

BAZE PODATKOV

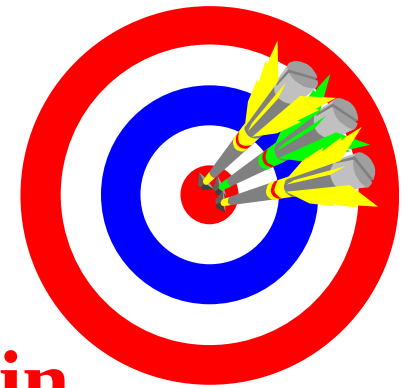
UPRAVLJANJE BAZ PODATKOV

Prof. dr. Vladislav Rajkovič

<http://lopes1.fov.uni-mb.si>

FO

Cilji predmeta:



- **Spoznati proces med analizo sistema in računalniško rešitvijo**
- **Od podatka do podatkovne baze na računalniku**
- **Osnove algoritmov za delo s podatki**
- **Načini modeliranja podatkov**
- **Sistemi za upravljanje baz podatkov (SUBP oz. DBMS)**
- **Organizacijski prijem**
- **Poklici v zvezi bazami podatkov**
- **Sodobne smeri razvoja**

Literatura:



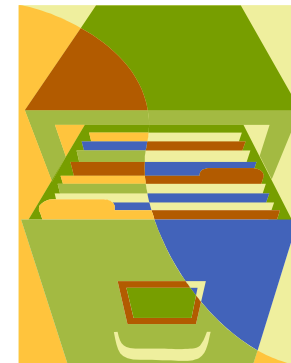
- Mohorič, T., **Uvod v podatkovne baze** (BI-TIM, Lj., 1995) ISBN 961-6046-02-0
- Mohorič, T., **Načrtovanje relacijskih podatkovnih baz** (BI-TIM, Ljubljana, 1995) ISBN 961-6046-05-5
- Mannino, M. V., **Databases: Design, Application Development & Administration, 2nd ed.** (McGraw Hill, Boston, 2004) ISBN 0-07-121489-5
- Post, G. V., **Database Management Systems: Designing and Building Business Applications, 3rd ed.** (McGraw Hill, Boston, 2005) ISBN 0-07-111180-8
- Hoffer, J. A., Prescott, M., McFadden, F. R. **Modern Database Management, 8th ed.** (Pearson, Prentice Hall, 2007) ISBN 0-13-221211-0
- Post, G. V., **Database Management Systems: Designing and Building Business Applications** (McGraw Hill, Boston, 2005) ISBN 0-07-111180-8

Literatura:



- **Thuraisingham, B.M., XML Databases and the Semantic Web (CRC Press, Boca Ranton, 2002) ISBN 0-8493-1031-8**
- **Kifer, M., Bernstein, A., Lewis, P.M., Database Systems: An Application-Oriented Approach, 2nd ed. (Pearson, Boston, 2006) ISBN 0-321-31256-2**
- **Dietrich, S.W., Urban S.D., An Advanced Course in Database Systems: Beyond Relational Databases (Pearson, NJ, USA, 2005) ISBN 0130428981**
- **Siau, K., Contemporary Issues in Database Design and Information Systems Development (IGI Publishing, Hershey, 2007) ISBN 978-159904289-3**
- **Kimball, R. et al., Kimball's Data Warehouse Toolkit Classics, 2nd ed. (John Wiley & sons, 2009) ISBN 978-0470479575**
- **Rob, P., Semaan, E., Databases: Design, Development & Deployment Using Microsoft Access, 2nd ed. (McGraw Hill, Boston, 2004) ISBN 0-07-282658-4**

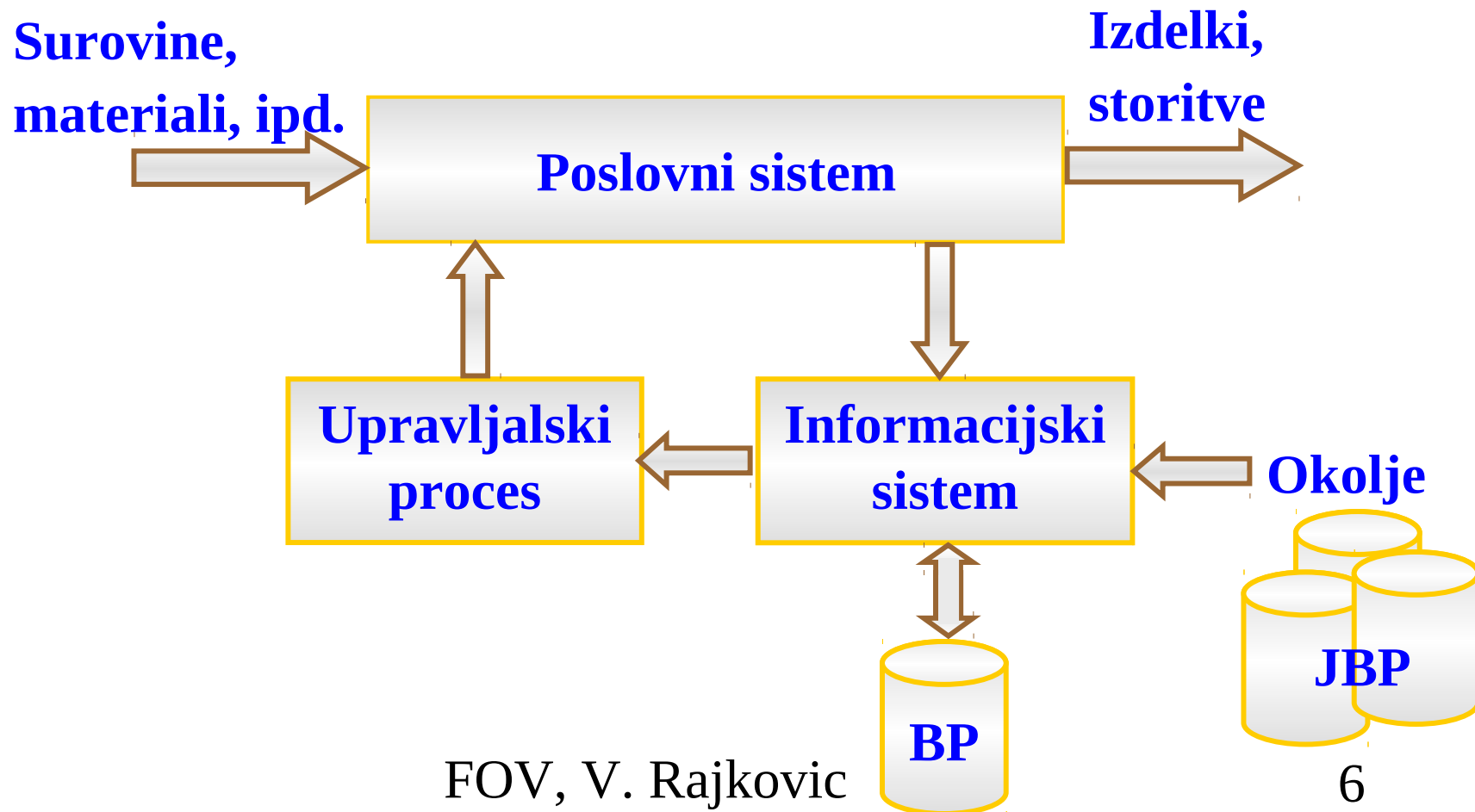
Osnovni pojmi:



- **Podatek** je predstavitev informacije
- **Informacija** je pomen, ki ga pripisujemo podatku
- **Znanje** so podatki oz. informacije, organizirani tako, da omogočajo neposredno reševanje problema
- **Informacijski sistem (IS):**
človek, programi, podatki in računalniško omrežje
- **Baza podatkov (BP)** je mehanizem, večuporabniška, formalno definirana in centralno nadzorovana zbirka podatkov

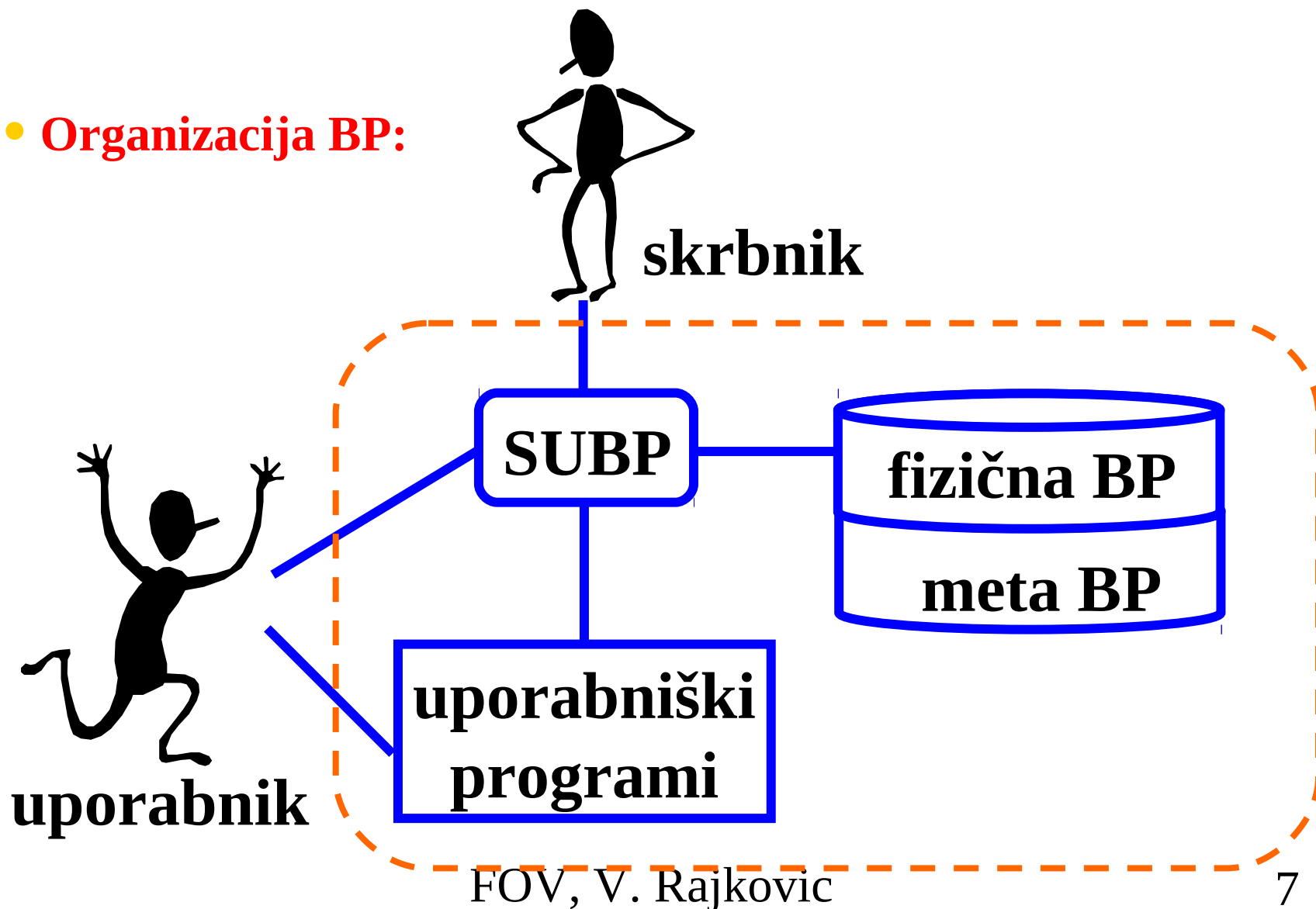
Osnovni pojmi:

- **IS:**



Osnovni pojmi:

- Organizacija BP:





Osnovni pojmi:

- **Podatkovni element (data item)** – atribut, podatek je najmanjša enota podatkovnega tipa, ki lahko samostojno sodeluje v obdelavi
- **Zapis (record)** je nanizanka podatkovnih elementov

Option Base 1

Private Type KUPEC

```
KST as String * 3
Priimek as String * 15
Ime as String * 10
Kredit as Currency
Stanje(3) as Currency
Datum as Date
Nar(10) as NAROCILO
```

End Type

Private Type NAROCILO

```
Sifra as String * 3
Cena as Currency
Kolicina as Integer
```

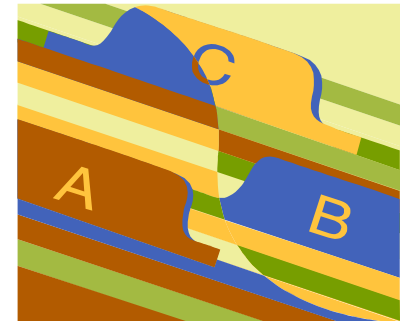
End Type

Osnovni pojmi:



- **Ključ:**
 - **Ključ je podatek oz. del podatka, po katerem zapis iščemo**
 - **Enoznačni ključ je ključ, ki pripada natanko enemu zapisu**
 - **Primarni ključ**
 - **Sekundarni ključ**
 - **Sestavljen ključ**
- **Datoteka je nanizanka zapisov**
- **Organizacija datotek:**
Datoteke organiziramo tako, da optimiziramo delo z njimi

Vrste organizacije datotek:



- **serijska** – nanizanka brez ključa
- **sekvenčna** – sortirana serijska datoteka
npr. po padajočem ali rastočem ključu
- **indeksna** – imamo tabelo indeksov, ki vsebuje ključ in naslov pripadajočega zapisa
- **indeks-sekvenčna** – ključ je del podatka, ki nas pripelje do zapisov, ki imajo ta ključ kot del svojega enoznačnega ključa
- **razpršena (hash)** – ključ in naslov sta povezana z izračunom in ne s podatkovno strukturo (tabelo) kot pri indeksni organizaciji
- **povezana** – naslov naslednika(ov) vsebovan v zapisu

INDEKSNA ORGANIZACIJA DATOTEKE

primer tabele indeksov

vrstica	ključ	naslov	veriga
1	1	6	8 5
2	3	26	0
3	4	15	0
4	5	25	8 6
5	2	40	2
6	6	100	0

Razpršena (hash) organizacija datoteke

KLJUČ

Francija	F	060000
Belgija	B	020000
Nizozemska	NL	141200
Nemčija	D	040000
Vel. Britanija	GB	070200
Irska	IRL	091812
Italija	I	090000
Danska	DK	041100

	0	1	2	3	4	5	6	7	8	9	desna cifra
0	:	A	B	C	D	E	F	G	H	I	
1	J	K	L	M	N	O	P	Q	R	S	
2	T	U	V	W	X	Y	Z	0	1	2	
3	3	4	5	6	7	8	9	+	-	*	
4	/	()	\$	=		,	.	□	[CDC (64)
5]	%	□	□	□	□	□	□	□	□	
6	□	□	-	;	FOV,	V.	Rajkovic				

Razpršena (hash) organizacija datoteke

KLJUČ

Francija	F	060000
Belgija	B	020000
Nizozemska	NL	141200
Nemčija	D	040000
Vel. Britanija	GB	070200
Irska	IRL	091812
Italija	I	090000
Danska	DK	041100



$$60000 \bmod 19 = 17$$

$$20000 \bmod 19 = 12$$

$$141200 \bmod 19 = 11$$

$$40000 \bmod 19 = 5$$

$$70200 \bmod 19 = 14$$

$$91812 \bmod 19 = 4$$

....

(hash code)

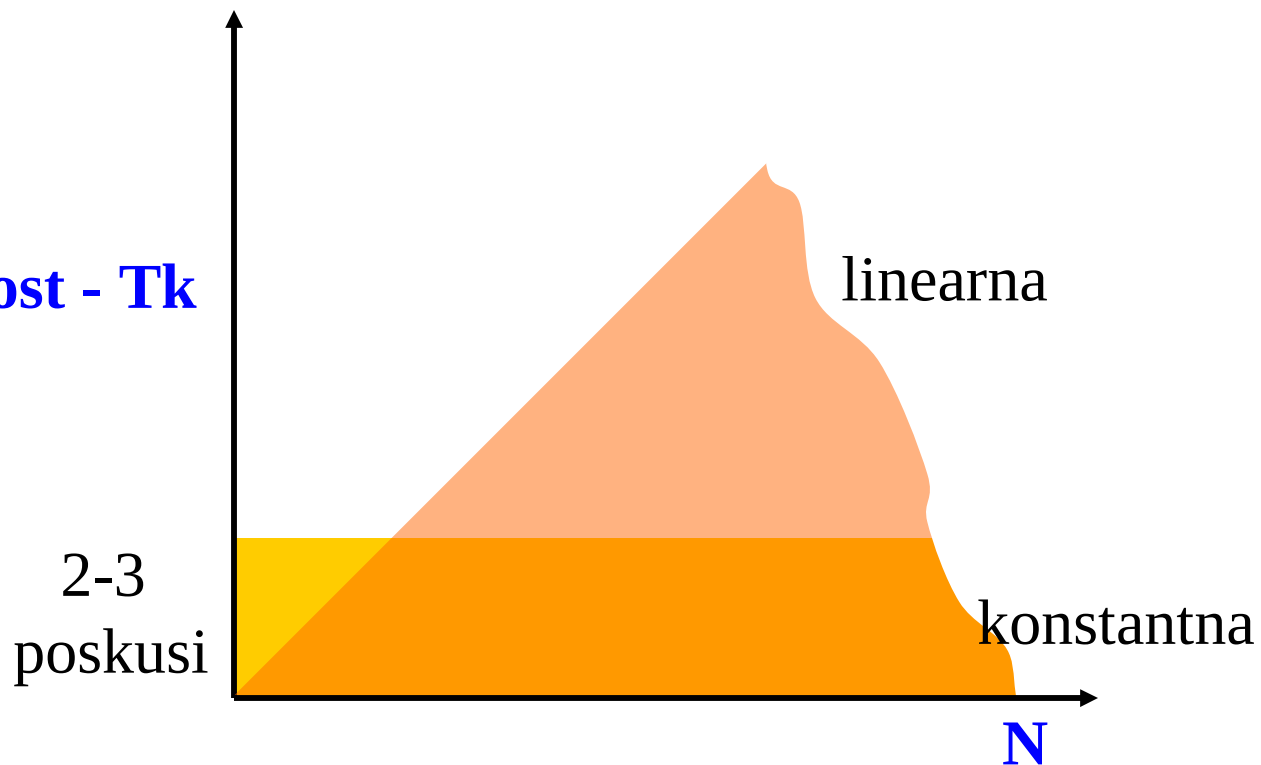
Razpršena (hash) organizacija datoteke

Finska SF 190600 11

NASLOV	KLJUČ	DRŽAVA	Ostali podatki
0			
1			
2			
3	041100	Danska	
4	091812	Irska	
5	040000	Nemčija	
6			
7	190900	Slovenija	
8			
9			
10			
11	141200	Nizozemska	
12	090600	Belgija	
13			
14	070200	Vel. Britanija	
15			
16	090000	Italija	
17	060000	Francija	

