta:
 - ❑ `/Krok1/Krok2/.../KrokN`
 - Relativní cesta:
 - ❑ `Krok1/Krok2/.../KrokN`
-

Výrazy v XPath - Příklady

/

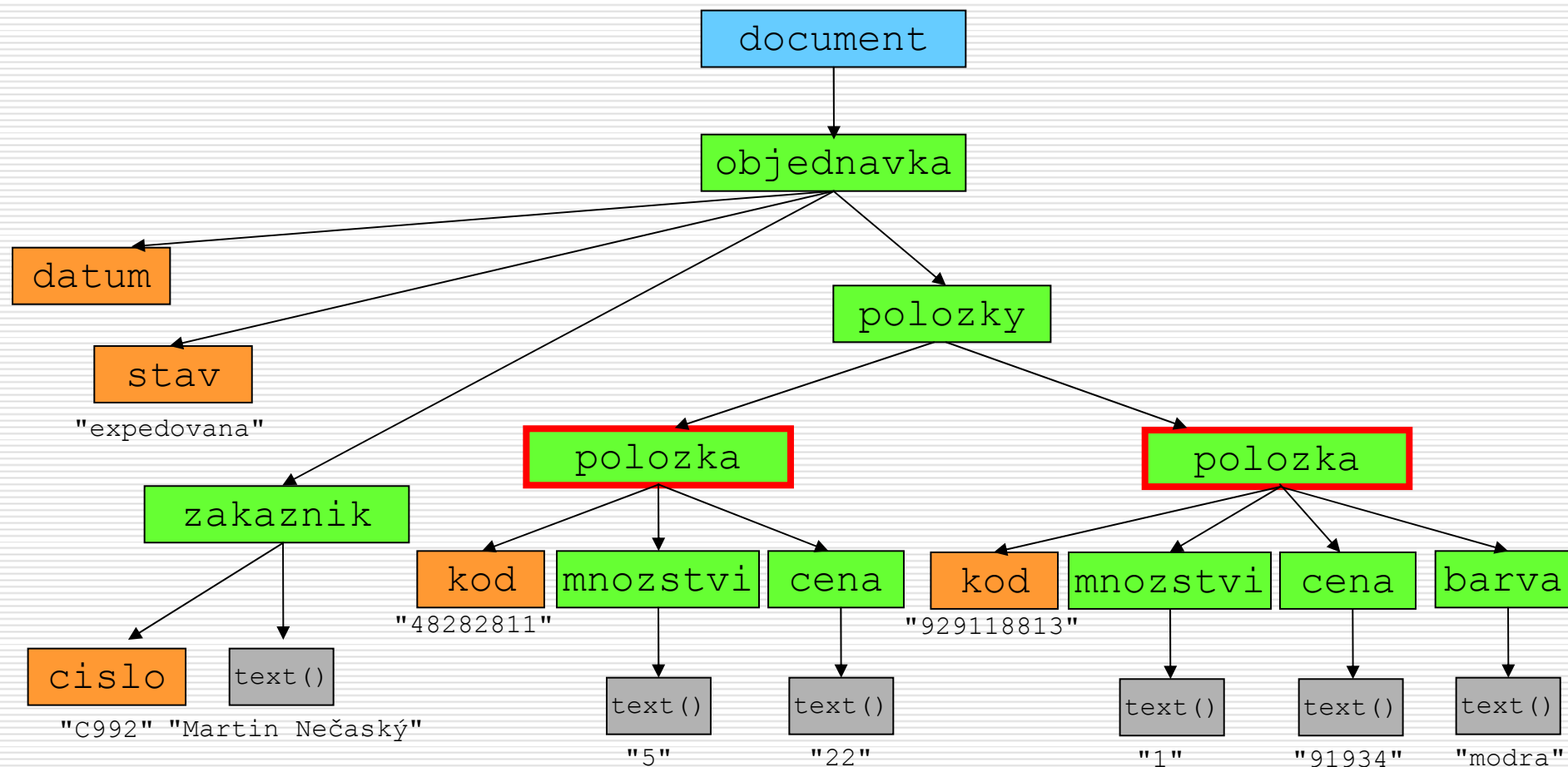


Výrazy v XPath - Příklady



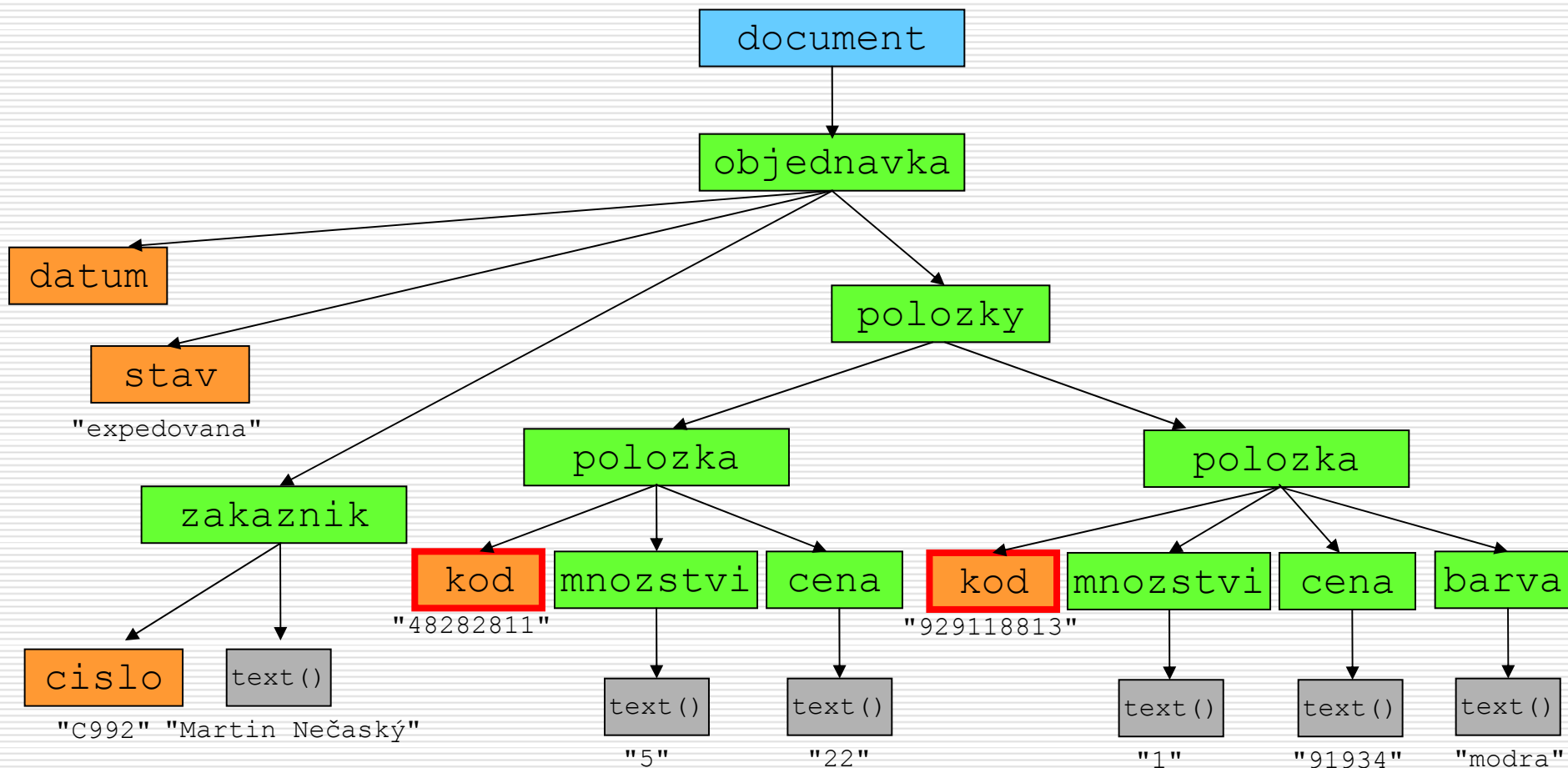
Výrazy v XPath - Příklady

`/objednavka/polozky/polozka`



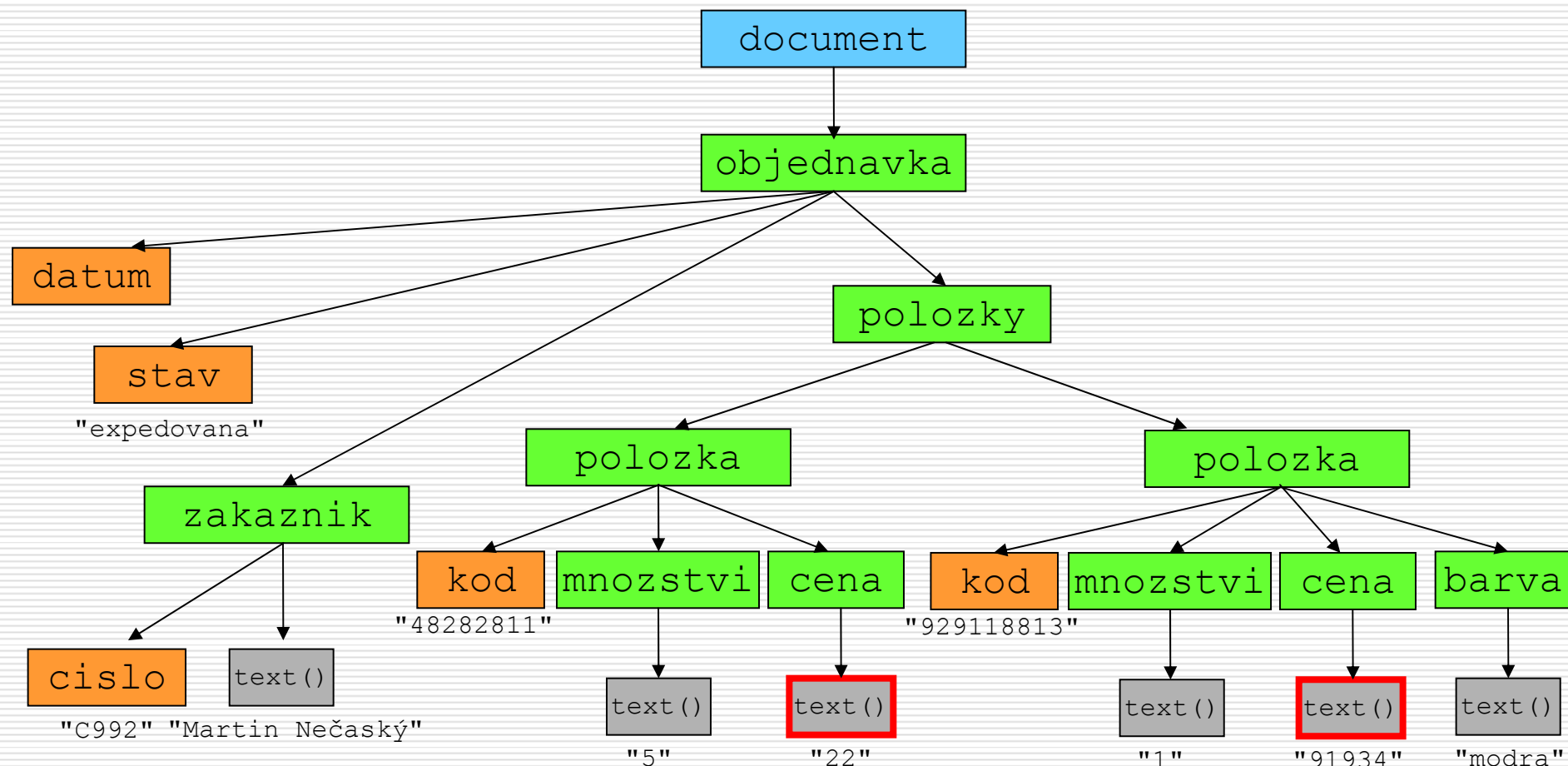
Výrazy v XPath - Příklady

`/objednavka/polozky/polozka/@kod`



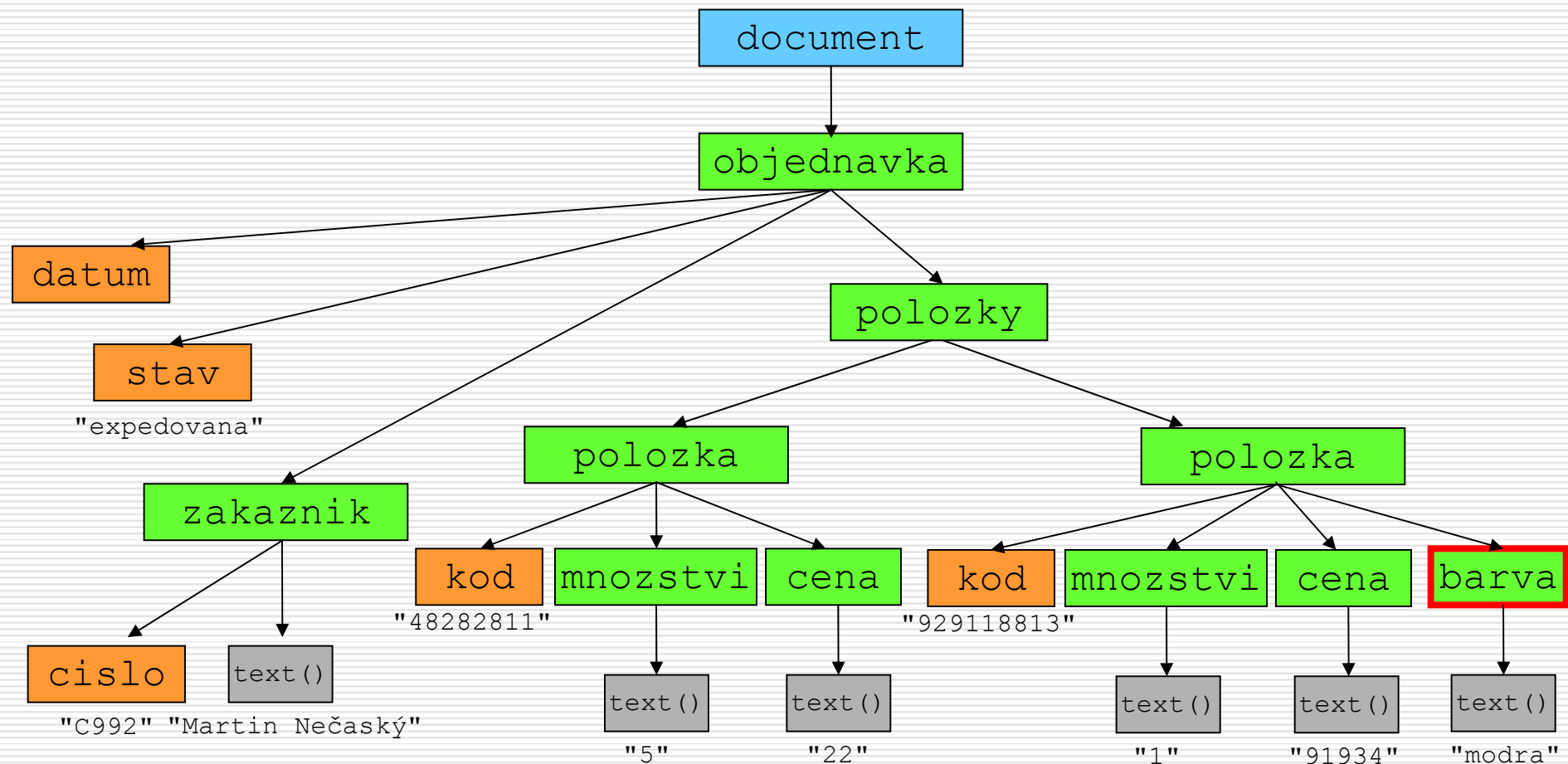
Výrazy v XPath - Příklady

`/objednavka/polozky/polozka/cena/text()`



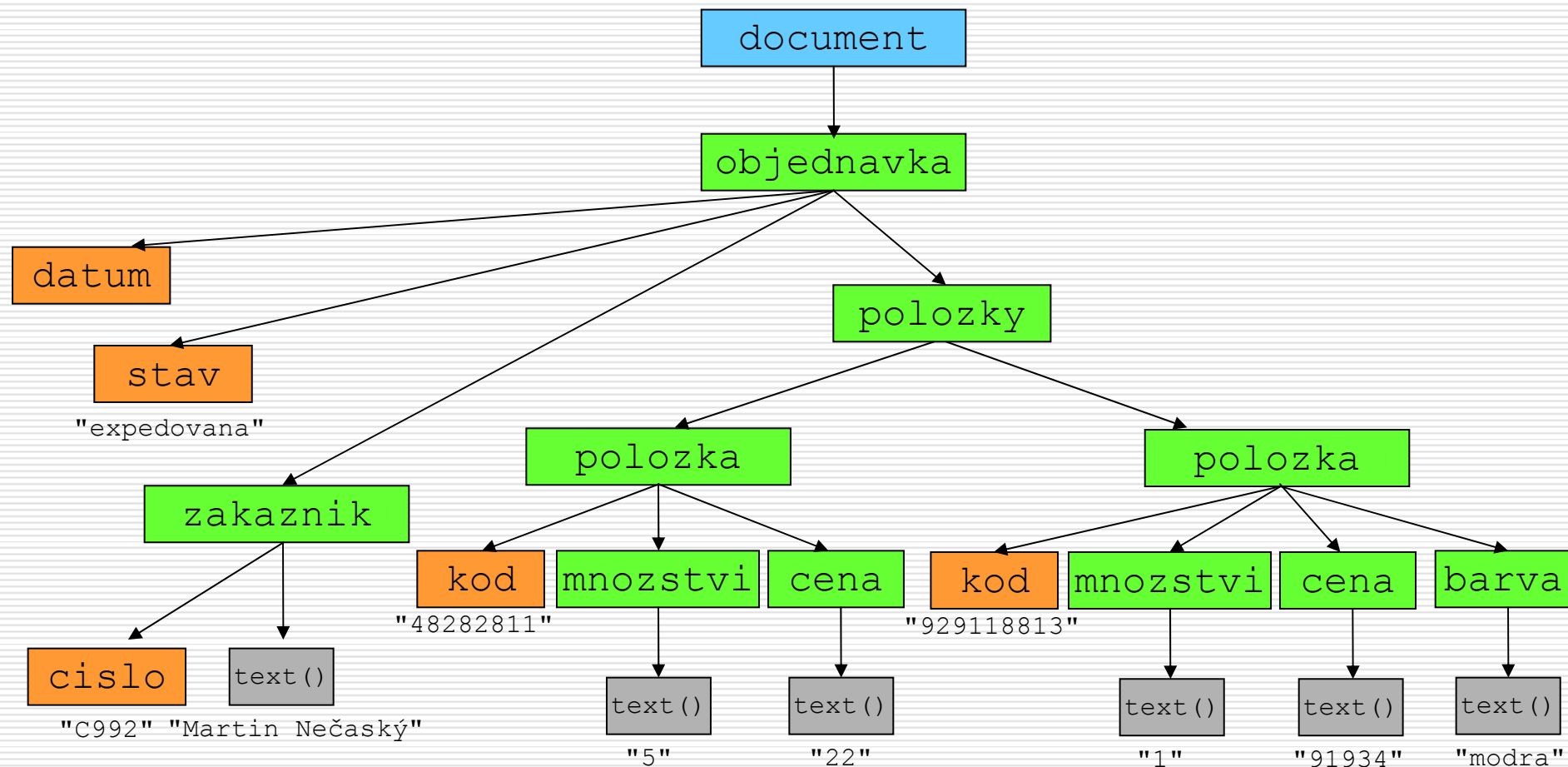
Výrazy v XPath - Příklady

`/objednavka/polozky/polozka/barva`



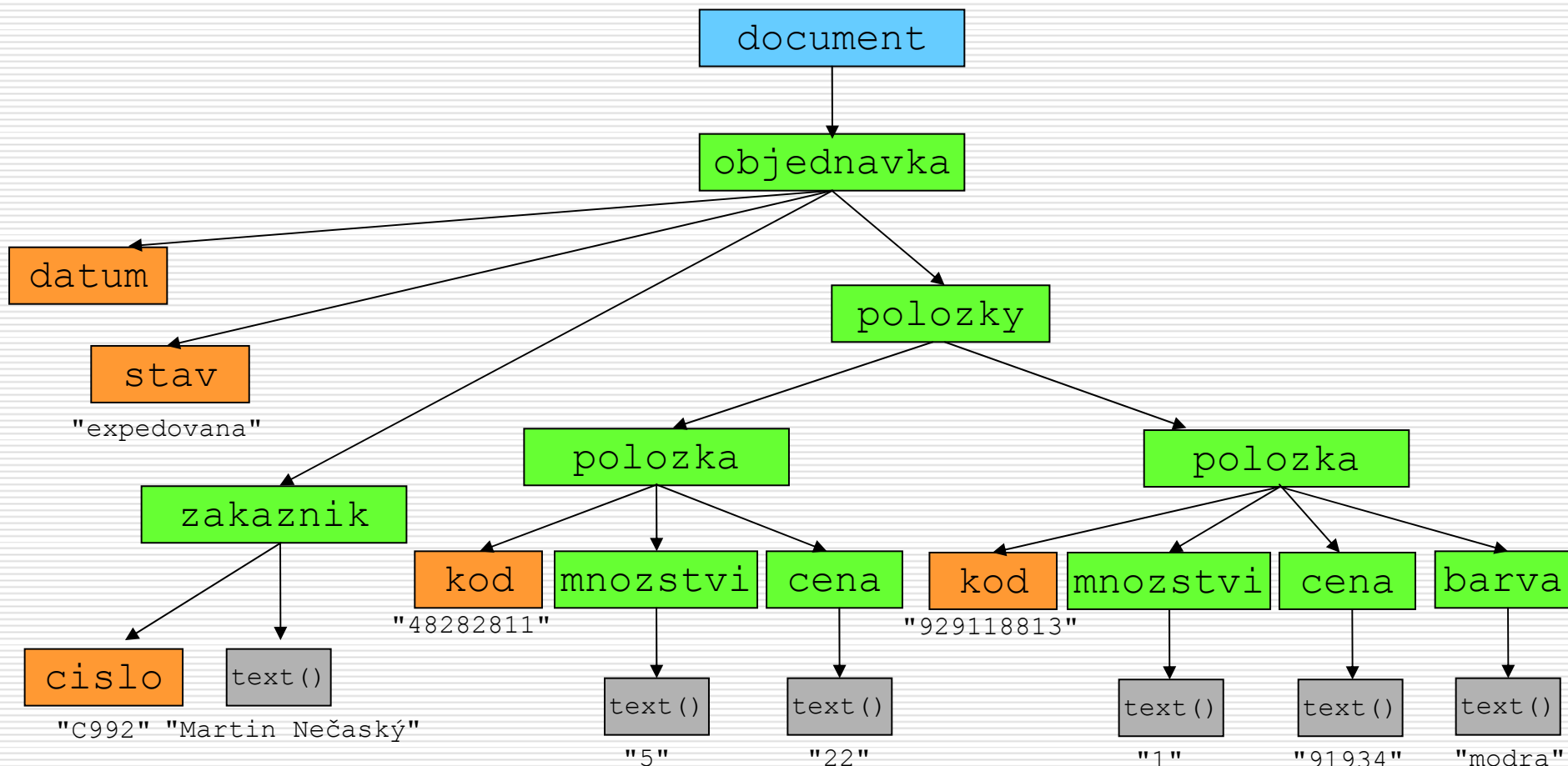
Výrazy v XPath - Příklady

`/objednavka/zakaznik/jmeno`



Výrazy v XPath - Příklady

`/objednavka/seznam-polozek/polozka`



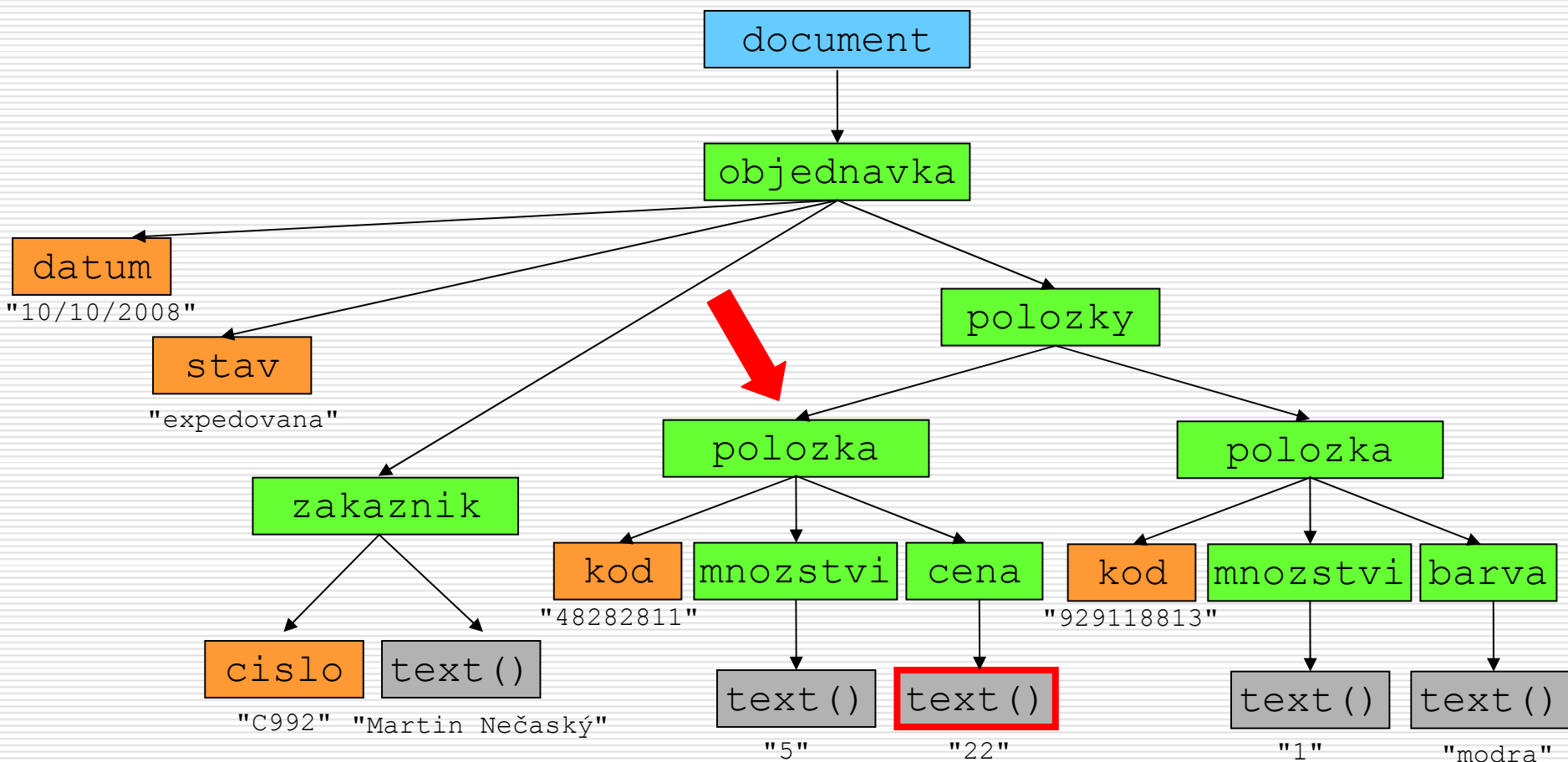
Výrazy v XPath - Příklady

`cena/text()`

- ❑ Relativní cesta sama o sobě nemůže být vyhodnocena
 - Nemá to žádný smysl, protože nevíme odkud máme začít
 - Vstupem musí být kromě samotné cesty také jeden a více uzlů v XML dokumentu, ze kterých máme vyhodnocení začít
 - tzv. Kontextová množina
-

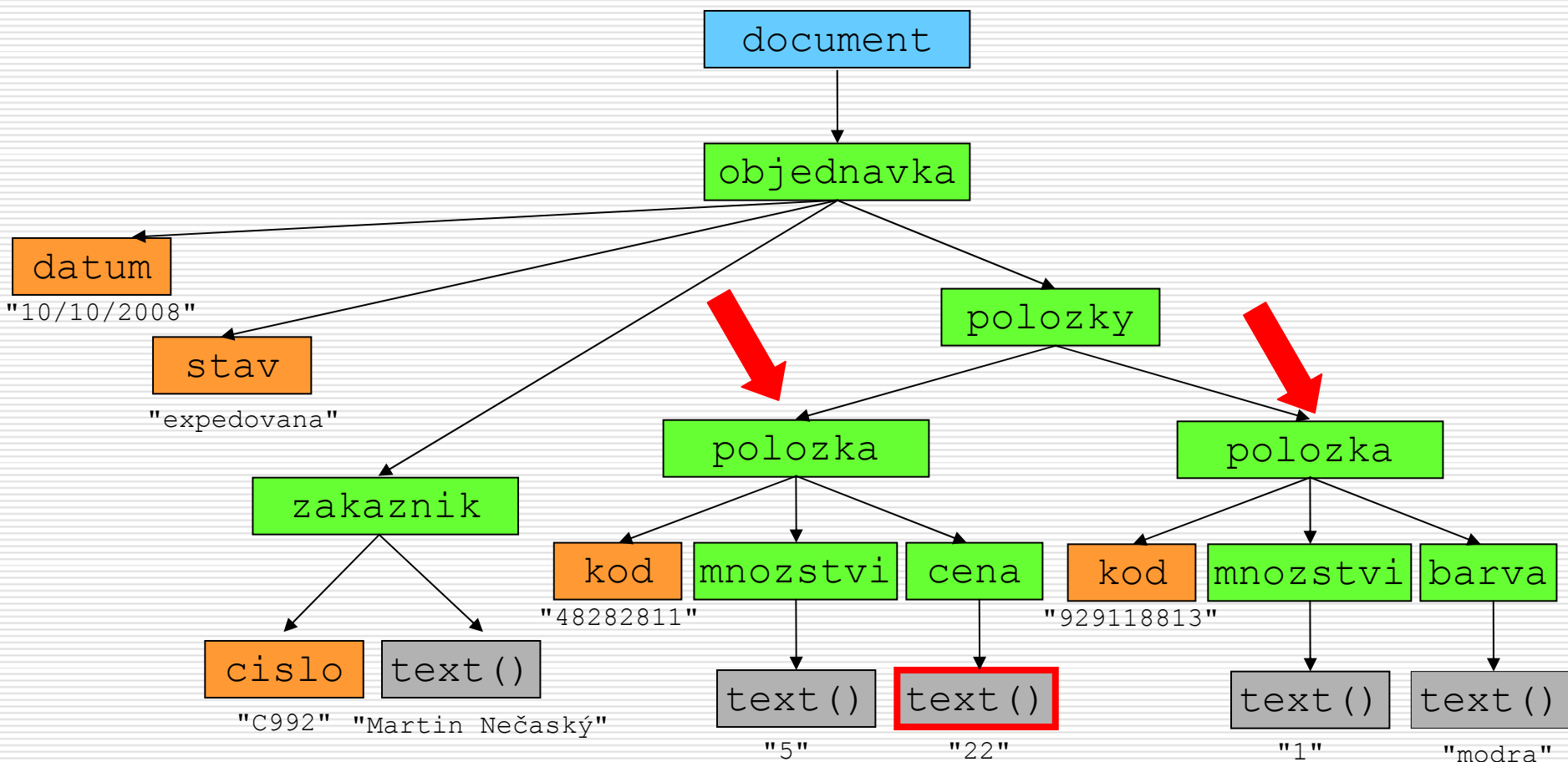
Výrazy v XPath - Příklady

`cena/text()`



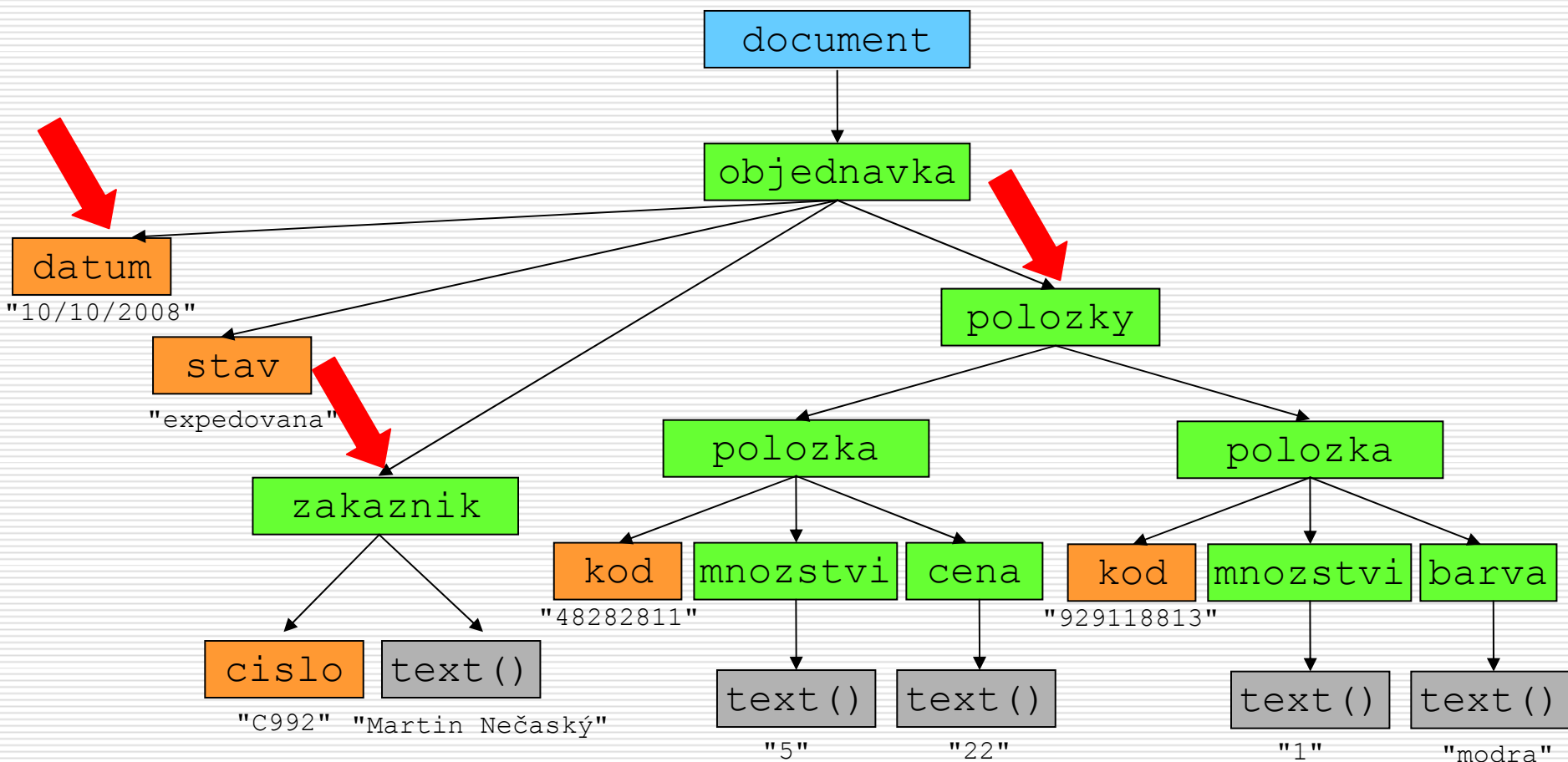
Výrazy v XPath - Příklady

`cena/text()`



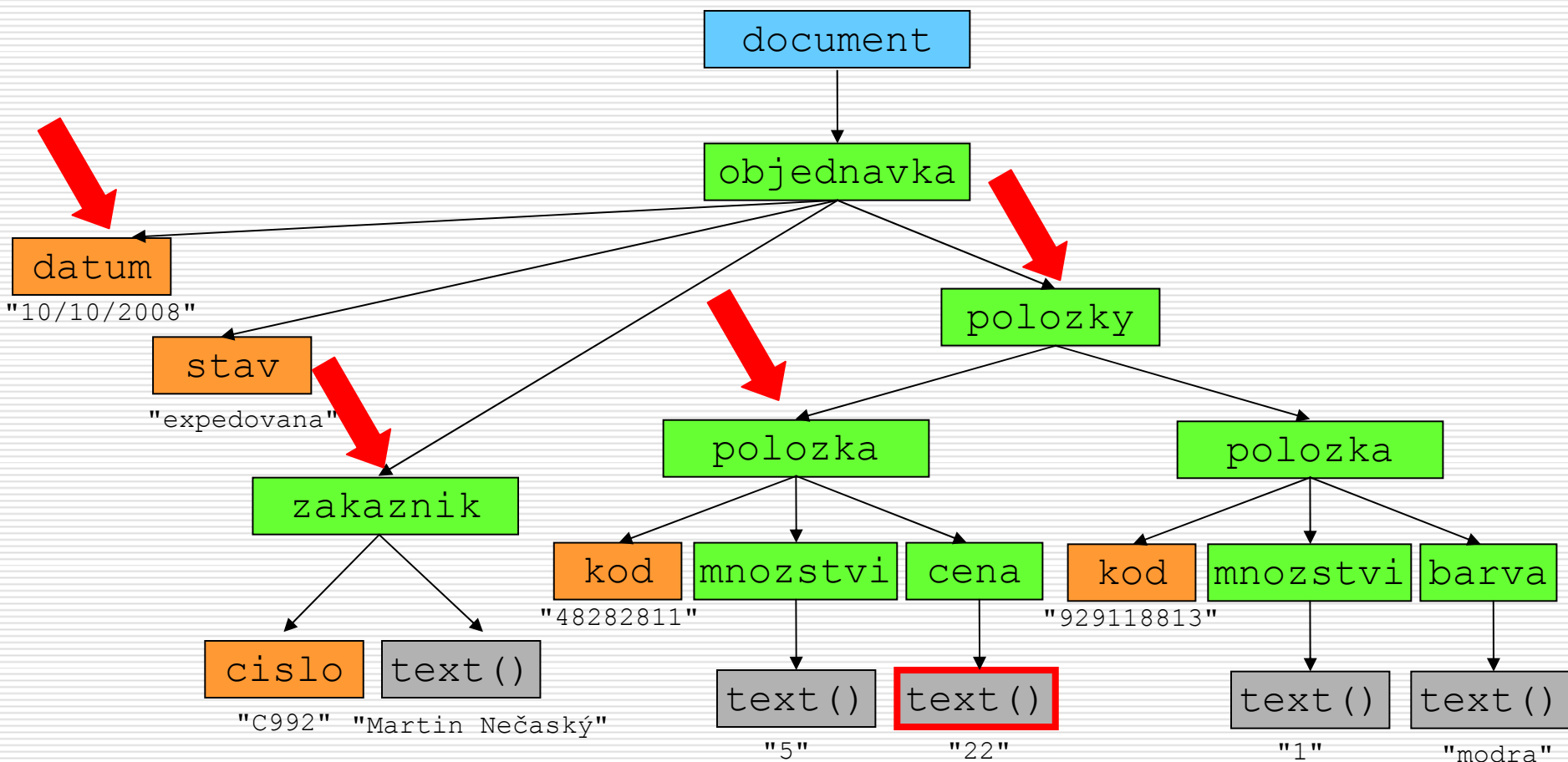
