

KP  
CV.

ASN1 = abstract syntax notation one ← ITU X 680

typy - jednoduchý (např. Integer, Boolean,  
invičt plom.) - štrukturovaný { }

- odvozený  
- iné typy (choice a any...)

hodnota je 1 prvok a 1 možnosť  
identifikátor = identifikátor premenných „a“ (malé písmená)

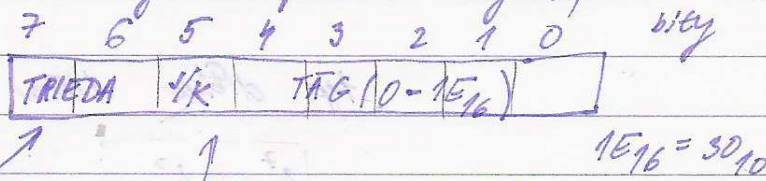
napis :  
a INTEGER

BER Basic Encoding Rules - kódovanie

DER Distinguished Encoding Rules kódovanie (ušetrenie počtu bytov)

### POLE TYPU DAT PRE TAGY MENŠIE AKO 31

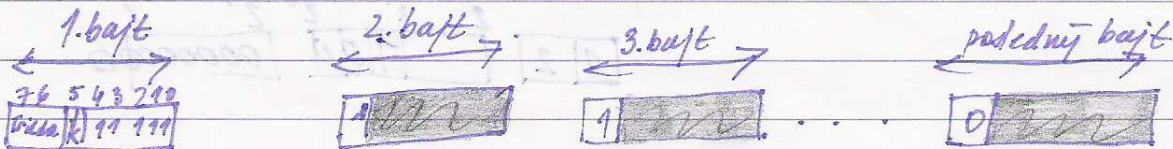
(pole sa deje do jedného bajtu)



- 00 = univerzálny tag (+0)
- 01 = aplikatívny tag (+40<sub>16</sub>)
- 10 = context-specific (+80<sub>16</sub>)
- 11 = privátny tag (+C0<sub>16</sub>)
- 0 = jednoduchý typ (+0)
- 1 = konštruovaný typ (+20<sub>16</sub>)

### POLE TYPU DAT PRE TAGY VÄČŠIE AKO 30

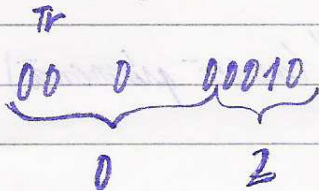
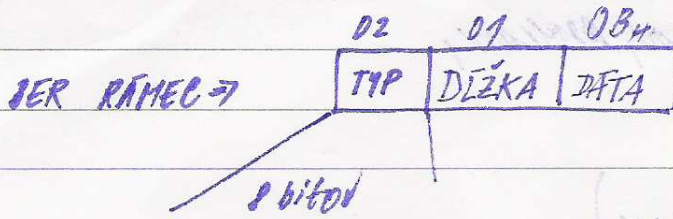
(pole sa skladá z viacerých bajtov)



$$11111_2 = 31_{10} = 1F_{16}$$

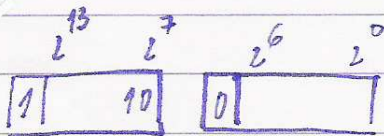
bity tvoriace tag = 11111 + 11111 + ... + 11111

Pr.  $linka ::= INTEGER < ::= -11$



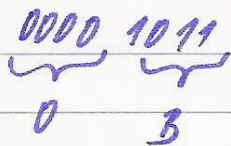
356

0001 1111

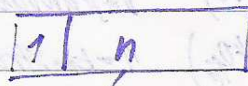


Pr. Integer (11)<sub>10</sub>

(1011)<sub>2</sub>



typ délka data



koliko

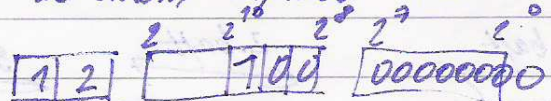
bytov za

typ, aby som

vedel

zaplnit

ak chcem zaplnit 1024



KP  
CV.

Pr 2 TEXT OCTET STRING ::= UT 0

Tag dec. 4 Tag head 4  
releak string

V 25 55H  
T 24 54H  
O 29 4FH

typ DEKA  
04 D3 55 54 4F #  
novi typ

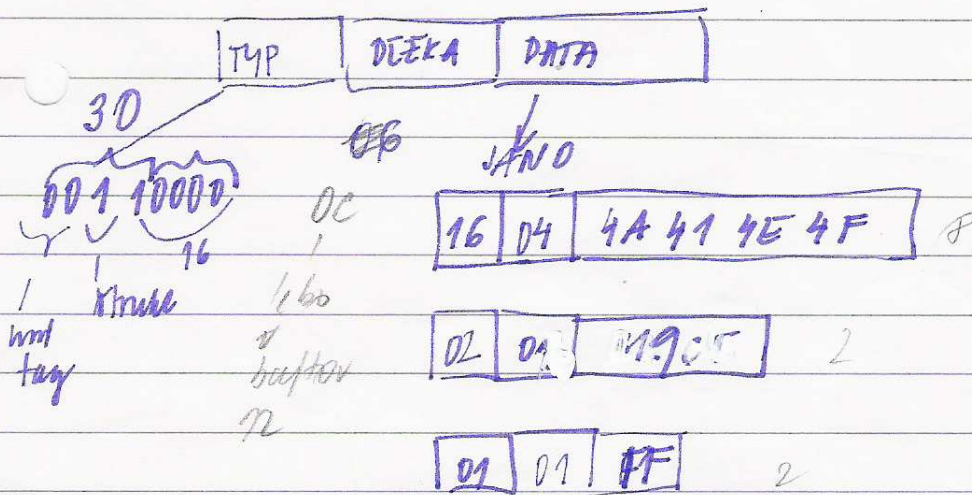
Pr. 3 clovek ::= SEQUENCE {

meno IAP string,  
vek INTEGER  
pohl BOOLEAN }

student clovek ::= { JANO 25 TRUE }  
FFH

J (4A)H  
A (41)16  
N (4E)16  
O (4F)16

sequence (10)16 (16)10



25: 16 = 19

9 · 16 = 9

9 · 16 = 16  
25

30 0C 16 04 4A 41 4E 4F 02 01 19 01 01 FF

# identifikácia objektu

$x_1 x_2 x_3 x_4$

(1) interný objekt IDENTIFIER ::= { 1. 3. 6. 1 }

BER = 10300! 1. etalo = 1 byte v dátach na nasledovnej

konverzijskej 1 byte =  $x_1 \cdot 40_{10} + x_2$

2 byte =  $x_3$

3. byte =  $x_4$

D4

BER

30 18 06 04 2B 06 01 07 04 04 42 AF 67 D6 14 08 42 36 49 49 2D 4B 56 45 54 4

B41