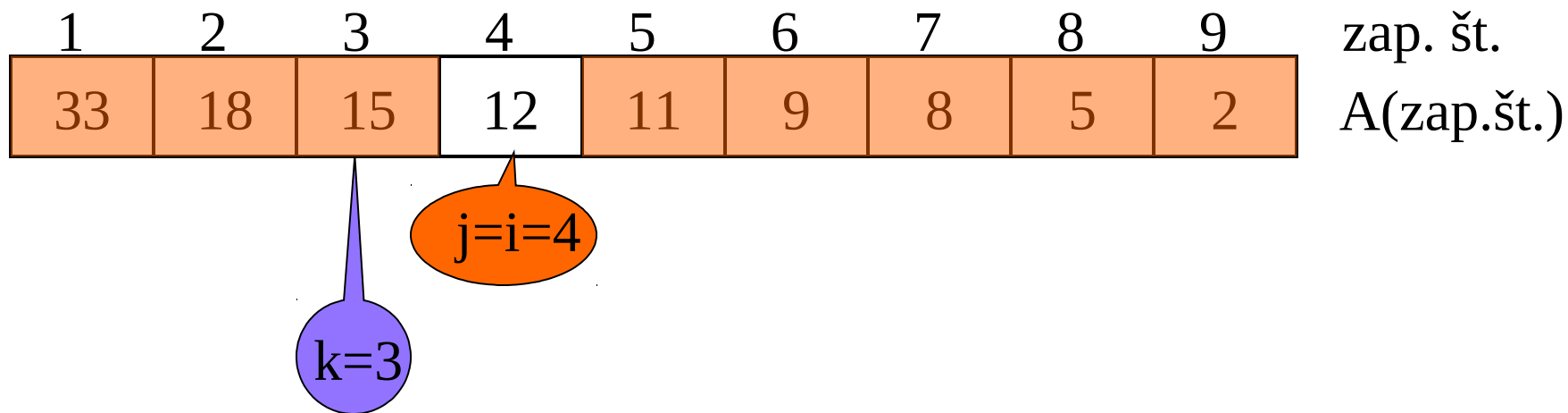
Časovna in prostorska kompleksnost

Časovna kompleksnost - T_k



Prostorska kompleksnost - $M_k: k N + \frac{1}{3} N$

Primer iskanja v urejenem nizu: bisekcija



- $A(k) > x$
 $15 > 12$
- 1) z i označimo prvi zapis, z j zadnjega
 - 2) $k = (i+j) \setminus 2$ - srednji zapis
 - 3) vrednost zapisa, ki ga označuje k primerjamo z iskano vrednostjo $x = 12$
 - 4) **IF** $A(k) > x$ **THEN** $i = k + 1$ **ELSE** $j = k - 1$
 - 5) ponavljamo korake 2, 3 in 4 dokler ne najdemo $A(k)=x$ ali $i > j$ (če zapisa ni)

Računalniški program (Visual Basic)

```
Const n =...  
Dim zapis(1 To n) as UserType1  
Dim x as UserType1  
Dim i as Integer, j as Integer, k as Integer
```

```
Private Sub Bisekcija()
```

```
    x = InputBox("Isci vrednost:")
```

```
    i = 1
```

```
    j = n
```

```
    Do
```

```
        k = (i + j) \ 2
```

```
        If x < zapis(k) Then i = k + 1 Else j = k - 1
```

```
    Loop Until (x = zapis(k)) Or (i > j)
```

```
    If x = zapis(k) Then
```

```
        MsgBox ("Podatek ustreza zapisu: #" & k)
```

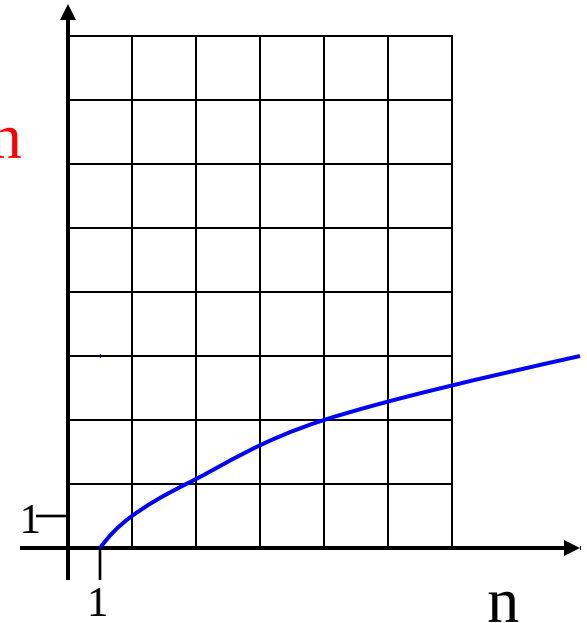
```
    Else
```

```
        MsgBox ("Podatka ni.")
```

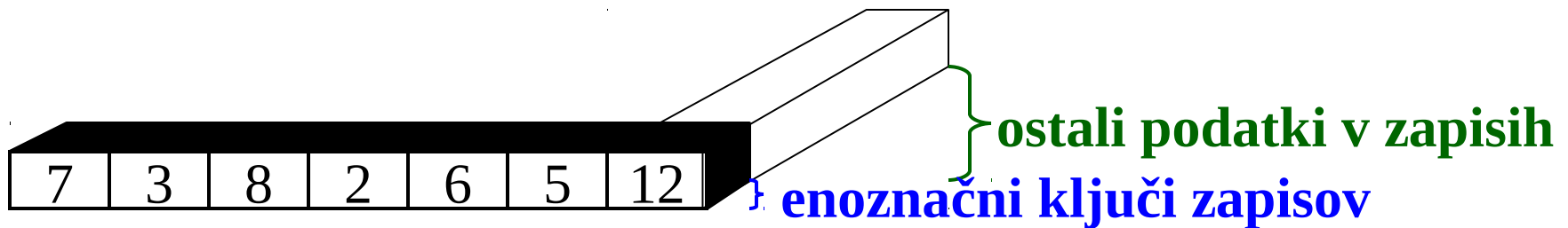
```
    End If
```

```
End Sub
```

Tk $\propto \log_2 n$



Primer sortiranja: mehurčno sortiranje



če je primerjalec $>$ primerjani, ju zamenjaj

```
IF primerjalec > primerjani THEN  
  x = primerjalec  
  primerjalec = primerjani  
  primerjani = x  
ENDIF
```

Računalniški program (Visual Basic)

```
Dim zapisX as Integer, n as Integer
Dim zapis(n) as Integer
Dim i as Integer, j as Integer
```

```
Private Sub Sort
```

```
For i = 0 To n-1
```

```
For j = i + 1 To n
```

```
If CInt(zapis(j)) > CInt(zapis(i)) Then
```

```
zapisX = zapis(i)
```

```
zapis(i) = zapis(j)
```

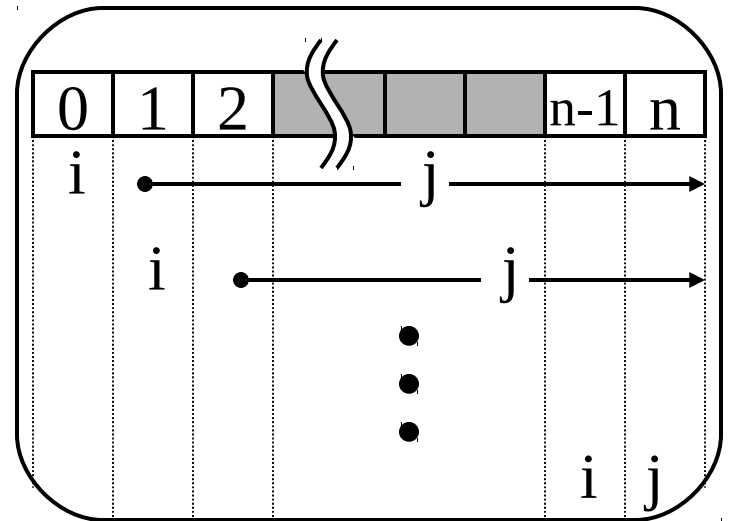
```
zapis(j) = zapisX
```

```
End If
```

```
Next j
```

```
Next i
```

```
End Sub
```



Časovna kompleksnost algoritma: mehurčno sortiranje

```
Dim zapisX as Integer, n as Integer  
Dim zapis(n) as Integer  
Dim i as Integer, j as Integer
```

```
Private Sub Sort
```

```
For i = 0 To n-1
```

```
For j = i + 1 To n
```

```
If CInt(zapis(j)) > CInt(zapis(i)) Then
```

```
zapisX = zapis(i)
```

```
zapis(i) = zapis(j)
```

```
zapis(j) = zapisX
```

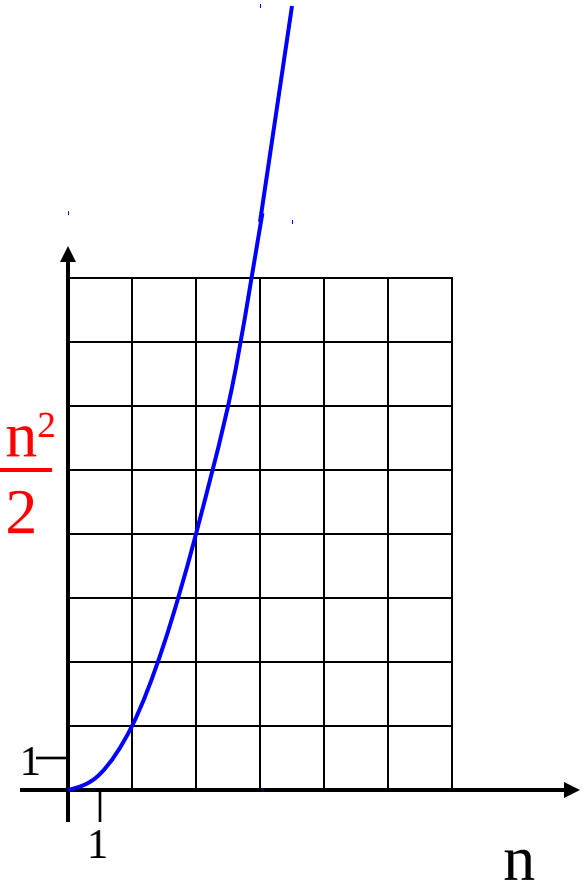
```
End If
```

```
Next j
```

```
Next i
```

```
End Sub
```