Výrazy v XPath - Příklady

`cena/text()`



Výrazy v XPath - Příklady

`cena/text()`



Vyhodnocení XPath výrazu

- Necht' P je XPath cesta, kde S je její první krok a P' je zbytek cesty
 - Necht' C značí kontextovou množinu uzlů pro vyhodnocení P
 - Pokud je P absolutní, pak C obsahuje pouze a jen kořenový uzel dokumentu
 - Pokud je P relativní, musí být C specifikována explicitně
 - Pokud je P prázdná, pak výsledkem vyhodnocení je C
 - Prázdná = 0 kroků
 - Jinak je P vyhodnocena vzhledem k C následovně:
 - $C' = \{ \}$
 - Pro každý uzel u z C vyhodnot' S a výsledek přidej do C'
 - Vyhodnot' P' vzhledem k C'
-

Vyhodnocení XPath výrazu

- Pozor, návratová hodnota cesty je množina uzlů
 - Množina je neuspořádaná
 - Tedy ani výsledek nelze očekávat uspořádaný (např. v pořadí, ve kterém se vyskytují uzly v XML dokumentu, viz. dále)
-

XPath cesty formálně

- ❑ XPath krok je formálně následující výraz

```
osa::test-uzlu predikát1 ... predikátN
```

- ❑ Osa, test uzlu a seznam predikáty
 - ❑ Osa a predikáty jsou nepovinné
 - ❑ Zatím jsme viděli pouze použití test-uzlu
 - Seznam predikátů byl prázdný
 - Osa byla defaultní
-

XPath osy

```
osa::test-uzlu predikát1 ... predikátN
```

- Osa specifikuje, v jaké relaci mají být uzly vybrané v tomto kroku vzhledem k uzlu **u** z kontextové množiny

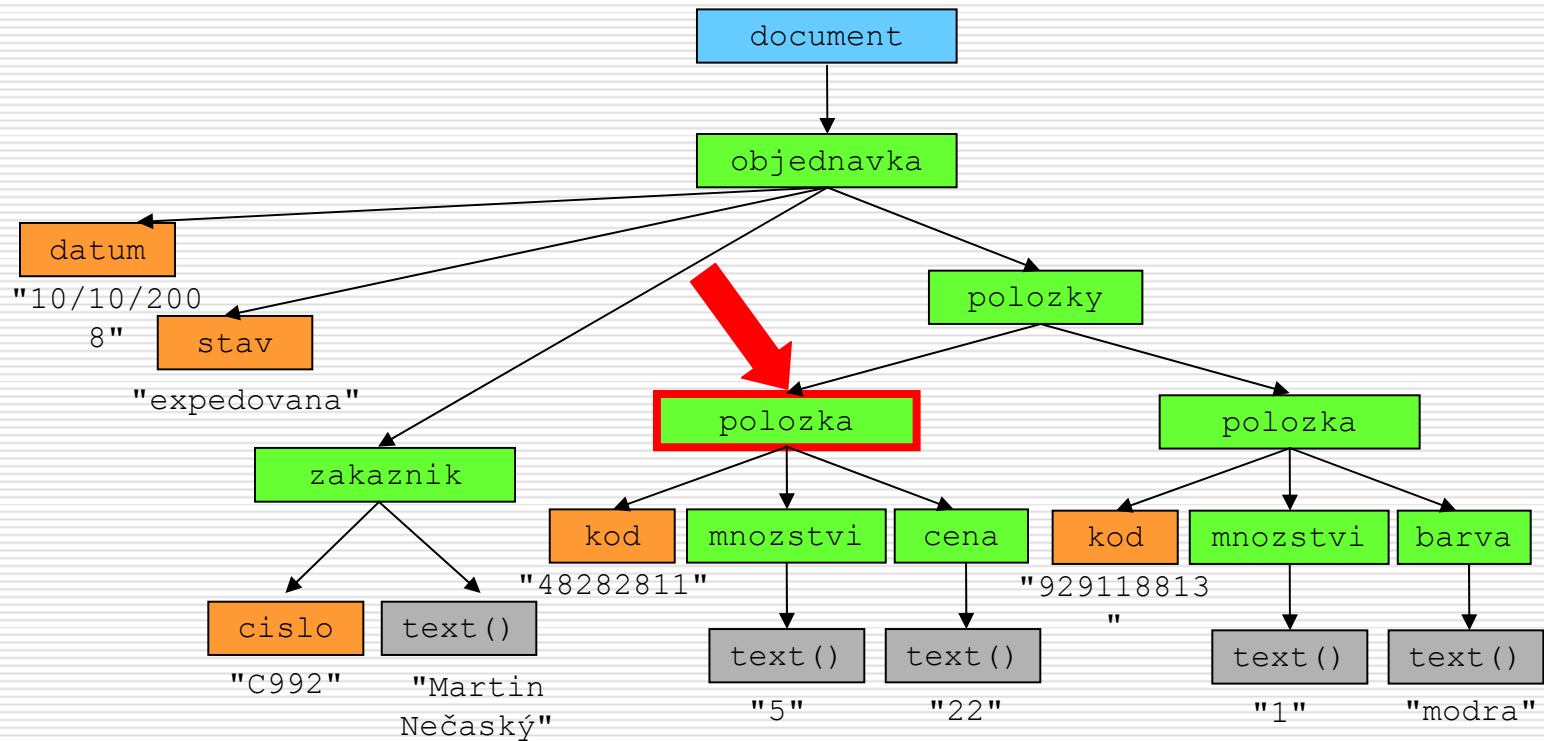
```
child
```

- Vybrané uzly jsou děti (přímí potomci) uzlu **u**
- Defaultní osa

```
/objednavka/zakaznik ↔  
  /child::objednavka/child::zakaznik
```

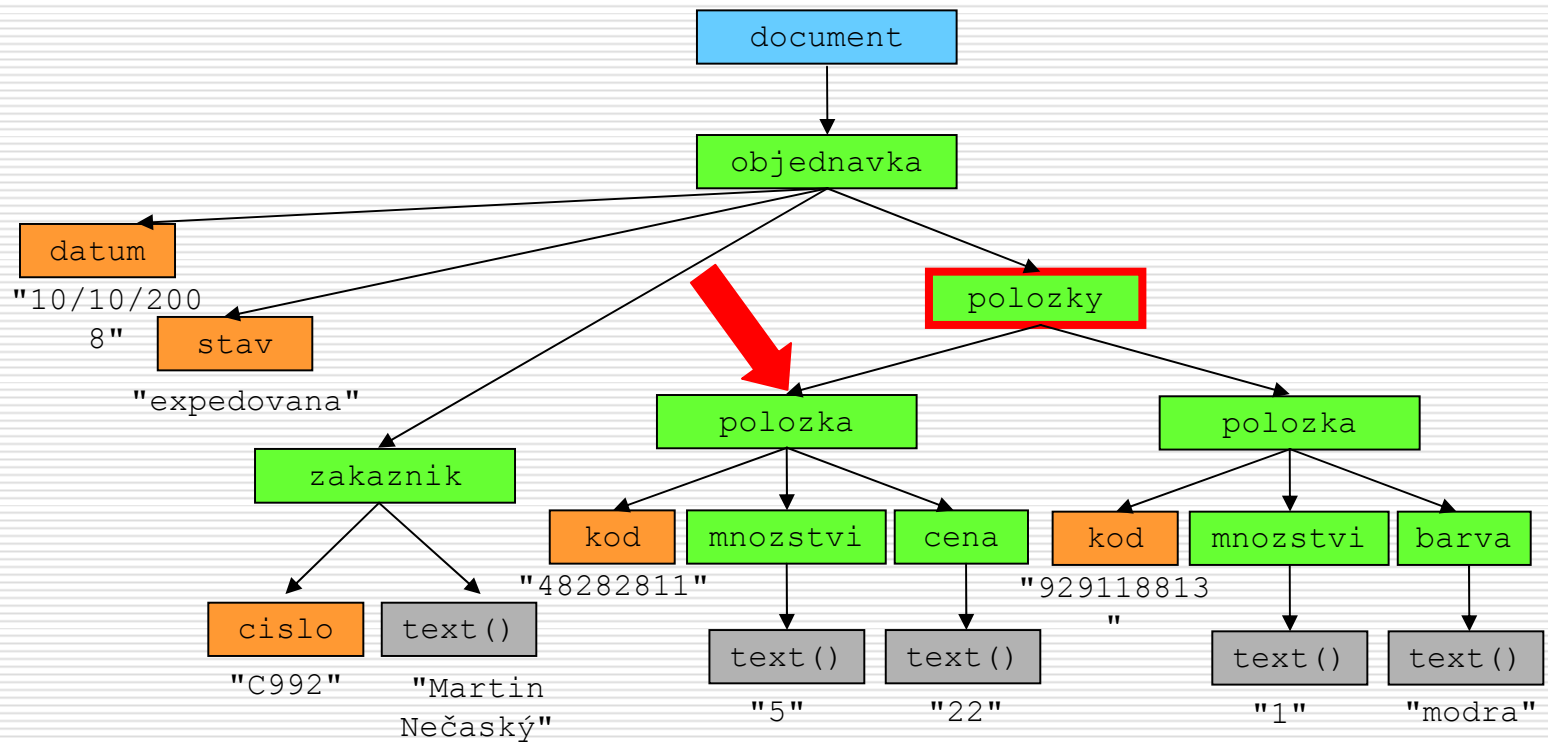
XPath osa **self**

- Vybraný uzel je samotný uzel **u**



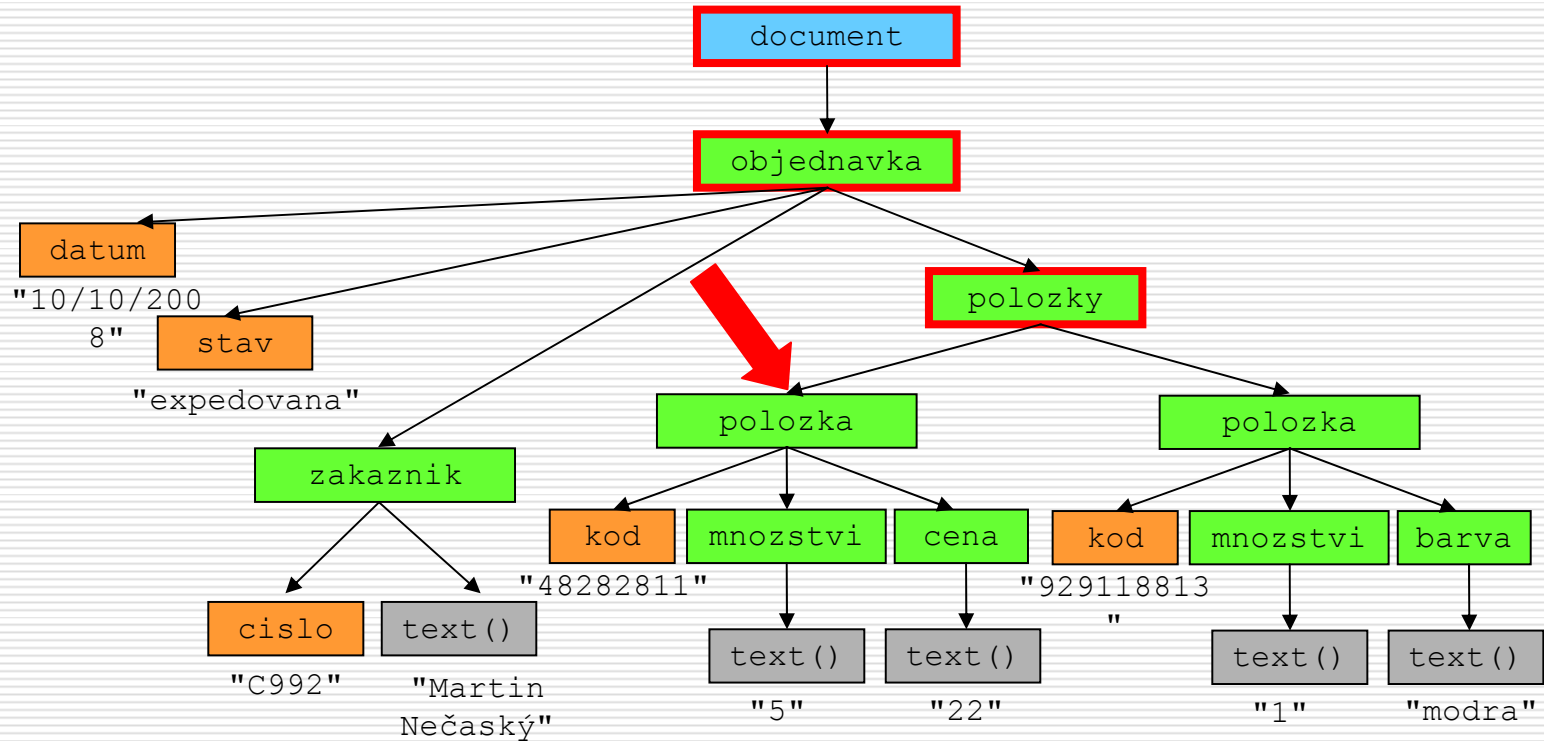
XPath osa **parent**

- Je vybrán rodič uzlu **u**



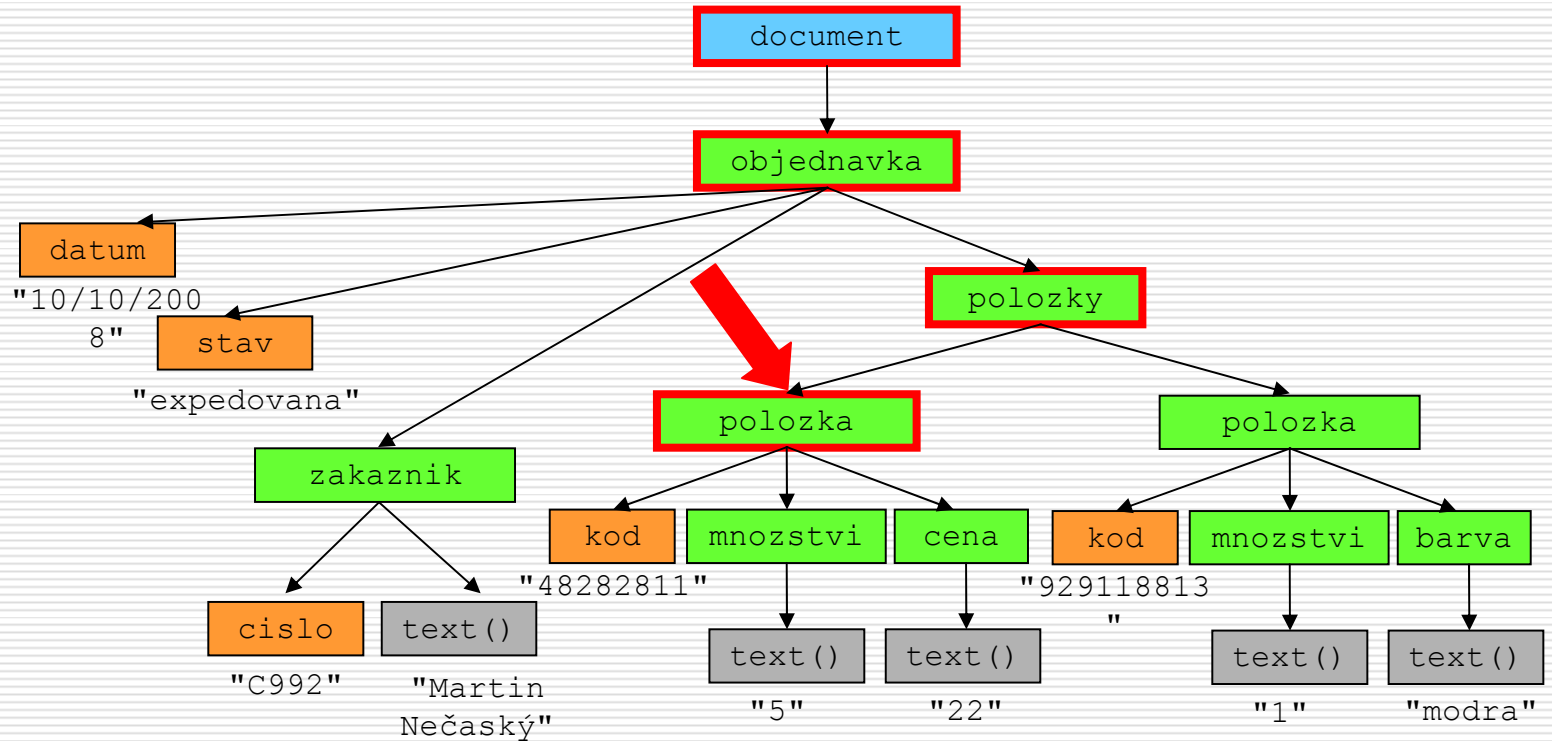
XPath osa **ancestor**

- Jsou vybrány všichni předci uzlu **u**
 - Tj. uzly ležící na cestě od **u** ke kořenovému uzlu