$$T_k \propto \frac{n^2}{2}$$

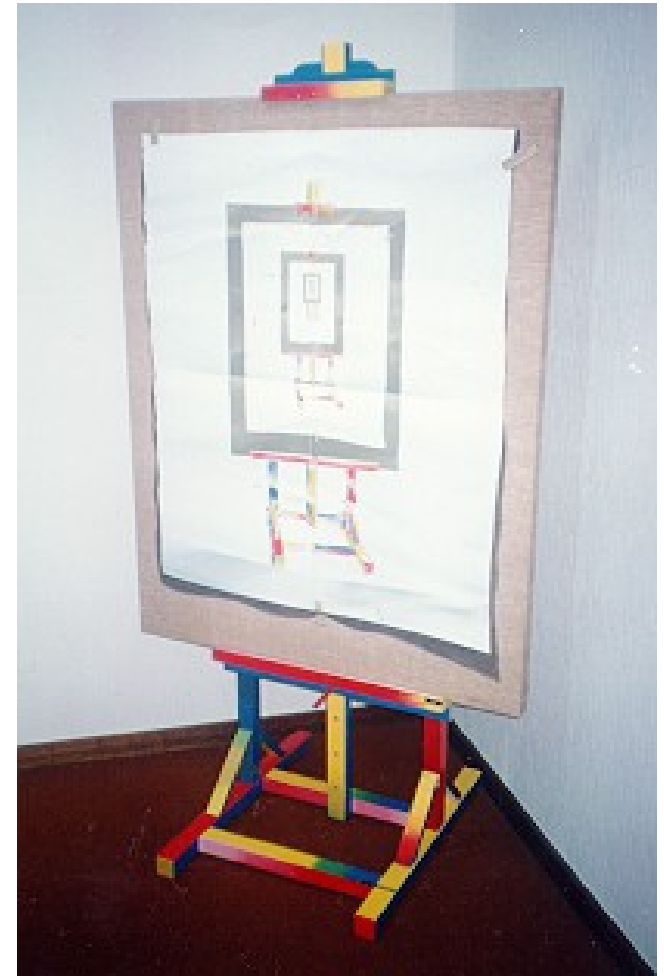


Rekurzija

**Kak objekt je rekurziven,
če je deloma zgrajen ali
definiran s samim seboj.**

Primeri rekurzije:

- slika med dvema vzporednima ogledaloma
- slika TV ekrana predvajana na isti ekran
- v matematiki, npr. $N! = N * (N-1)!$
- rekurzivni program

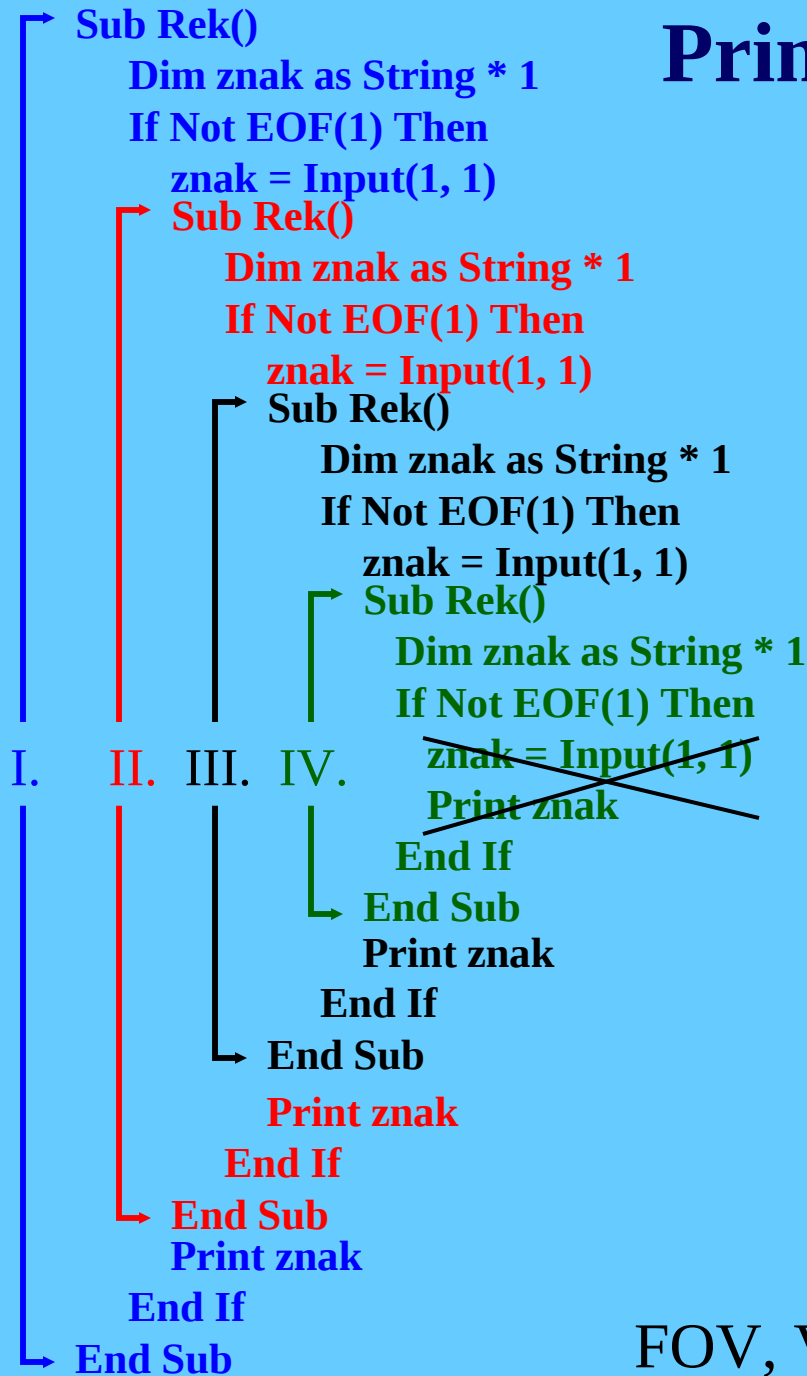


Primer rekurzije - računalniški program (Visual Basic)

```
Sub Gumb_Click()  
    Open "test.txt" for Input as #1  
    Print "Podatki iz datoteke v obratnem vrstnem redu:"  
    Rek  
    Close #1  
End Sub
```

```
Sub Rek()  
    Dim znak as String * 1  
    If Not EOF(1) Then  
        znak = Input(1, 1)  
        Rek  
        Print znak  
    End If  
End Sub
```

Primer rekurzije - rač. program (Visual Basic)



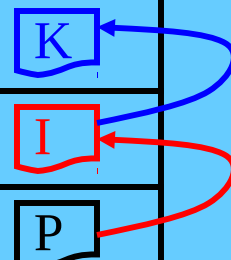
datoteka:

K	I	P	eof
---	---	---	-----

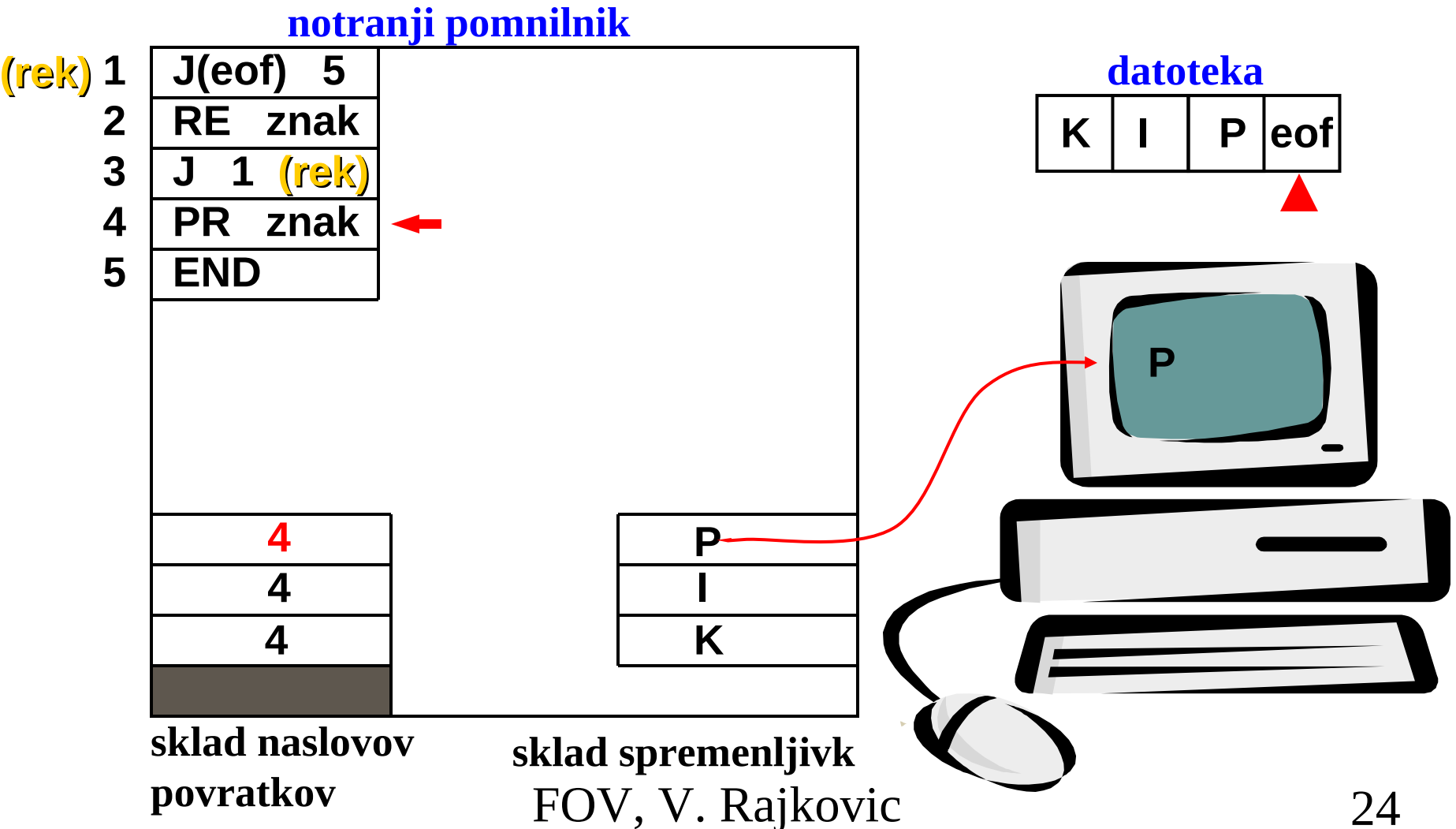


spomin:

eof	ZNAK
ne	K
ne	I
ne	P
da	null



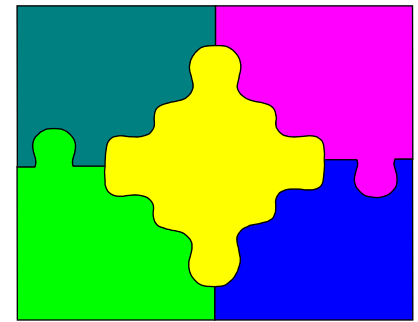
Odvijanje rekurzivnega programa v računalniku