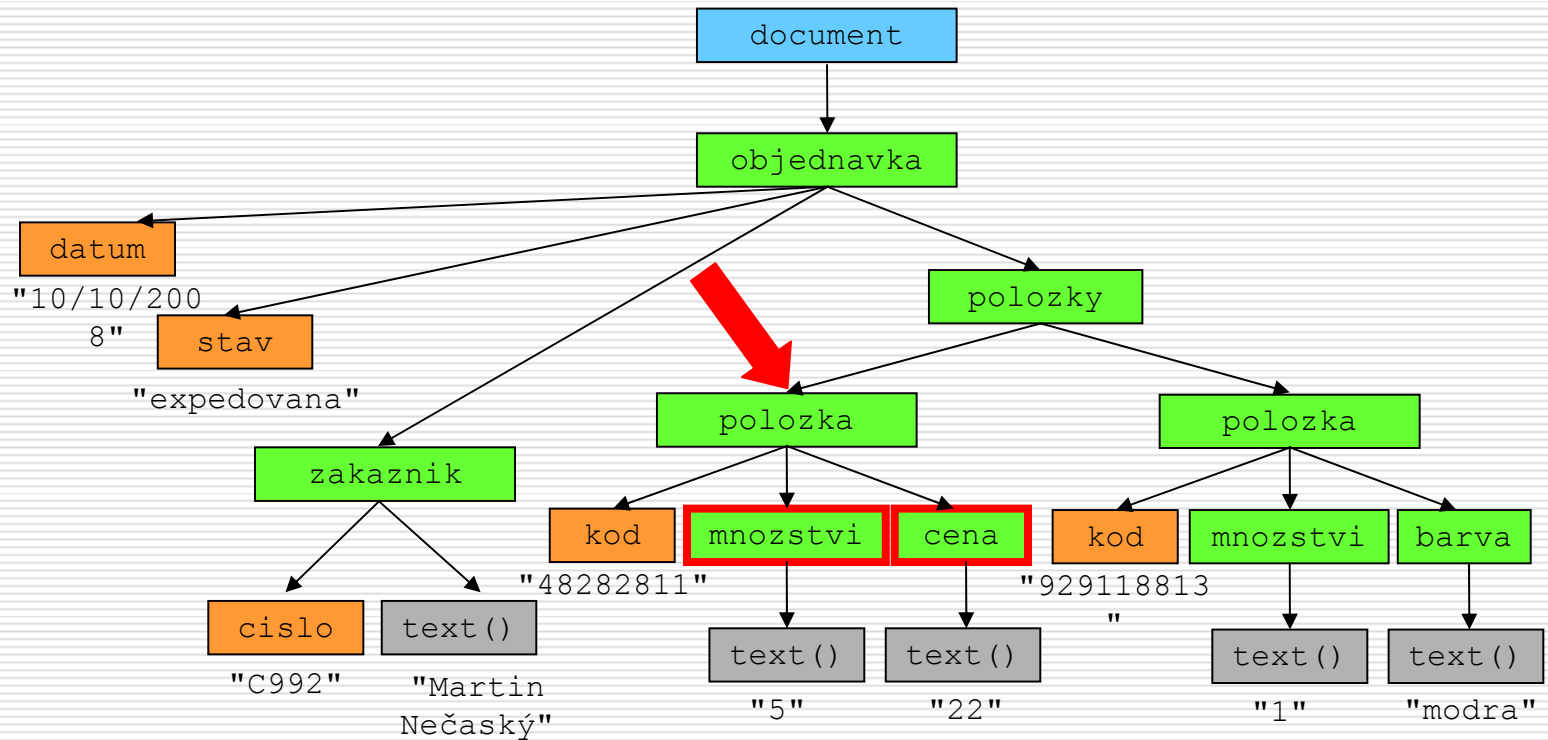
XPath osa **ancestor-or-self**

- Jsou vybrány všichni předci uzlu **u** včetně uzlu **u**



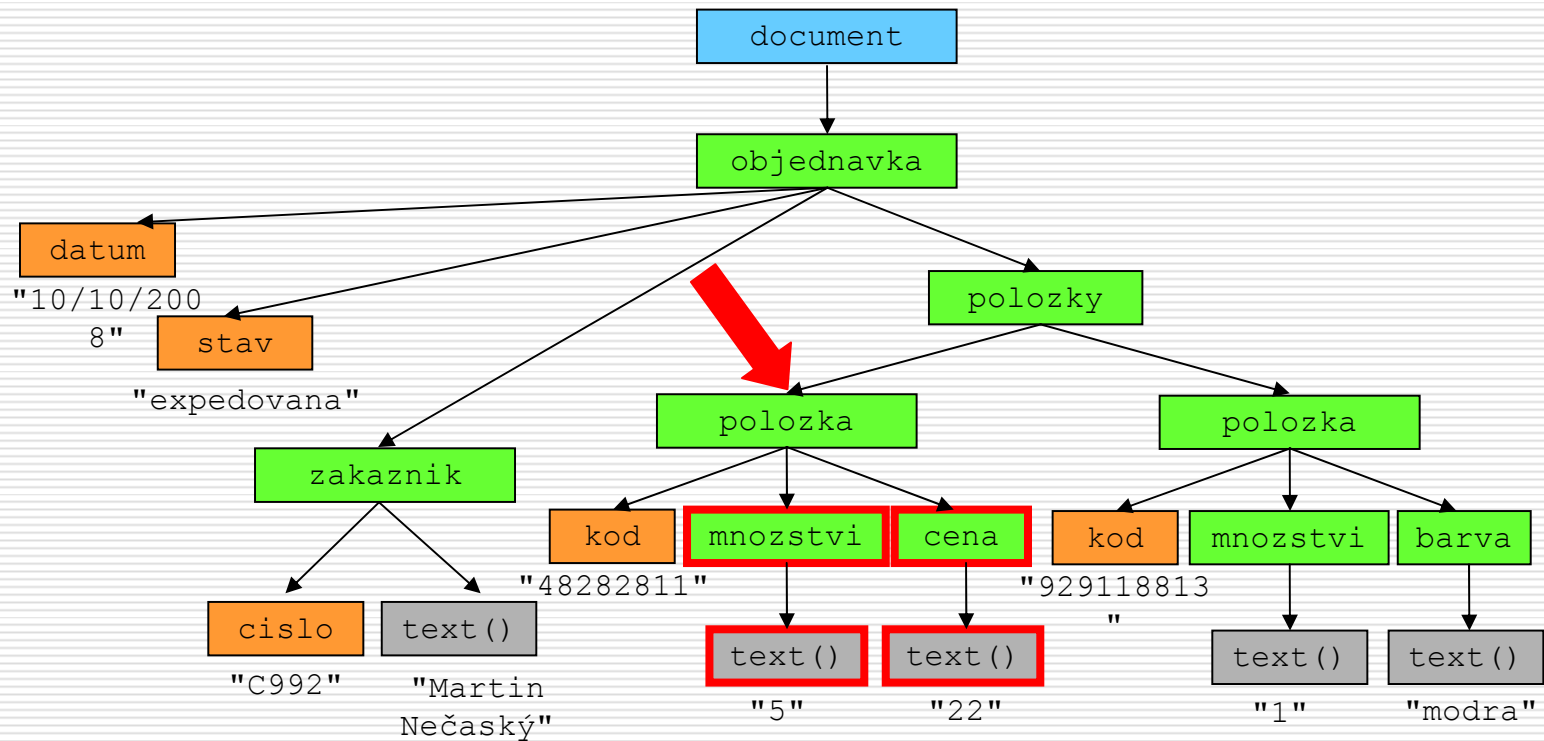
XPath osa **child**

- Jsou vybráni přímí potomci (děti) uzlu **u**



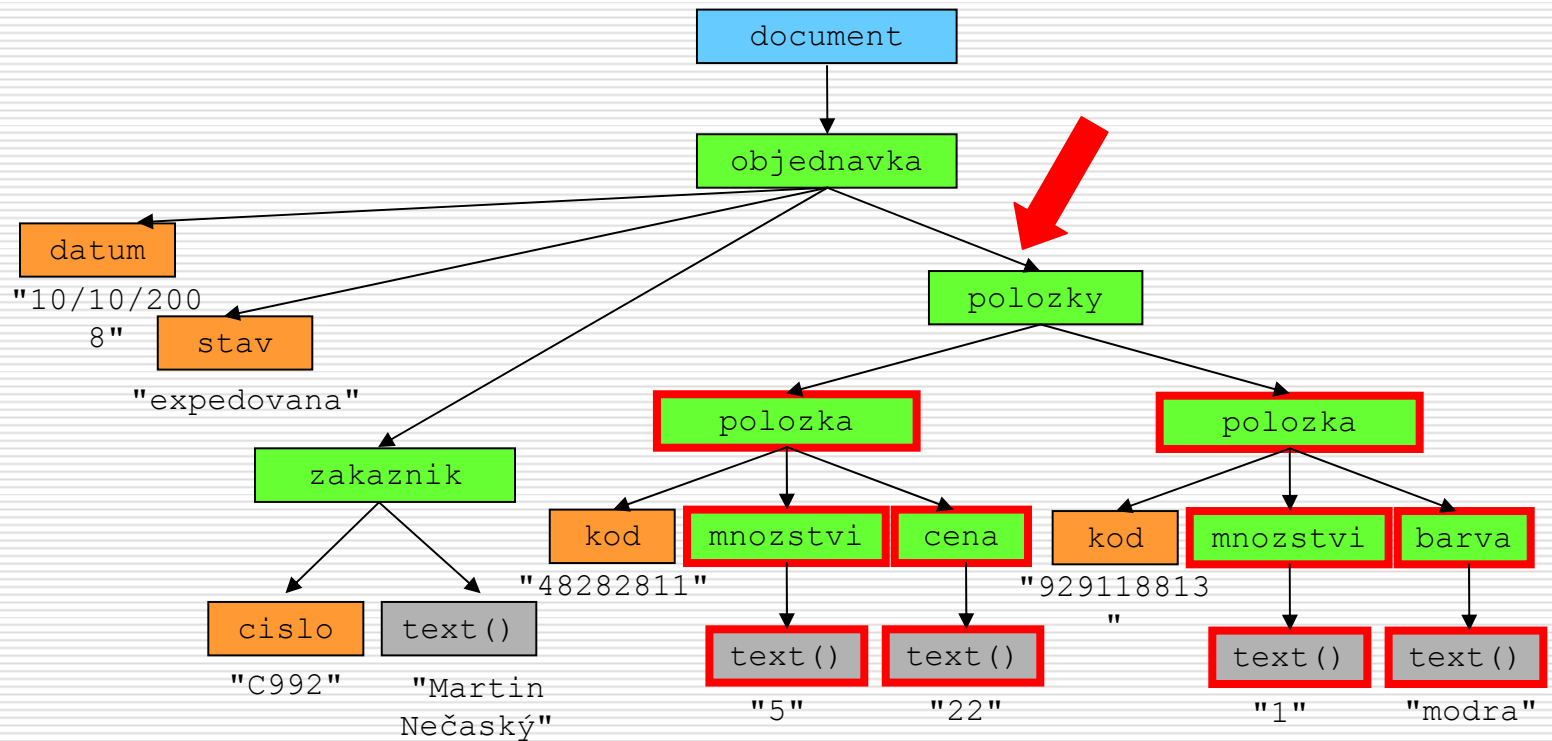
XPath osa descendant

- Jsou vybráni potomci uzlu **u**



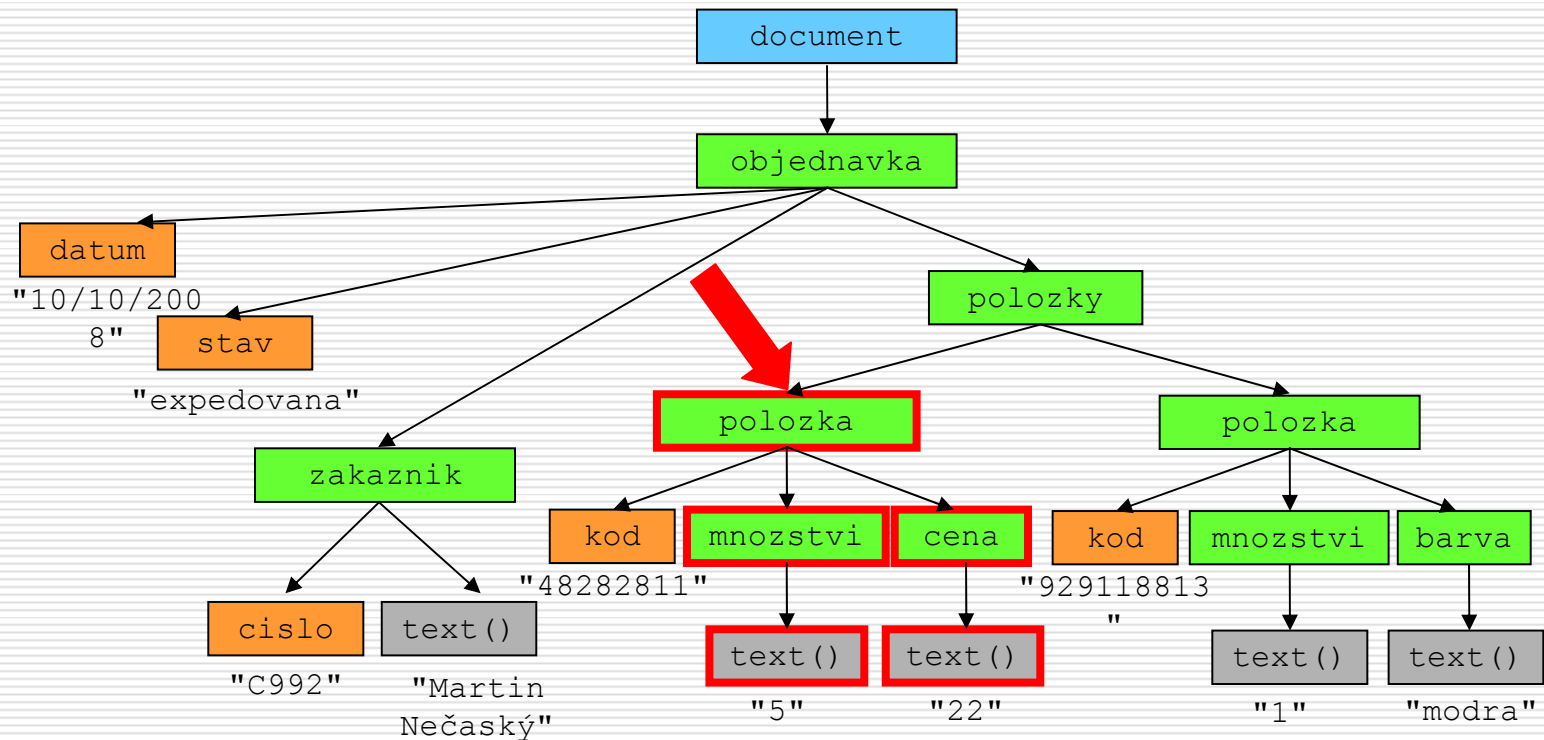
XPath osa descendant

- Jsou vybráni potomci uzlu **u**

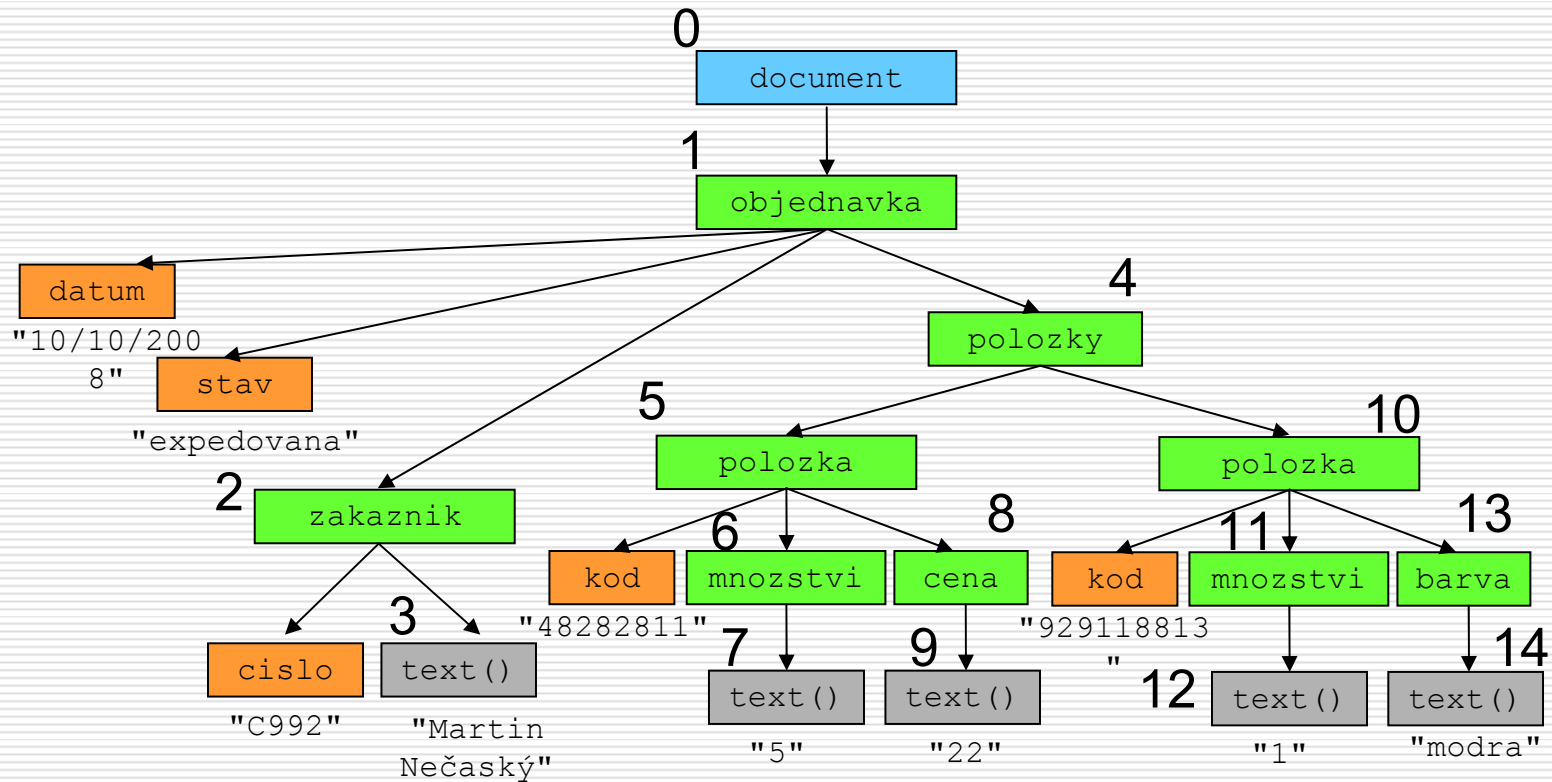


XPath osa descendant-or-self

- Jsou vybráni potomci a samotný uzel **u**

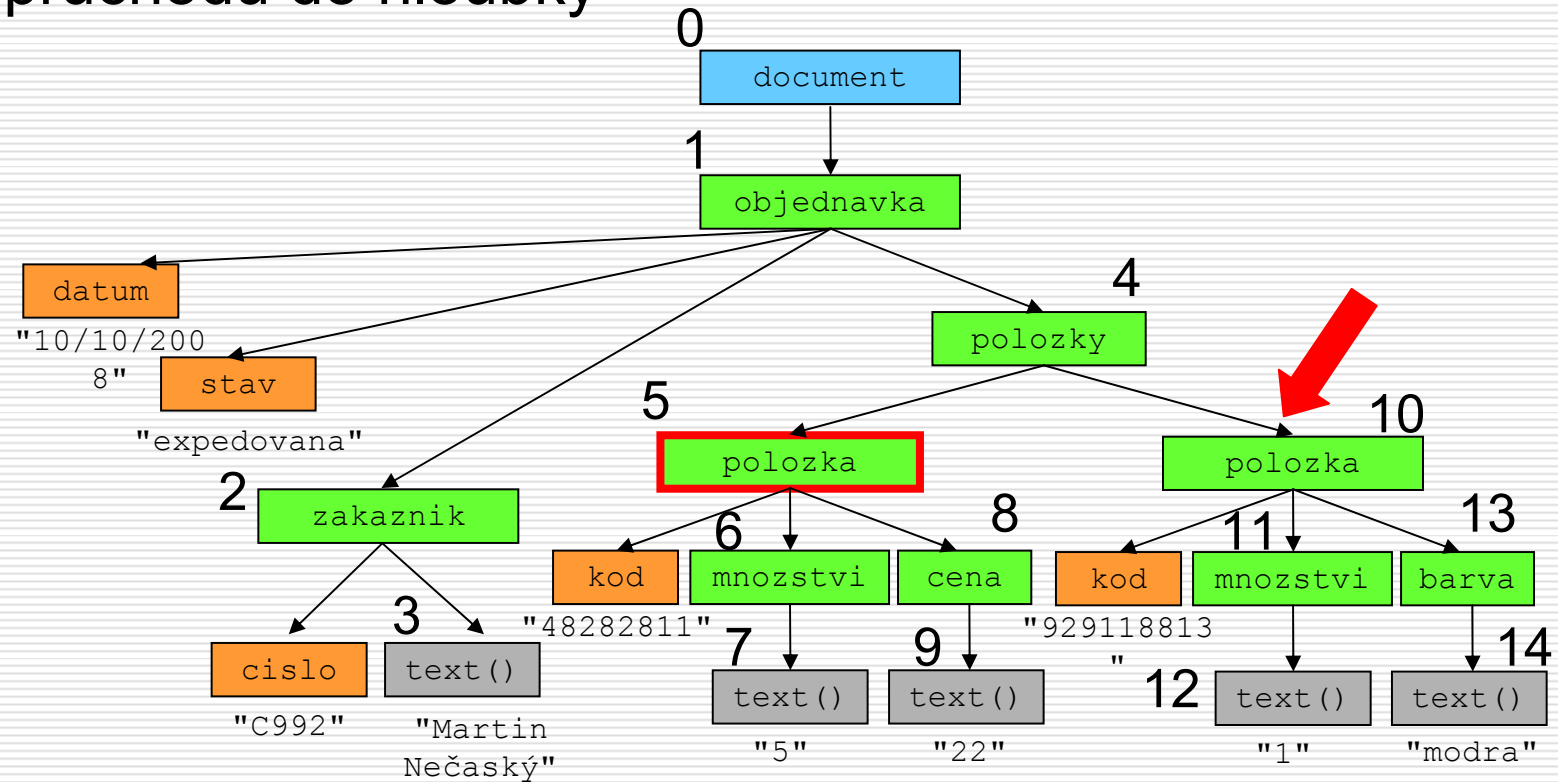


Průchod stromu do hloubky



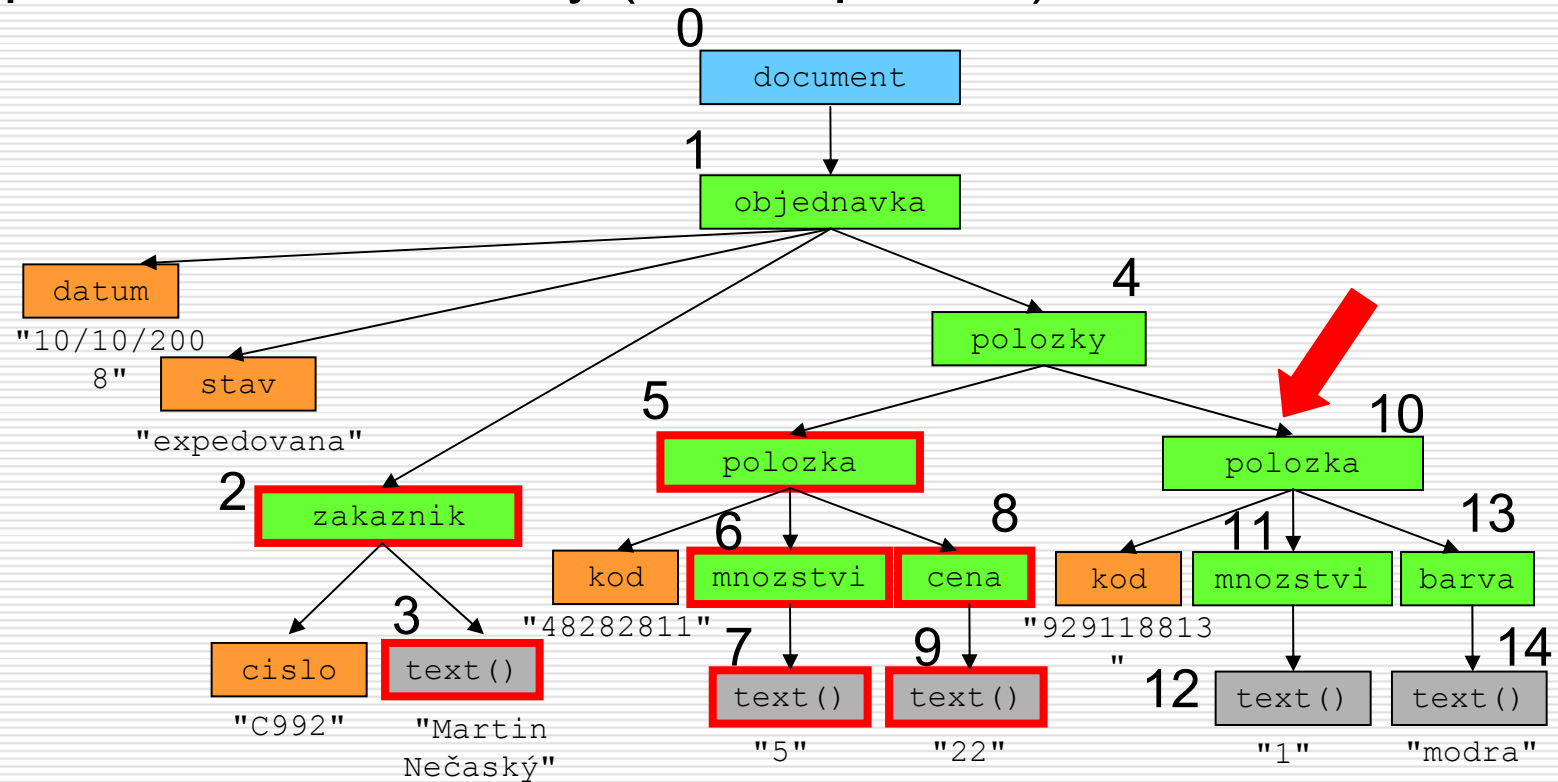
XPath osa preceding-sibling

- Jsou vybráni sourozenci uzlu **u**, které mu předcházejí v průchodu do hloubky



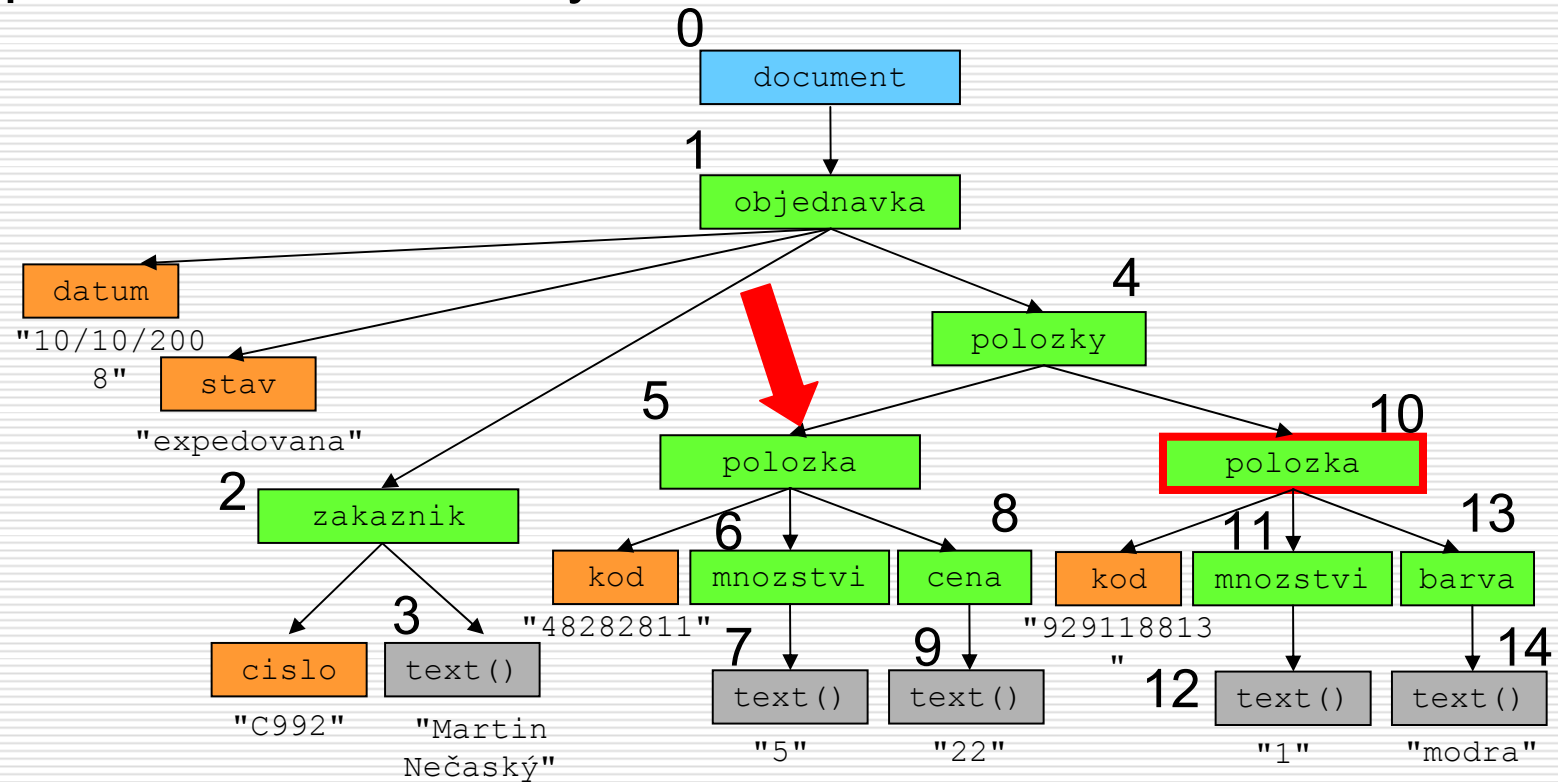
XPath osa **preceding**