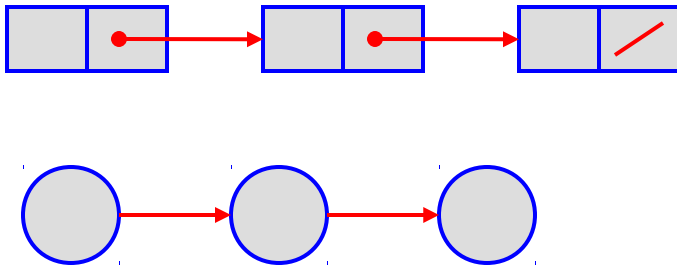
Zaključki (rekurzija):

- **omogoča kompakten zapis algoritma**
- **če je možna preprosta iterativna rešitev, se raje odločimo zanjo**
- **rekurzivni program v principu potrebuje več prostora (sklad naslovov povratkov), v splošnem pa porablja pomnilniški prostor sorazmerno potrebi**
- **prikladna za delo s povezanimi strukturami, v okviru večpredstavnosti in umetne inteligence**

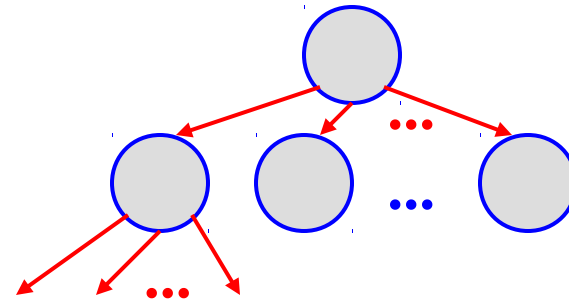
Povezane strukture:



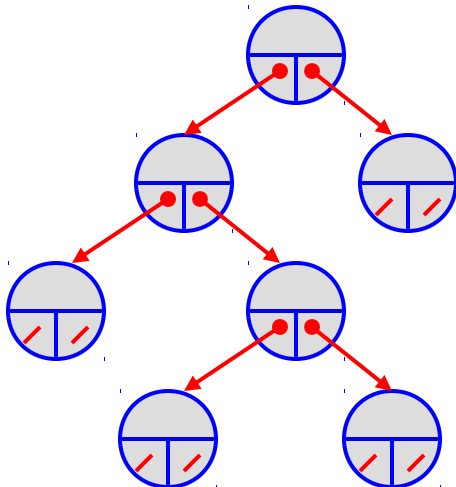
- **Seznam (list)**



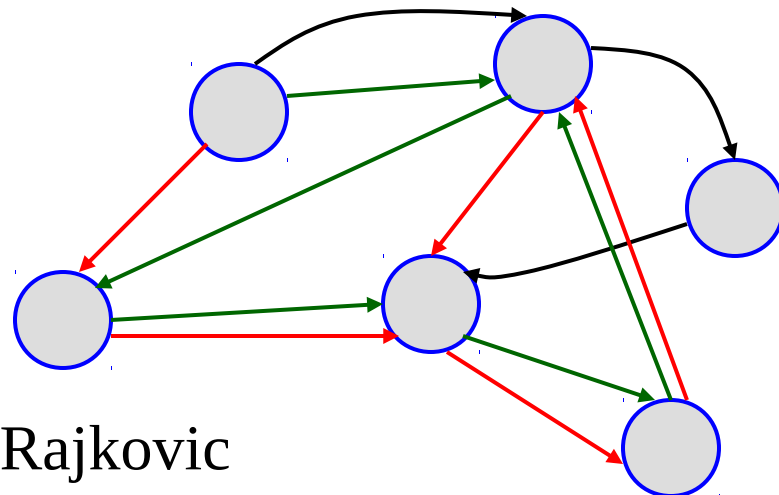
- **Drevo (tree)**



- **Binarno drevo (binary tree)**



- **Graf (graph)**



Naslov	Ključ	mesto	velikost	bazen
1	Turist	/	/	/
2	Slon	1	/	1
3	Hol. Inn	2	/	/
4	Kompas	/	2	3
5	Golf	4	4	4
6	Krim	5	1	2
7	Creina	/	3	5
8	Jelen	7	6	6
9	Lek	/	7	7
10	Lariks	9	9	9

mesto

lastnost	naslov	D
Bled	6	3
Kranj	8	2
Kr. gora	9	10 12
Ljubljana	3	3

velikost

lastnost	naslov	D
velik	5	3
srednji	10	4
majhen	8	3

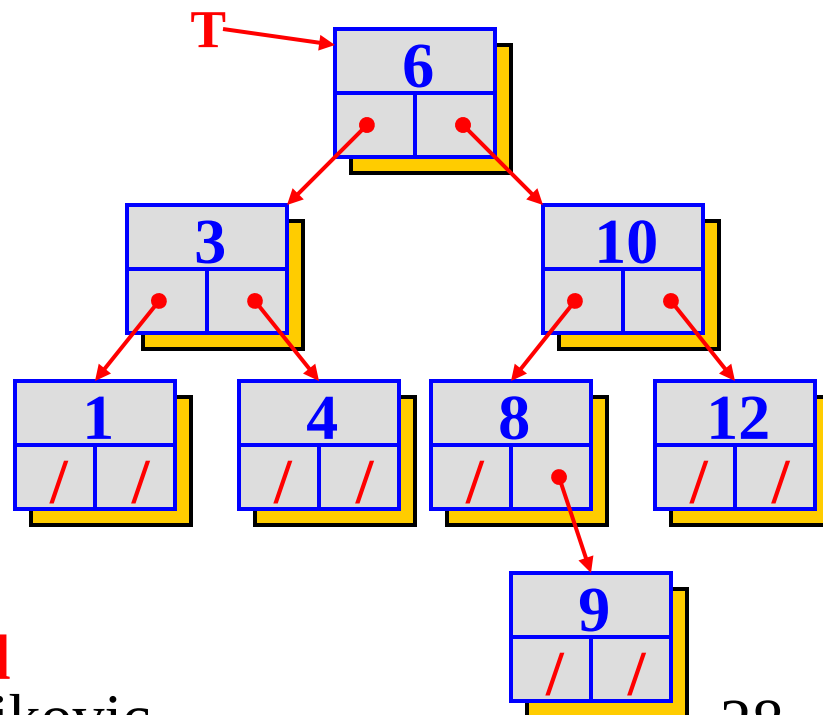
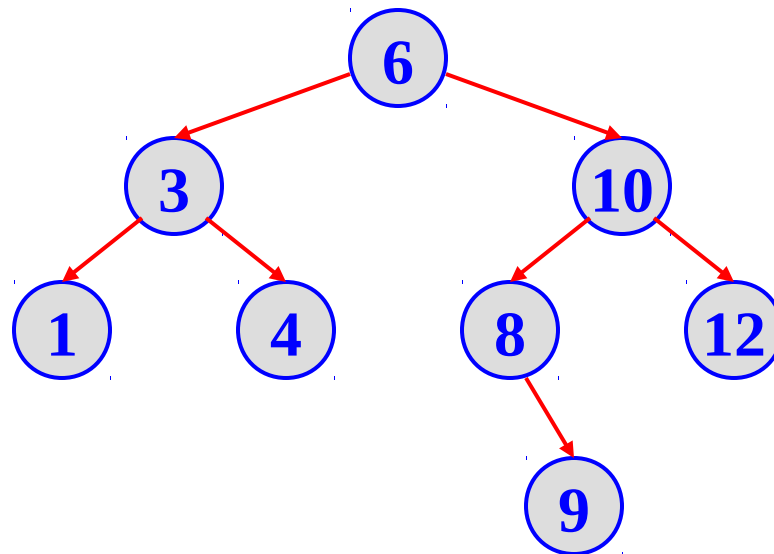
bazen

lastnost	naslov	D
DA	9 10 6	
NE	8	4

Lariks, Kr. gora, srednje velik, ima bazen

Binarno drevo:

```
type vozela = record  
  kljuc: integer;  
  podatek: string;  
  levo, desno: ^vozela;  
end;  
var T: ^vozela;
```



/ ≡ nil

Iskanje podatka X v binarnem drevesu

Poišči:

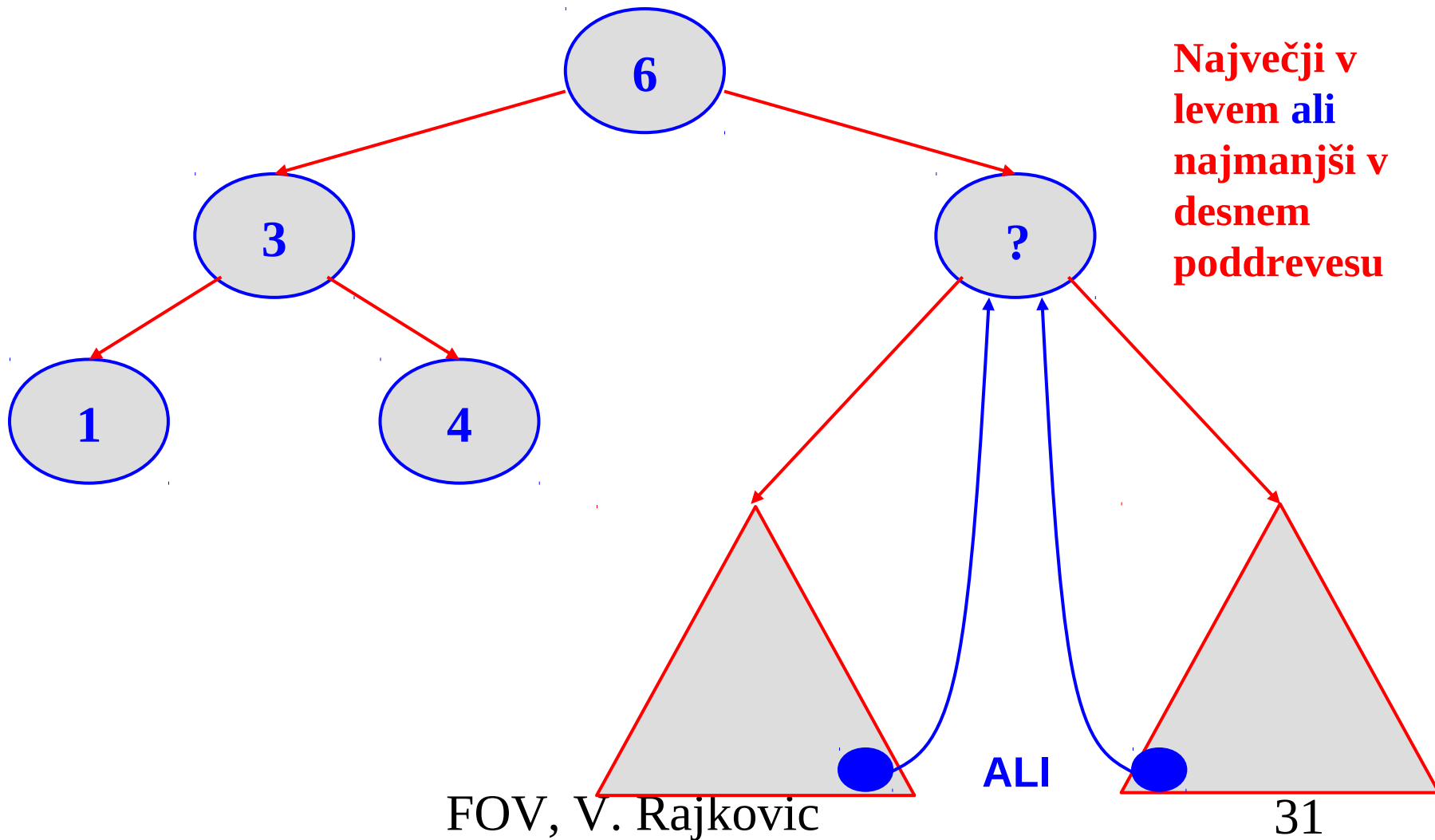
1. če je drevo prazno, X ni v drevesu
2. primerjaj X s korenom drevesa in
 - 2.1 če velja $X = \text{koren}$, X je najden
 - 2.2 če velja $X < \text{koren}$, **poišči** ga v levem poddrevesu
 - 2.3 če velja $X > \text{koren}$, **poišči** ga v desnem poddrevesu

Vstavljanje podatka X v binarno drevo

Vstavi:

1. če je drevo prazno, postane X koren drevesa
2. sicer X primerjaj s korenem
 - 2.1 če velja $X = \text{koren}$, končaj
 - 2.2 če velja $X < \text{koren}$, **vstavi** X v levo poddrevo
 - 2.3 če velja $X > \text{koren}$, **vstavi** X v desno poddrevo

Brisanje podatka X v binarnem drevesu



Brisanje podatka X v binarnem drevesu

1. če X ni v drevesu, drevo ostane nespremenjeno
2. če ima X največ enega naslednika, X preskočimo s prevezavo
3. če ima X dva naslednika
 - 3.1 v levem poddrevesu poiščemo največji element (skrajno desni)
 - 3.2 ta element dvignemo namesto X

Zgodovinski razvoj:

- Luknjana kartica

RTV LJUBLJANA EVIDENCA
IN STATISTIKA NAROČNIKOV
LJUBLJANA, KERSNIKOVA 4
PP 261

POTRDILO 300 DIN

NAROČNIK JE DOLŽAN
PRIJAVITI VSAKO SPREMEMBO
V DESETIH DNEH
STROŠKI INKASA DIN 20

1964 LETO MES. INK. EVID. A
RAJKOVIČ VELJAVIR PRIIMEK IN IME
019 RIB. št. OKLENEGA U. NASELJE — ULICA
LJUBLJANA POŠTA — MESTO

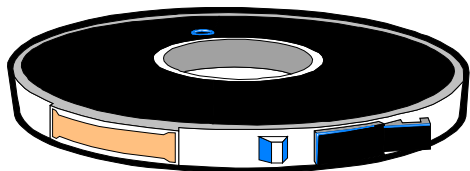
PROSIMO, NAVEDITE PRI VSEH DOPISIH EVIDENČNO ŠTEVILKO

1 SOBNI RADIO	13 NEPOZNAN	2E PLAČANO DNE	FO
2 PRENOSNI RADIO	14 OKL. PLAČILO	91 INKASANTU	93 POŠTI
3 RADIO VGRAJEN V AVTOMOBILU		92 BANKI	94 BLAGAJNI
4 TV NEVGRAJEN UH			
5 TV VGRAJEN UH			