- Jsou vybráni všechny uzly, které předcházejí uzlu **u** v průchodu do hloubky (kromě předků)



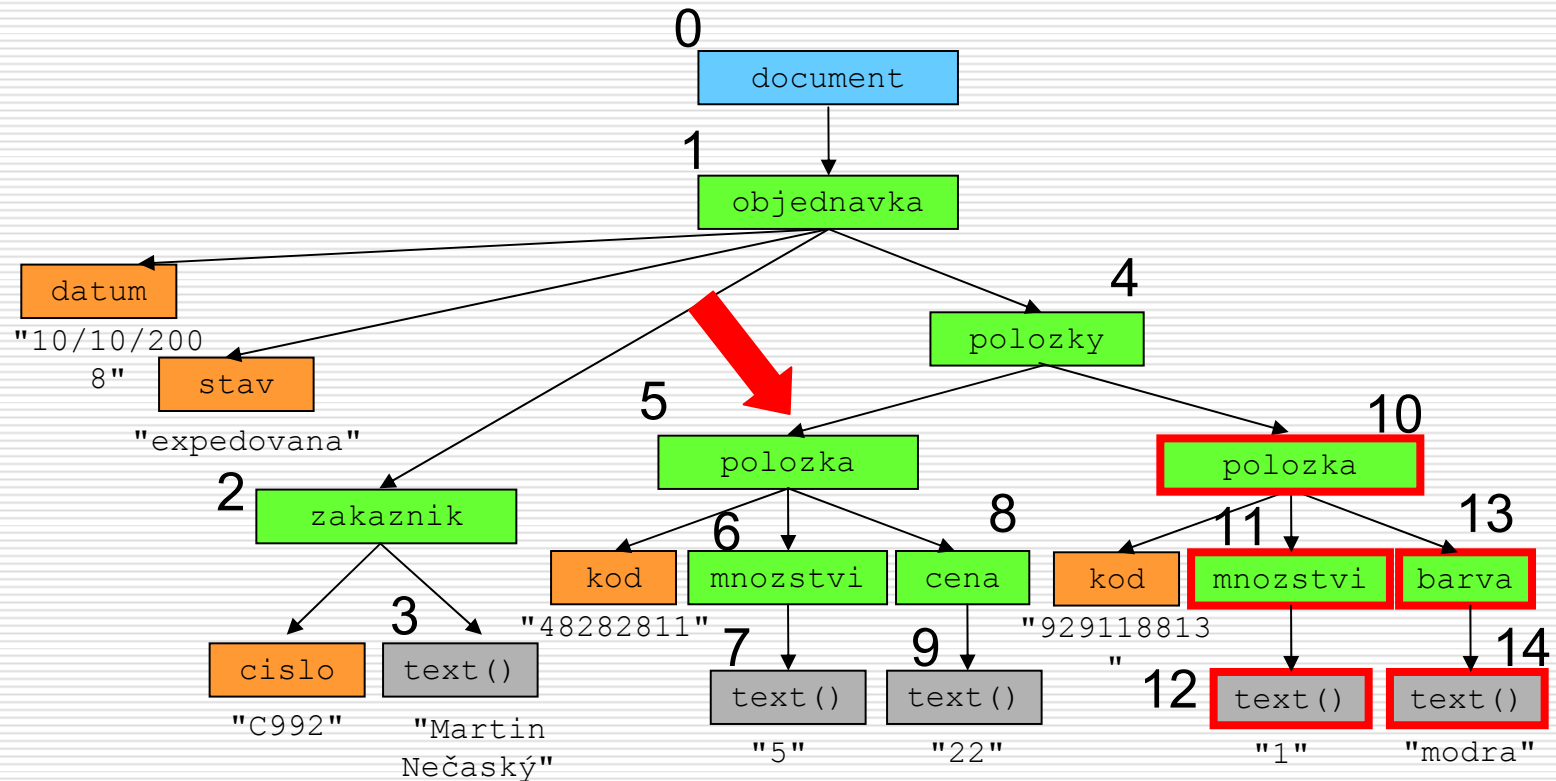
XPath osa following-sibling

- Jsou vybráni sourozenci uzlu **u**, které ho následují v průchodu do hloubky



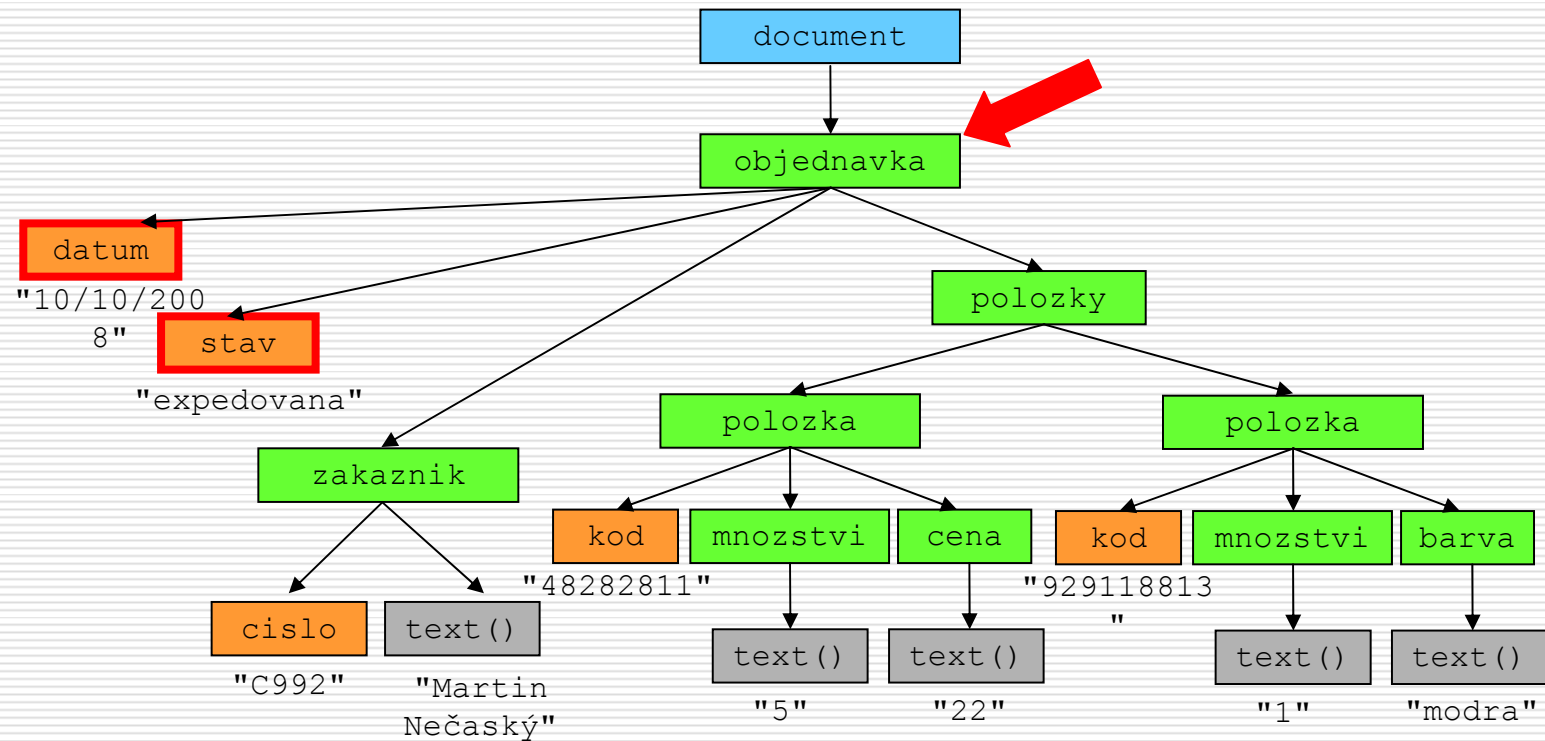
XPath osa following

- Jsou vybráni všechny uzly, které následují uzel **u** v průchodu do hloubky (kromě potomků)



XPath osa **attribute**

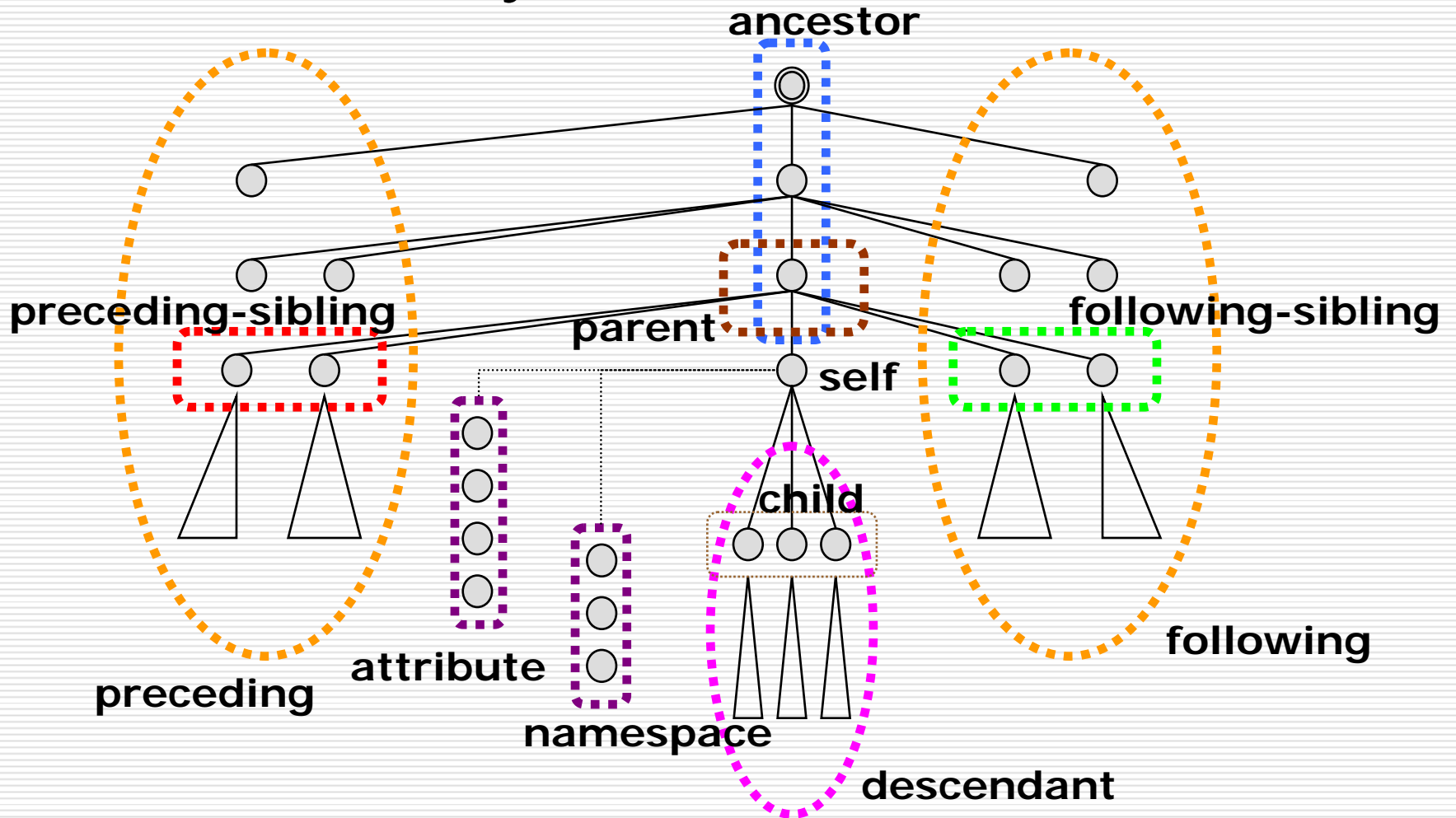
- Vybere atributy uzlu **u**



XPath osa namespace

- Vybere atributy ze jmenného prostoru **xmlns** uzlu **u**
-

XPath - osy



XPath test uzlu

```
osa::test-uzlu predikát1 ... predikátN
```

- Testuje uzly vybrané osou
 - Typ uzlu a případně název
-

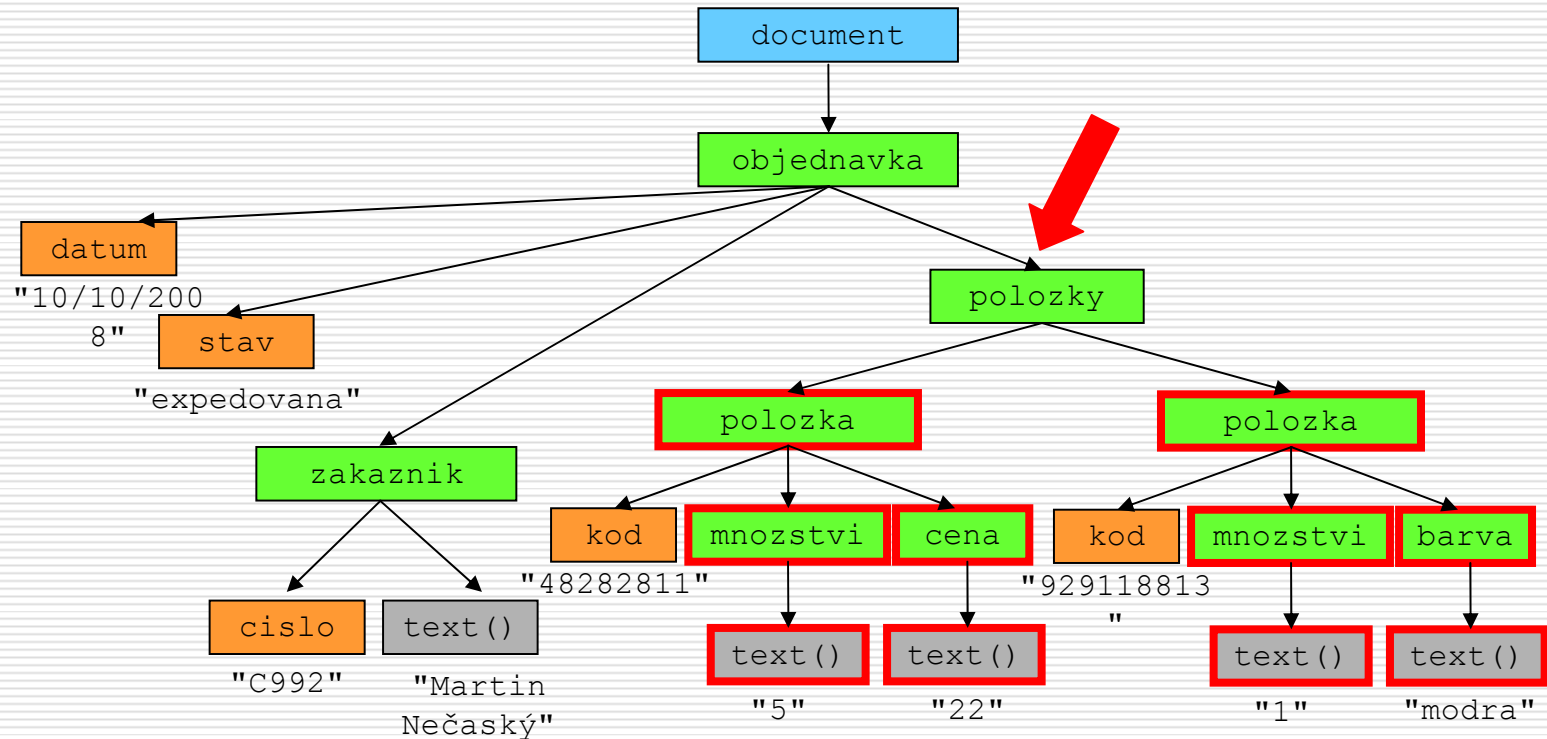
XPath test uzlu

```
osa::node() predikát1 ... predikátN
```