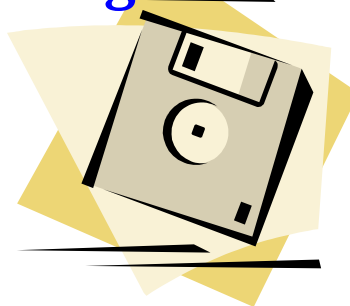
DATUM INKASANTOV POGPIS

KO SPOROČATE SPREMEMBE, UPORABLJAJTE PREDPISANE TISKOVINE. DOBITE JIH NA POŠTI ALI PRI INKASANTU

- Magnetni trak



- Magnetni disk



- Optični disk

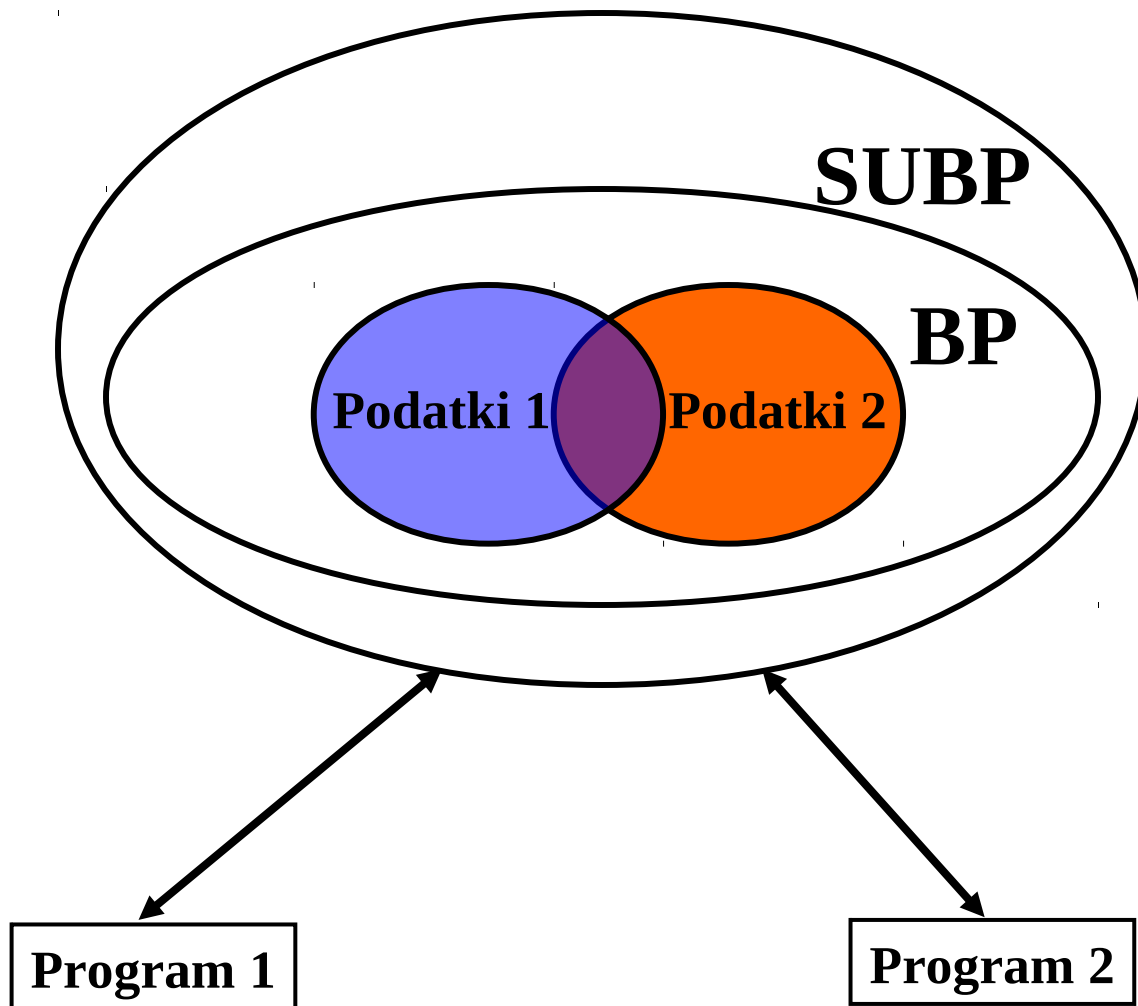


Komunikacijska tehnologija

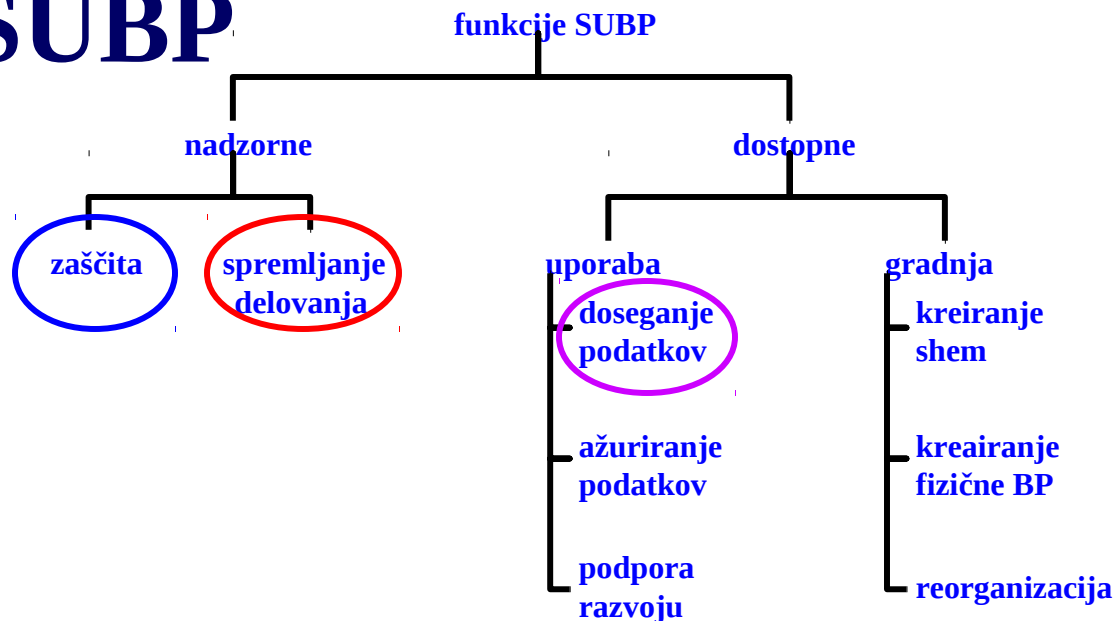
FOV, V. Rajkovic

Kako naprej?

Od datoteke do BP



Kaj prinaša SUBP



- **Zaščita podatkov – fizična, operacijska** (proti napakam: kopije, kontrolne točke ipd.), avtorska (zaupnost: nivoji dostopa)
- **Spremljanje delovanja BP – statična: št.zapisov, kolizij, lastnosti na zapis, ipd., dinamična: uporaba podatkov** (pogostost, odzivni čas, načini doseganja, uporaba diska, ipd.)
- **Doseganje podatkov – SQL (strukturiran poizvedovalni jezik)**

Kaj prinaša SUBP

- **Ažuriranje podatkov – sočasna raba (concurrent usage)**
različni procesi izvajajo različne transakcije v BP
transakcija je osnovna enota uporabnikovega procesa v BP
npr. branje podatkov, obdelava oz. pisanje
sočasno ažuriranje lahko povzroči zagato, slepo ulico (dead lock)
 - Proces A nadzira vir 1
 - Proces B nadzira vir 2
 - Proces A zahteva vir 2 in mora čakati, da ga B sprosti
 - Proces B zahteva vir 1 in mora čakati, da ga A sprosti
- **odpravljanje zagate (roll back)**
- **identifikacija zagate**
- **izogibanje zagatam (npr. samo 1 vir na 1 proces)**

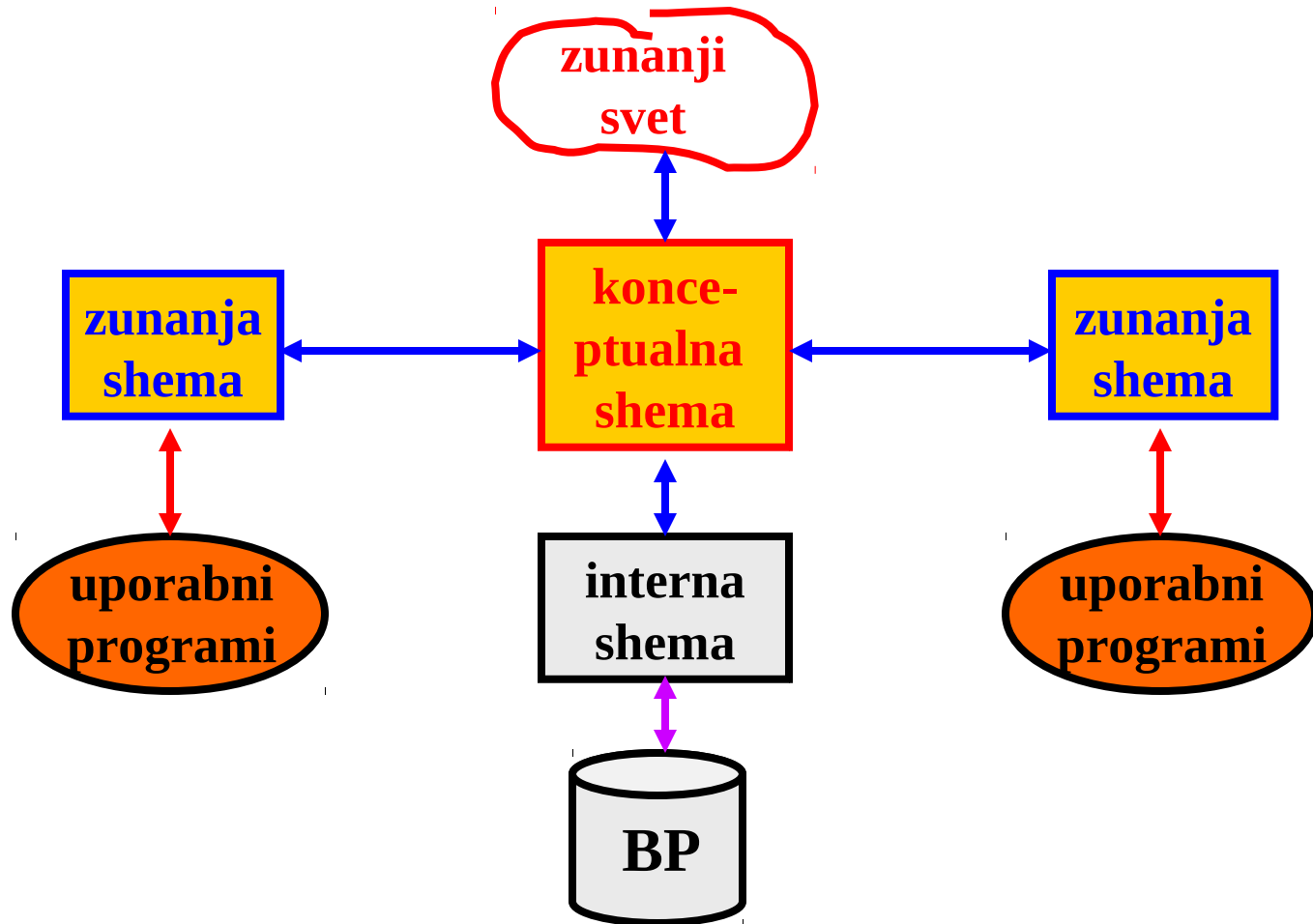
Kaj prinaša SUBP

podatkovni slovar je nastal iz potrebe po dokumentaciji BP

vsebuje npr.:

- **podatki o uporabnikih (kode, status v hierarhiji, ipd.)**
- **opis podatkov**
- **konceptualna shema**
- **zunanje sheme**
- **interna shema**
- **podatki o delovanju**
- **kontrolne točke**
- ...

Konceptualni nivoji SUBP:



Relacijski model podatkov:

- **Relacija**

- matematični izraz za 2D tabelo
- karakterizirajo jo vrstice in stolpci

- **Teorija relacij:**

- izvira iz začetka tega stoletja
- 1970 T.Codd (IBM) predlaga posplošen relacijski model, da bi dosegel podatkovno neodvisnost
- gre za konceptualno shemo podatkov, predstavljeno z relacijami in jeziki za delo z njimi

Relacijski model podatkov:

Ime hotela	Mesto	Velikost	Bazen
Creina	Kranj	S	DA
Golf	Bled	V	DA
Hol. Inn	Ljubljana	S	DA
Kompa	Bled	V	DA
Krim	Bled	M	NE
Lek	Kr. gora	S	DA
Slon	Ljubljana	V	NE
Turist	Ljubljana	M	NE
Jelen	Kranj	M	NE

$$K = D_{\text{IME}} \times D_{\text{MESTO}} \times D_{\text{VELIKOST}} \times D_{\text{BAZEN}}$$

relacija $\subseteq K$

Podmnožica kartezičnega produkta je relacija.

Relacijski model podatkov:

- **stolpec** vsebuje vrednosti atributov
- **vrstica** je n -terica (relacije n stolpcev)
- **število stolpcev** stopnja (degree) relacije
- **število vrstic (n -teric)** moč relacije ali kardinalnost

Relacijski model podatkov:

značilne lastnosti:

- vsi elementi stolpca so iz istega razreda
- stolpci imajo svoja (različna) imena – imena atributov
- vrstni red stolpcev ni pomemben
- n -terice so različne med seboj
- ni pomembna urejenost n -teric

Relacijski model podatkov:

primerjava s tradicionalnimi koncepti podatkov:

- **relacije:** tipi zapisov
- ***n*-terica:** določen zapis
- **atribut:** lastnost, podatkovni element
- **stolpec:** vse vrednosti kake lastnosti
- **stopnja:** število lastnosti v zapisu entitete
- **moč:** število zapisov v datoteki

Normalizacija

Normalizacija pomeni razbijanje relacije na več relacij

Option Base 1

Private Type KUPEC

KST as String * 3

Priimek as String * 15

Ime as String * 10

Kredit as Currency

Stanje(3) as Currency

Datum as Date

Nar(10) as NAROCILO

End Type

Private Type NAROCILO

Sifra as String * 3

Cena as Currency

Kolicina as Integer

End Type

VB zapis

1. normalna oblika

- odstranimo podatkovne agregate
- vsak element je osnoven, če je relacija v obliki 1NF

KUPEC (KST, PRIIMEK, IME, KREDIT, STANJE1, STANJE2, STANJE3,
DATUM)

NAROCILO (KST, SIFRA, CENA, KOLICINA)

Normalizacija

Normalizacija pomeni razbijanje relacije na več relacij

1. normalna oblika

- odstranimo podatkovne agregate
- vsak element je osnoven, če je relacija v obliki 1NF

KUPEC (KST, PRIIMEK, IME, KREDIT, STANJE1, STANJE2, STANJE3,
DATUM)