- Testem projde jakýkoliv uzel vybraný osou
-

XPath test uzlu

`descendant::node()`



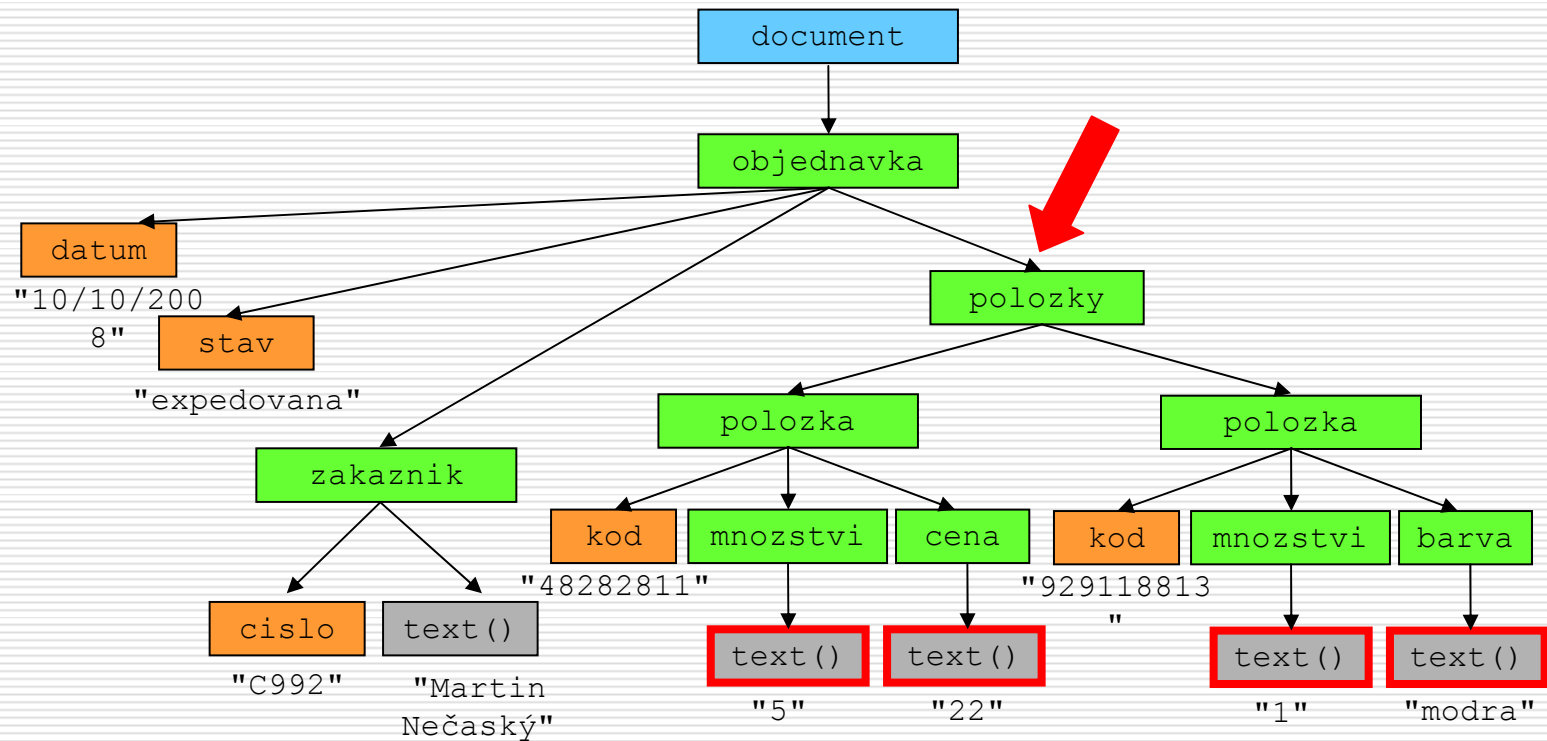
XPath test uzlu

```
osa::text() predikát1 ... predikátN
```

- Testem projde textový uzel vybraný osou
-

XPath test uzlu

`descendant::text()`



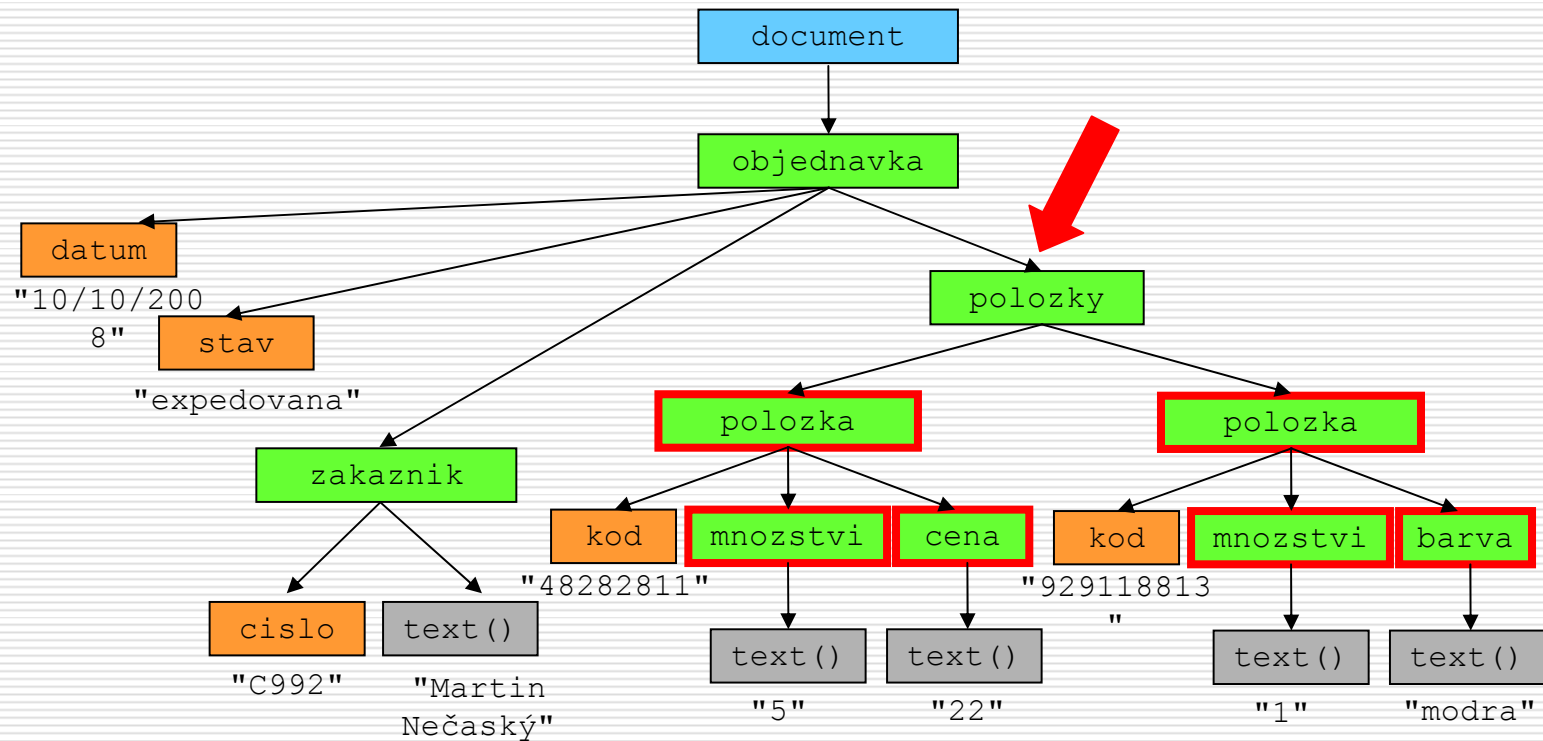
XPath test uzlu

```
osa::* predikát1 ... predikátN
```

- Testem projde jakýkoliv uzel vybraný osou, který má jméno
 - jméno má každý element a atribut
 - Ale pozor: neexistuje osa, která najednou vybere elementy i atributy
-

XPath test uzlu

descendant::*



XPath test uzlu

- ☐ `osa::comment()`
 - ☐ `osa::processing-instruction()`
 - ☐ `osa::processing-instruction("php")`
-

XPath osy a testy uzlů - zkratky

- Pro nejpoužívanější osy a testy uzlů jsou zavedeny zkratky

```
P/zakaznik ↔ P/child::zakaznik
```

```
P/@kod ↔ P/attribute::kod
```

```
P/../zakaznik ↔ P/parent::* /zakaznik
```

```
P//zakaznik ↔ P/descendant-or-self::node() /zakaznik
```

- `//zakaznik` vybere všechny elementy `zakaznik` v XML dokumentu
-

XPath predikáty

```
osa::test-uzlu predikát1 ... predikátN
```

- Predikát umožňuje specifikovat pokročilejší podmínky na uzly, které byly vybrány osou a prošly testem uzlu
- Pro kontextový uzel **u** jsou nalezeny všechny uzly ležící na dané ose z uzlu **u**, na výstup jdou ale pouze ty, které splňují test-uzlu a všechny predikáty

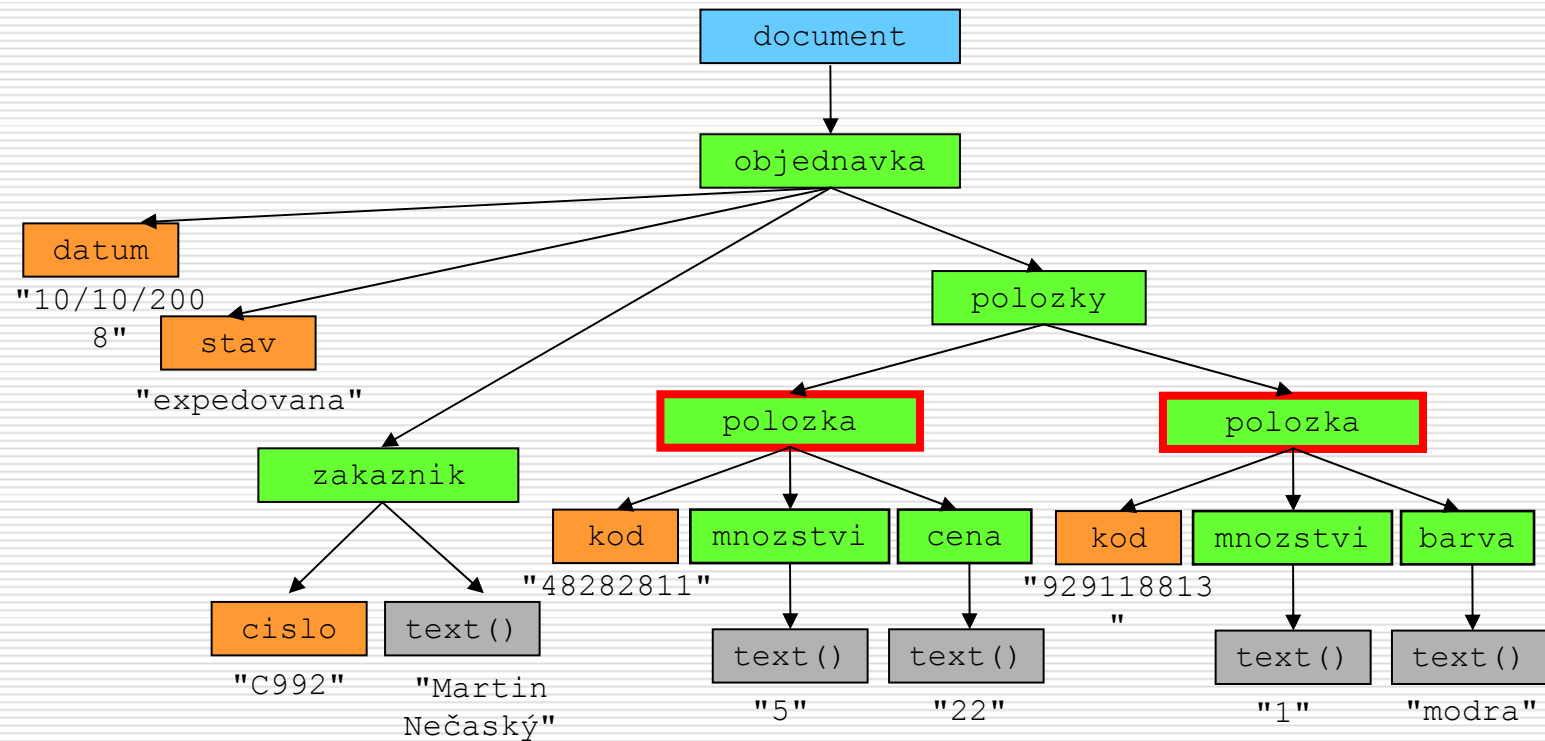
```
predikát ::= '[' podmínka ']'  
podmínka ::= 'not(' podmínka ')' |  
             podmínka 'and' podmínka |  
             podmínka 'or' podmínka
```

Testování existence uzlu

- Podmínkou může být relativní XPath cesta **P**
 - Pro daný uzel **u** se vyhodnotí **true**, pokud množina uzlů zacílená cestou **P** z **u** je neprázdná
-

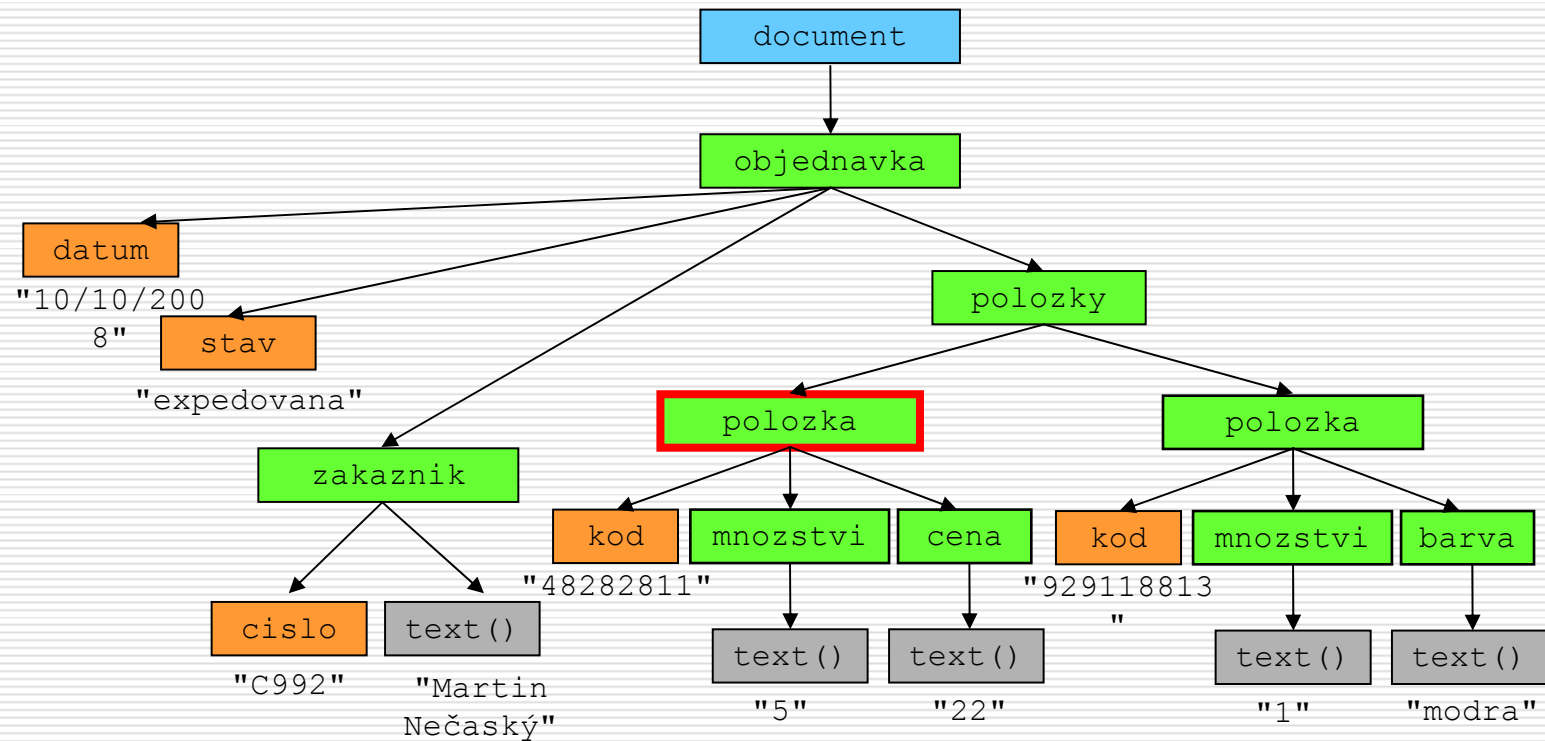
Testování existence uzlu

`//polozka[@kod]`



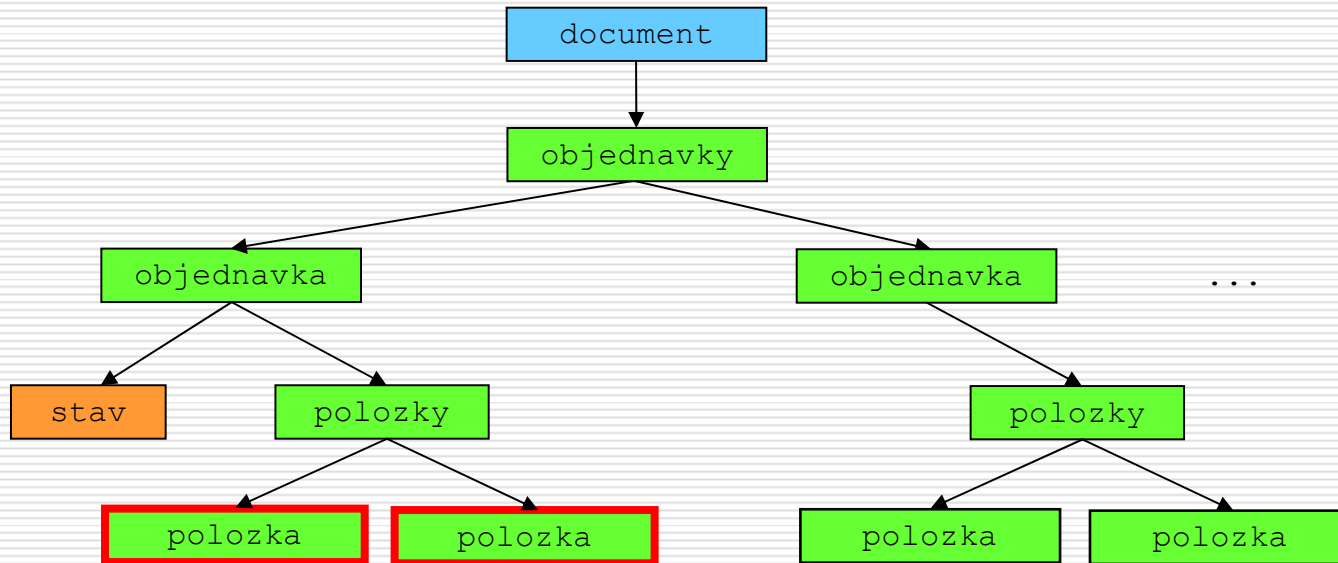
Testování existence uzlu

//polozka[cena]



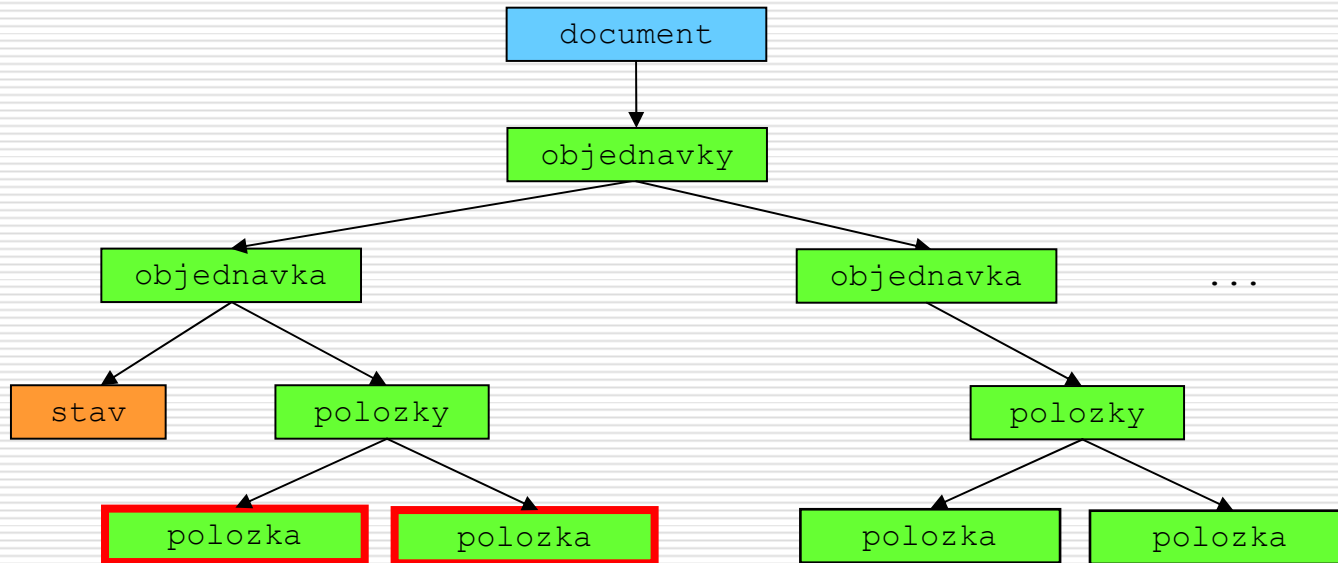
Testování existence uzlu

`//polozka[.../.../@stav]`



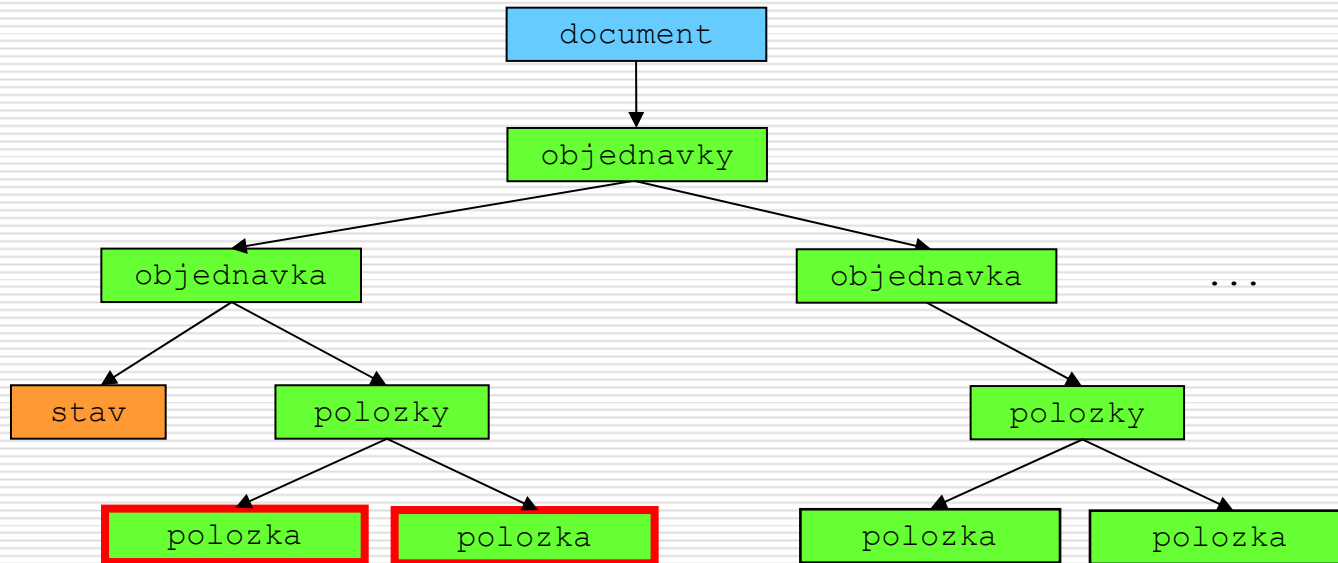
Testování existence uzlu

`//polozka[ancestor::objednavka/@stav]`



Testování existence uzlu

`//objednavka[@stav]//polozka`

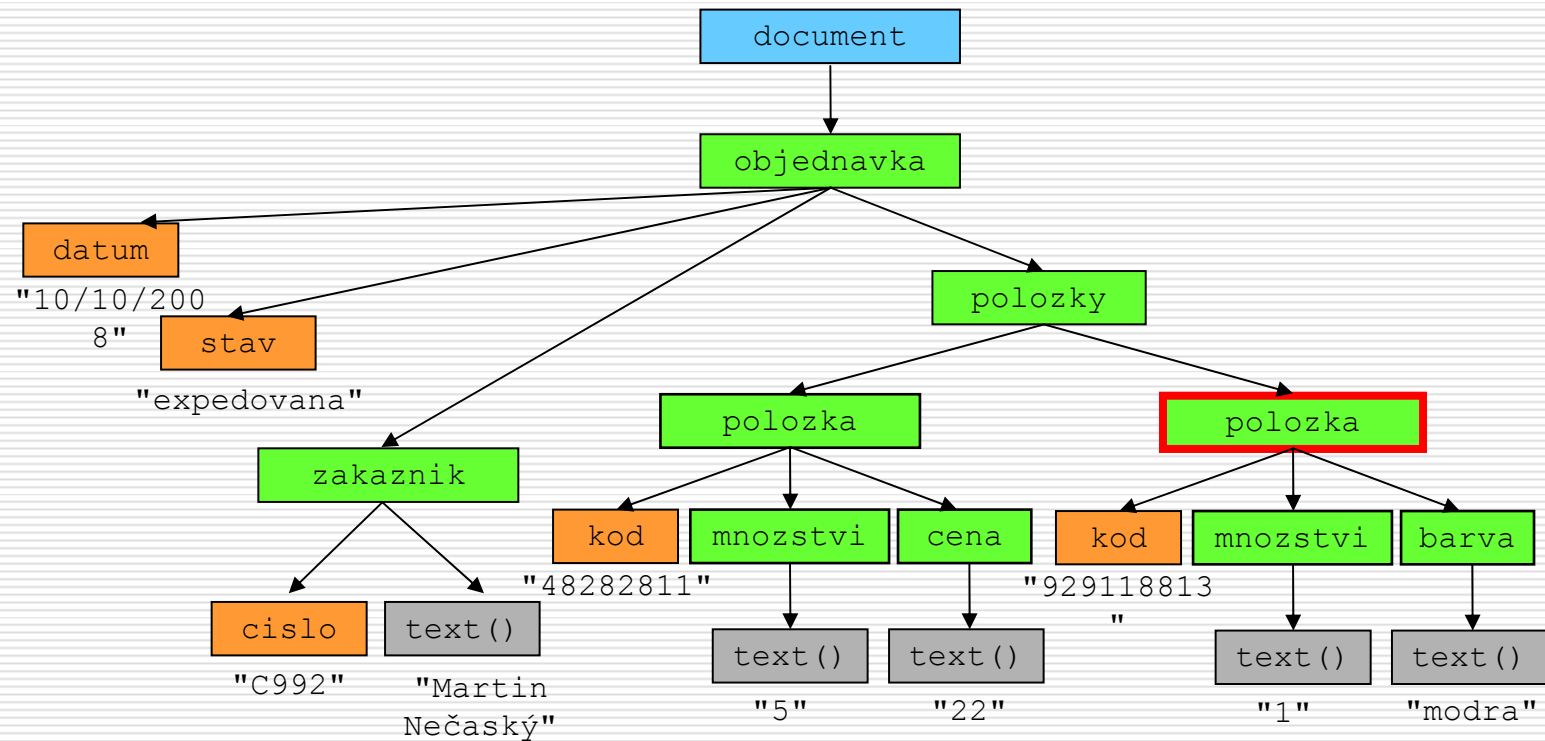


Testování hodnoty

- Podmínkou může být porovnání dvou operandů
 - Operandy jsou XPath výrazy
 - Cesta, hodnota ...
 - Operátor je
 - = != < > <= >=
 - &eq; ≠ < > ≤ ≥
-

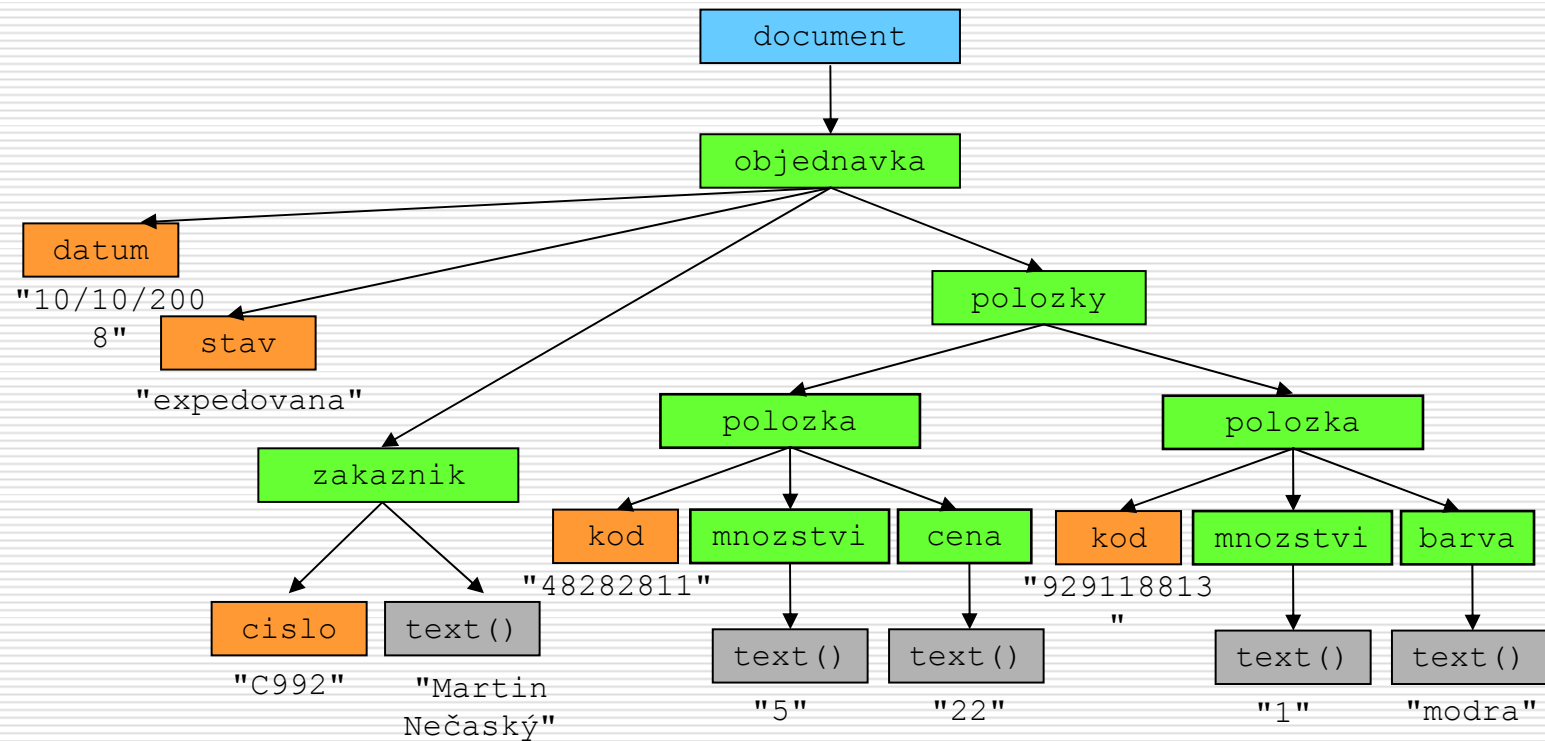
Testování hodnoty

```
//polozka[barva = "modra"]
```



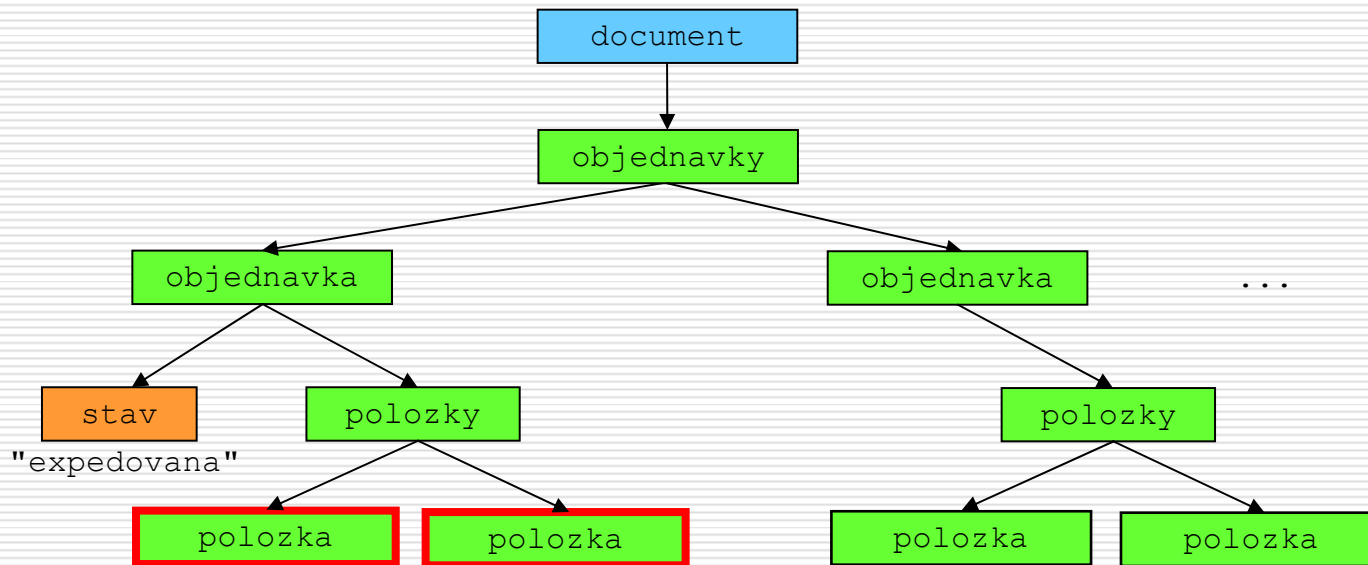
Testování hodnoty

```
//polozka[cena > 30]
```



Testování hodnoty

```
//objednavka[@stav = "expedovana"]//polozka
```

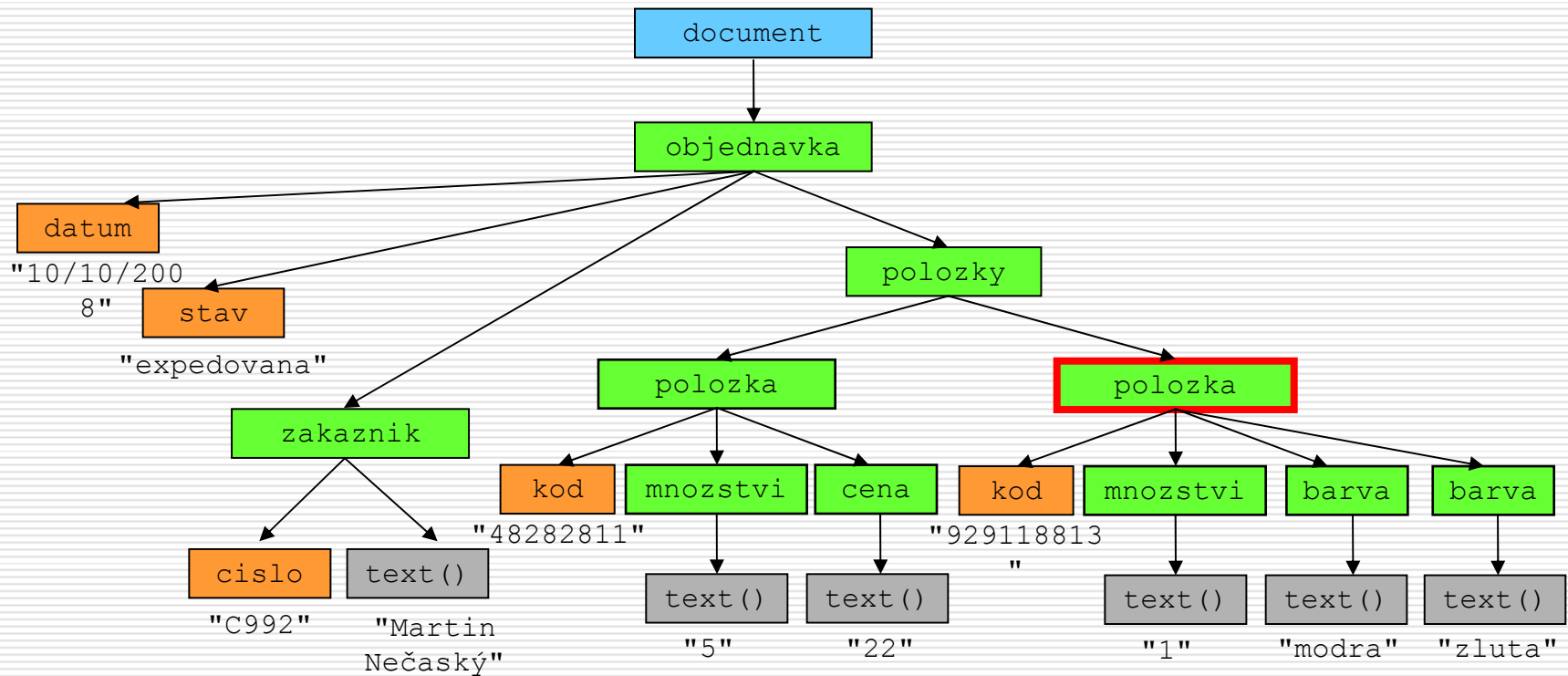


Testování hodnoty

- Operátory = != ...
 - Operandy jsou množiny hodnot/uzlů
 - Vyhodnotí se jako true pokud existuje hodnota/uzel v levém a hodnota/uzel v pravém operandu, pro něž je operátor true
 - Operátory &eq; &neq; ...
 - Operandy nesmějí být množiny
 - Pokud je operandem uzel/množina uzlů, je/jsou převeden(y) na textovou hodnotu
 - atomizace
-

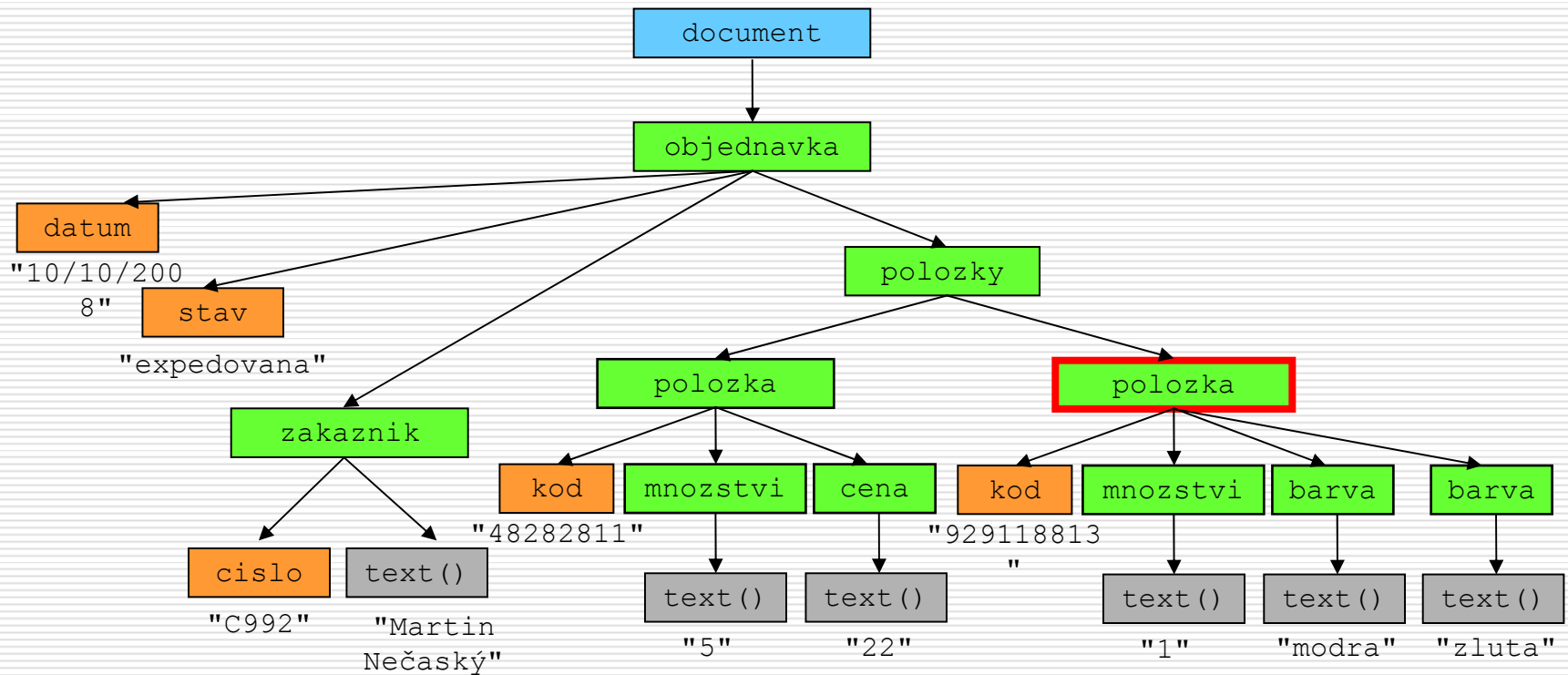
Testování hodnoty

```
//polozka[barva = "modra"]
```



Testování hodnoty

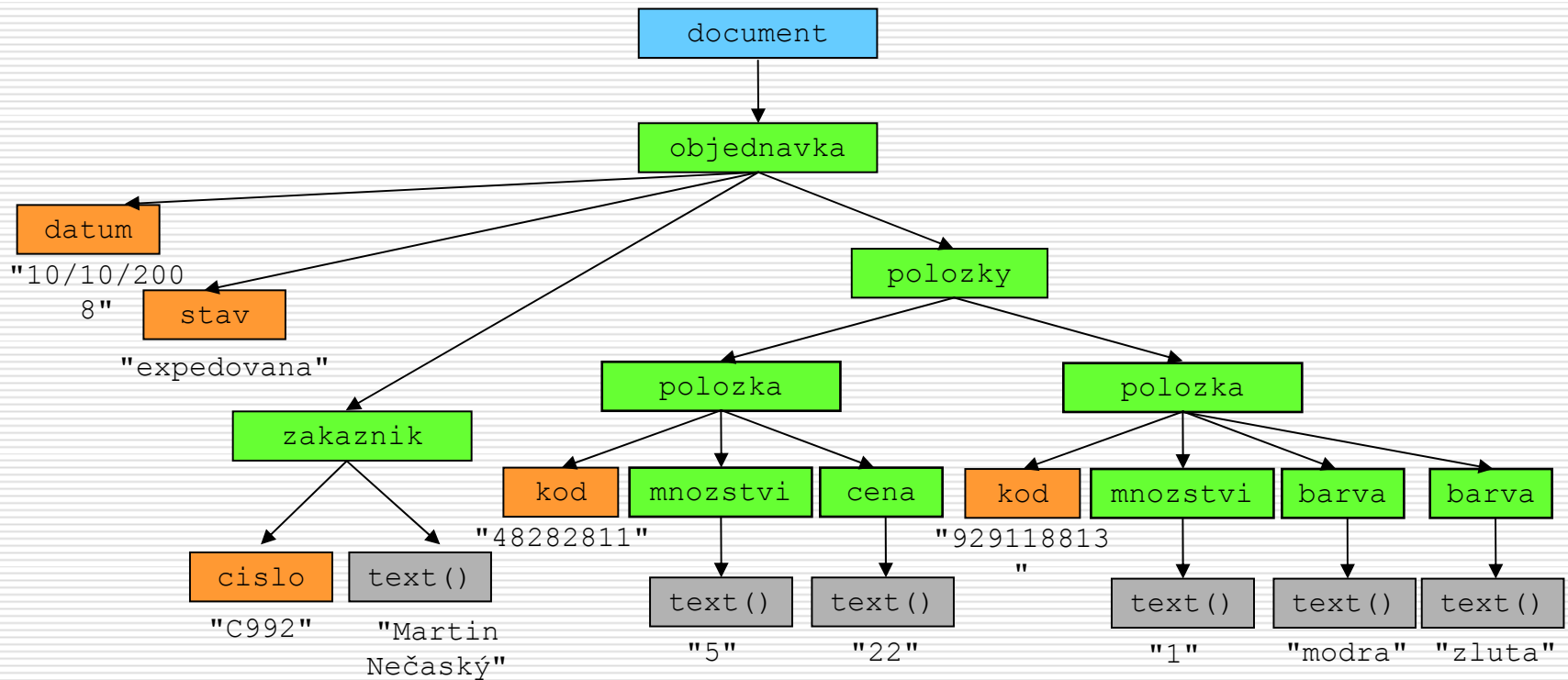
```
//polozka[barva != "modra"]
```



Testování hodnoty

```
//polozka[barva &eq; "modra"]
```

CHYBA!



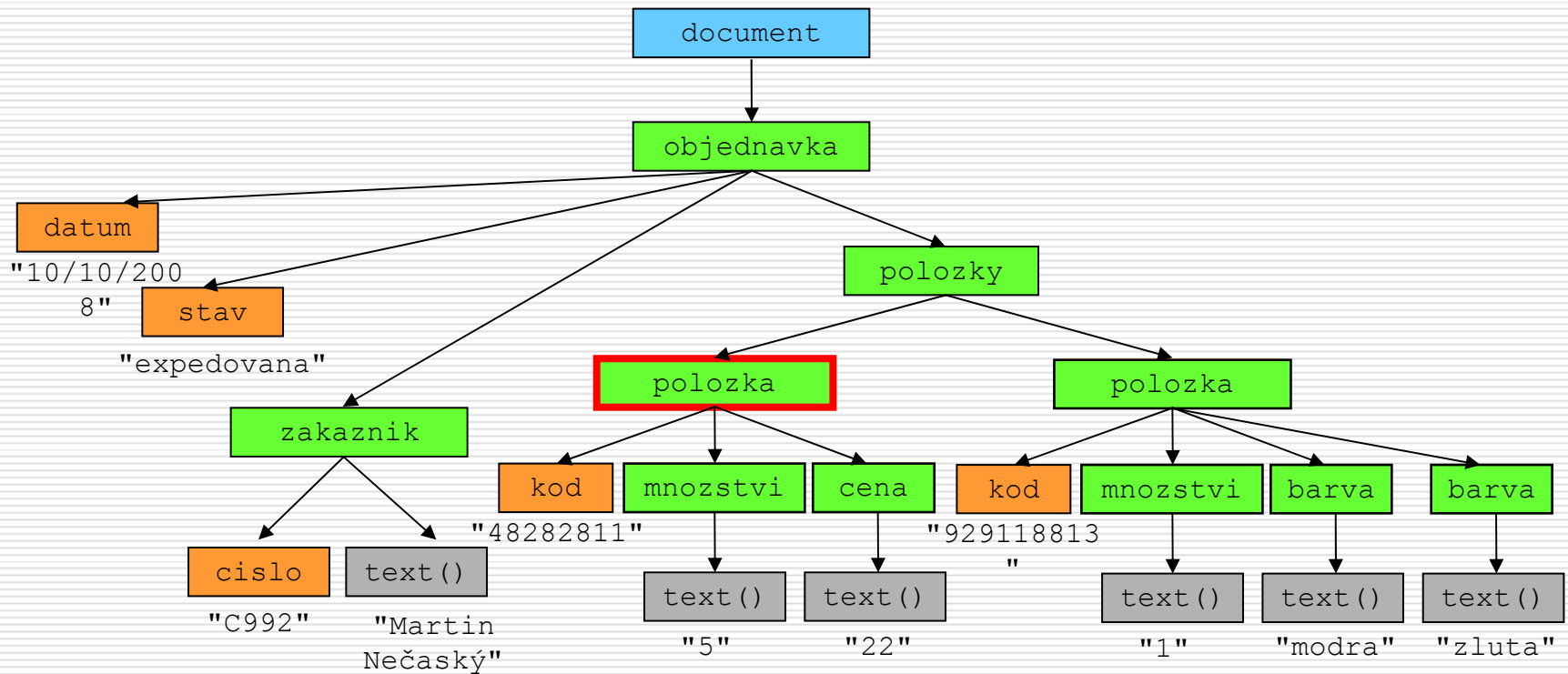
Vestavěné funkce

■ Testování pozice

- každý uzel v kontextové množině má přiřazenu pozici (určena pozicí v dokumentu a směrem osy)
- `position()`
 - Vrací pozici uzlu v kontextové množině
- `last()`
 - Vrací počet uzlů v kontextové množině

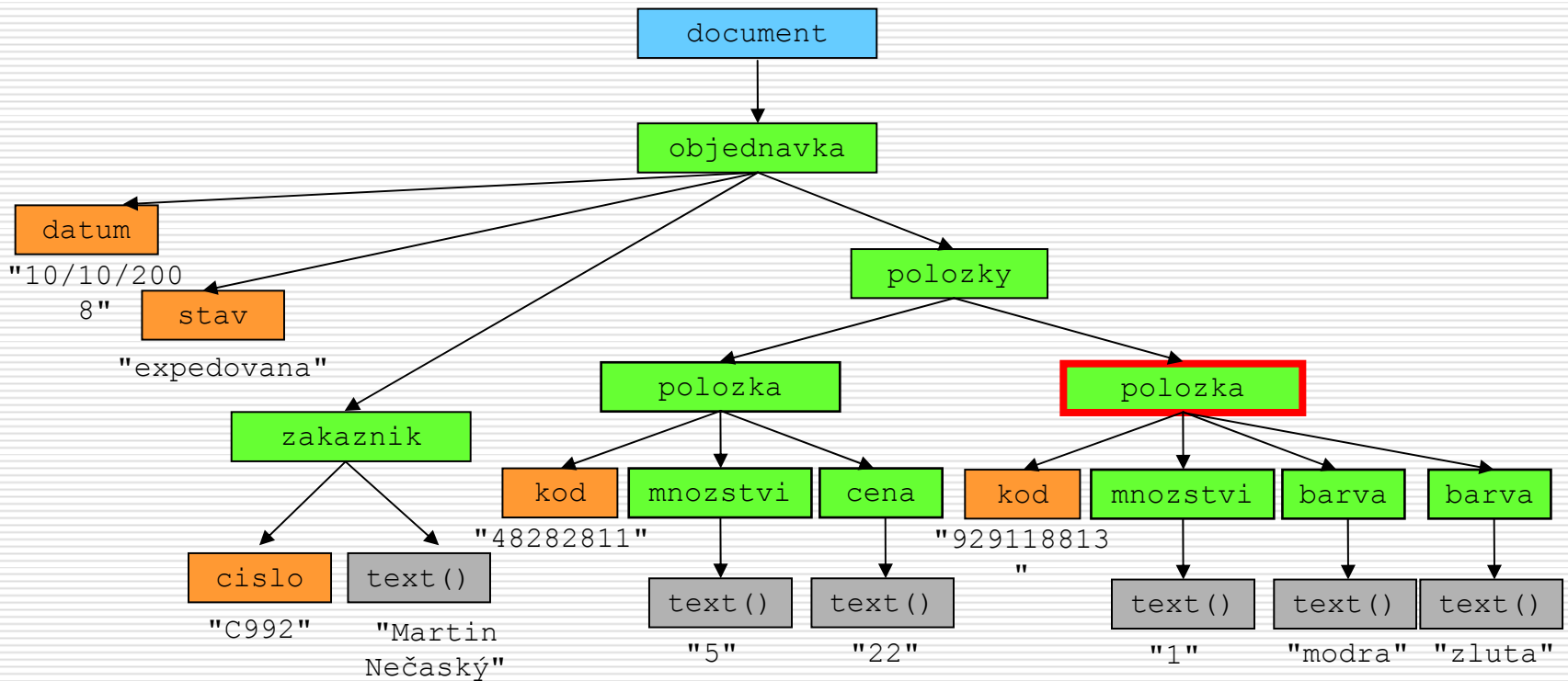
Vestavěné funkce

```
//polozky/polozka[position() = 1]
```



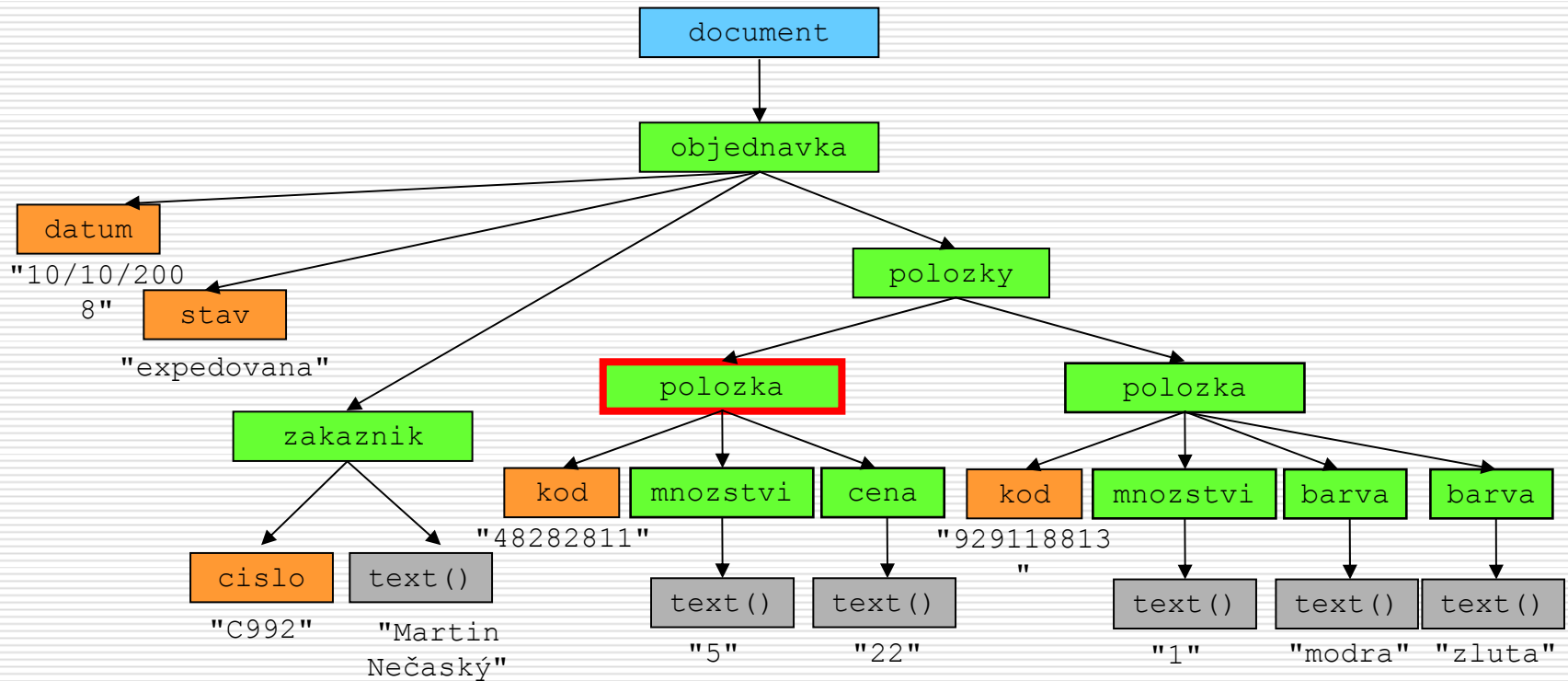
Vestavěné funkce

```
//polozky/polozka[position() = last()]
```



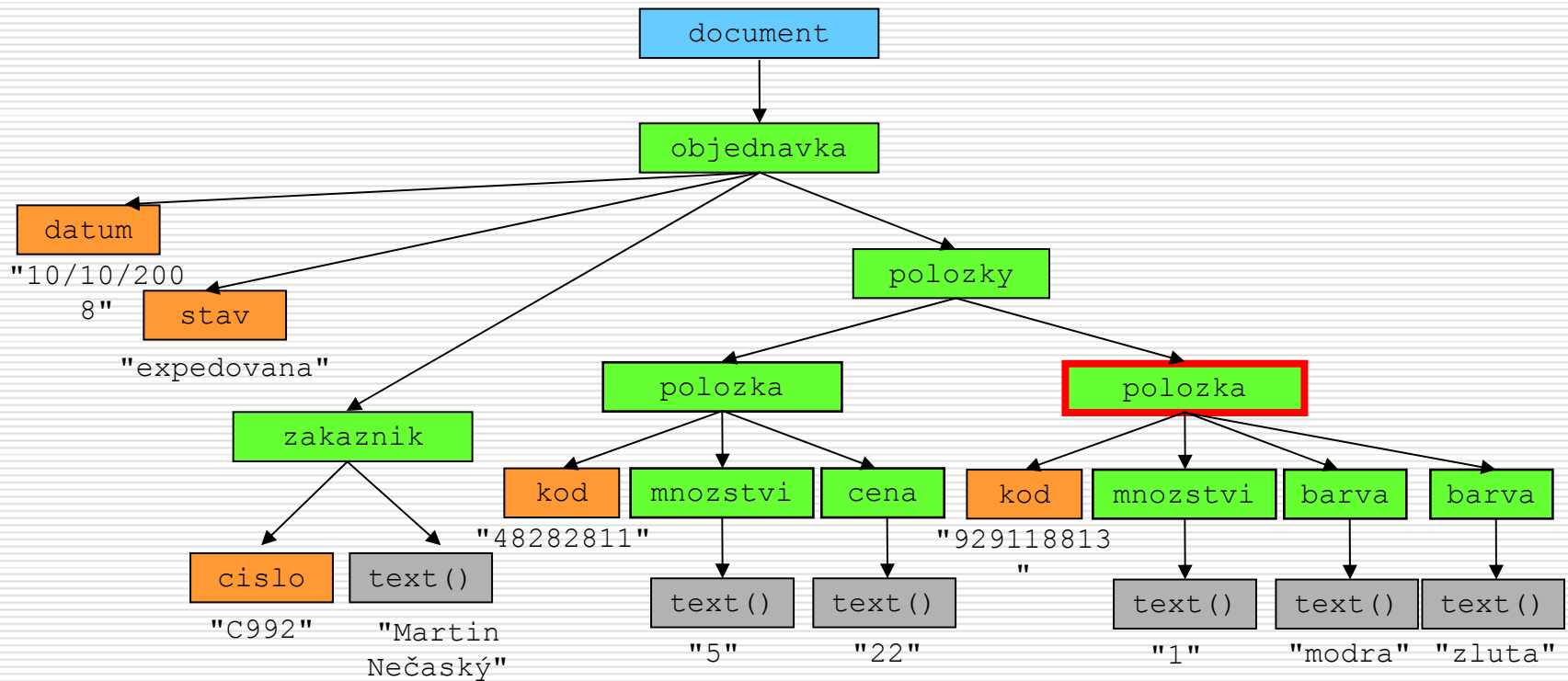
Vestavěné funkce

```
//polozky/polozka[1]
```



Vestavěné funkce

```
//polozky/polozka[last()]
```



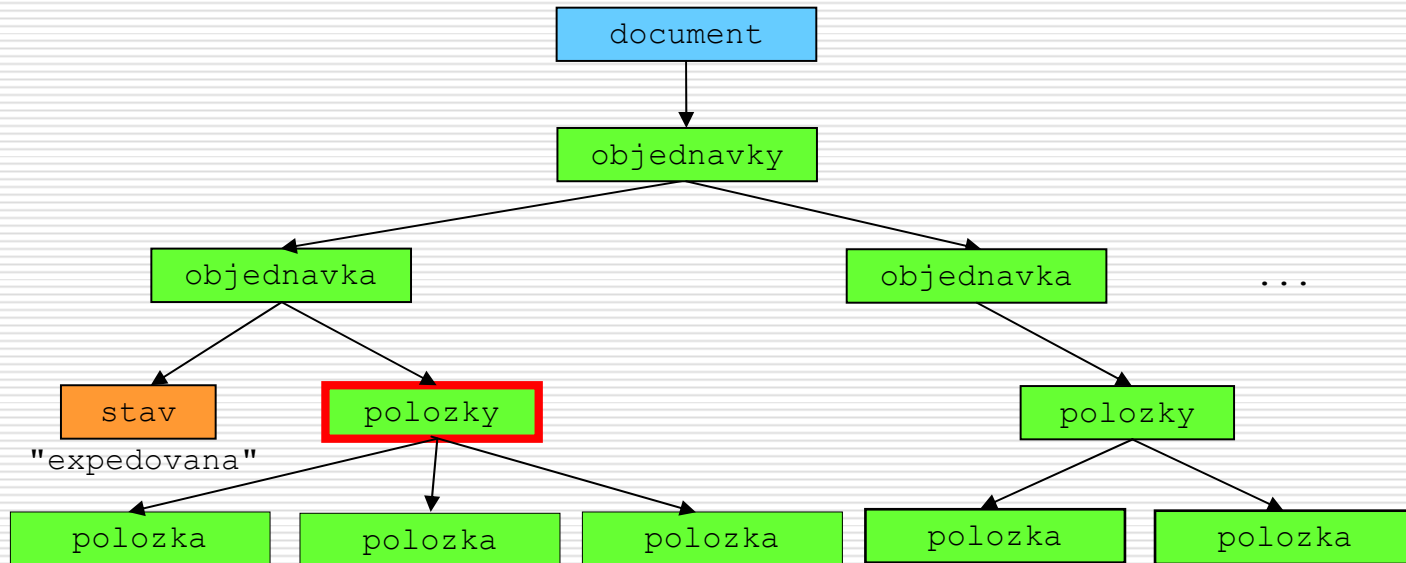
Vestavěné funkce

`count (expression)`

- Vrací počet uzlů vrácených XPath výrazem *expression*

Vestavěné funkce

```
//polozky[count(polozka) > 2]
```



Vestavěné funkce

- `id(value)`
 - Vrátí element s *value* jako id
 - `concat, starts-with, contains, substring-after, substring-before, substring(value), ...`
 - `sum, floor, ceiling, ...`
-



Konec