NAROCILO (KST, SIFRA, CENA, KOLICINA)

2. normalna oblika

- odstranimo delno odvisnost od ključa

NAR (KST, SIFRA, KOLICINA)

CENIK (SIFRA, CENA)

Delo s podatki: Relacijska

algebra

Deluje nad eno ali več relacijami in naredi novo relacijo kot rezultat

SQL – Structured Query Language
– strukturiran poizvedovalni jezik

- V začetku le povpraševalni jezik, temelječ na n-teričnem relacijskem računu
- Kasneje jezik za specifikacijo konceptualne, notranje in zunanje sheme

**KUPEC (KST, PRIIMEK, IME, KREDIT, STANJE1, STANJE2, STANJE3,
DATUM)**

NAR (KST, SIFRA, KOLICINA)

CENIK (SIFRA, CENA)

```
CREATE TABLE KUPEC ( KST CHAR(3),  
PRIIMEK CHAR(15),  
IME CHAR(10),  
KREDIT MONEY,  
STANJE1 MONEY,  
STANJE2 MONEY,  
STANJE3 MONEY,  
DATUM DATE,  
KEY (KST) );
```

```
CREATE TABLE NAR ( KST CHAR(3),  
SIFRA CHAR(3),  
KOLICINA INTEGER,  
KEY (KST, SIFRA) );
```

```
CREATE TABLE CENIK ( SIFRA CHAR(3),  
CENA MONEY,  
KEY (SIFRA) );
```

DEFINICIJA TABELLE

```
INSERT INTO KUPEC VALUES ('109', 'Porenta', 'Janez',  
300000, 100000, 0, 0, '2/2/2000');
```

```
DROP TABLE KUPEC;
```

```
DELETE FROM KUPEC WHERE DATUM='2/2/2000';
```

```
UPDATE KUPEC SET KREDIT=KREDIT*1,05 WHERE KST='109';
```

```
SELECT PRIIMEK, IME FROM KUPEC WHERE KREDIT>10000  
ORDER BY PRIIMEK;
```

```
CREATE VIEW FINANCE AS SELECT KST, KREDIT, STANJE1,  
STANJE2, STANJE3 FROM KUPEC;
```

```
DROP VIEW FINANCE;
```

```
GRANT SELECT, INSERT ON KUPEC TO USER_1, USER_2  
WITH GRANT OPTION;
```


Zaključki (relacijski model):

- Niso izpolnjena vsa pričakovanja.
Nekatere operacije so zelo trd oreh. (NP - polni problemi)
- Podatke je možno modelirati na teoretični osnovi in ne le ad-hoc.
- Možno je raziskovanje drugih zakonitosti
(npr. integriteta in zasebnost).
- Gre za k človeku obrnjeno analizo podatko v duhu normalnih oblik.
- Neproceduralno gledanje
- Osnova za optimizacijo operacij in razumevanje standardnih

Model Codasyl:

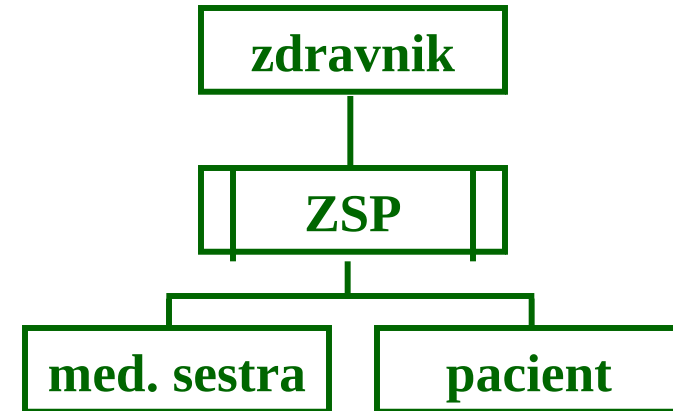
Zapis je osnovni element, ki lahko vsebuje tudi agregate.
Imajo svoje interne številke.

Coset Codasylova množica zapisov

Tip zapisa	Zapisi
Zdravnik	Z1, Z2, Z3, Z4
Med.sestra	S1, S2, S3, S4, S5
Pacient	P1, P2, P3, P4, P5, P6, P7, P8

- ime množice: ZSP
- lastnik: zdravnik
- imena elementov: med.sestra in pacient

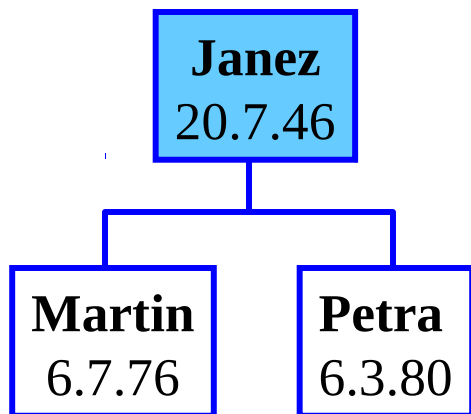
Množica	Lastnik	Elementi
M1	Z1	
M2	Z2	P2, P3
M3	Z3	S1, S2, P1, P4, P8
M4	Z4	S2, S4, P5, P6, P7
...



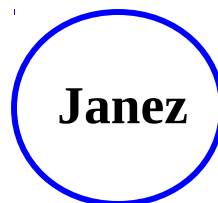
Objektni model podatkov:

Objektni model je množica konceptov, s katerimi predstavimo kompleksne entitete in procese realnega sveta kot objektno shemo.

Družinsko drevo kot podatkovna abstrakcija



Označitev primera objekta



← Predstavitev primera objekta oseba z imenom Vlado

Osnovni koncepti:

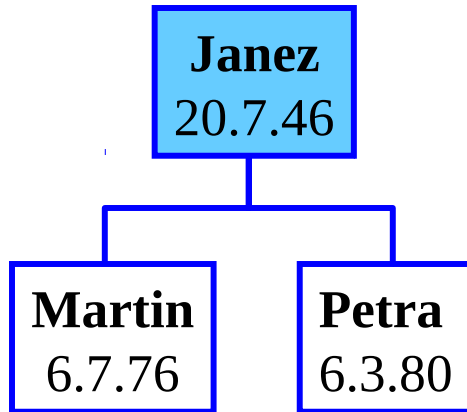
Podatkovna abstrakcija

Vgnezdenje (encapsulation)

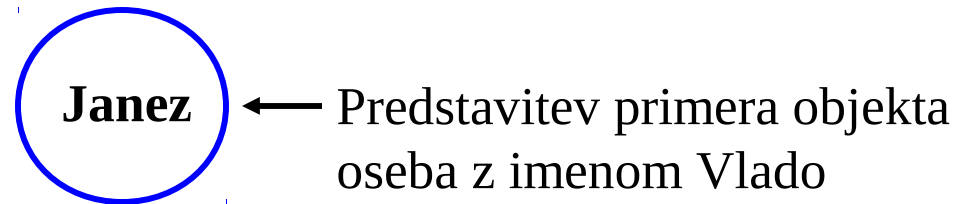
Dedovanje (inheritance)

Podatkovna abstrakcija:

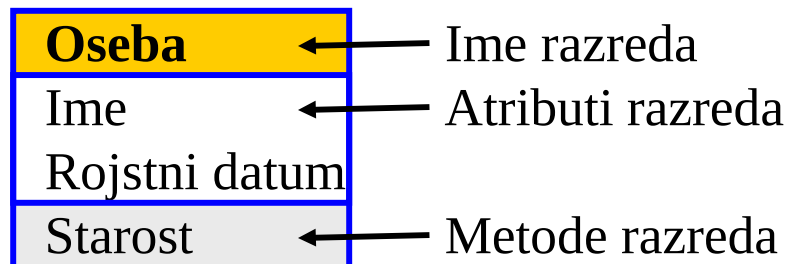
Družinsko drevo kot podatkovna abstrakcija



Označitev primera objekta

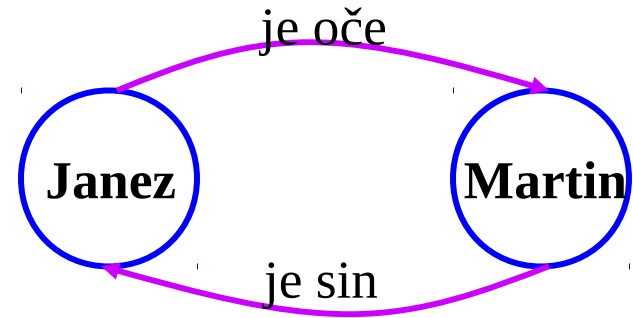
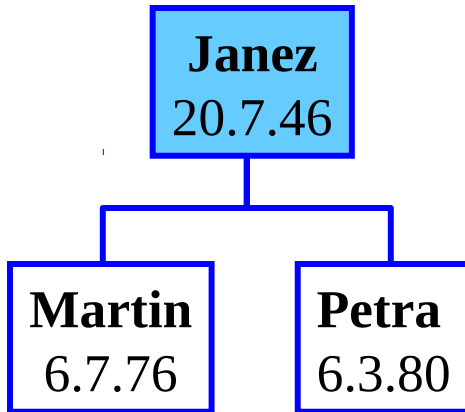


Zapis razreda

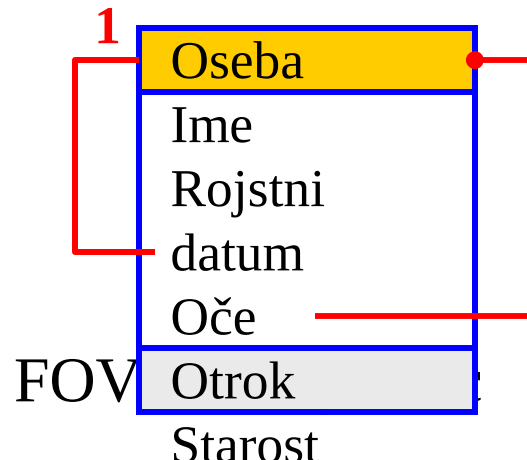


Podatkovna abstrakcija:

Družinsko drevo kot podatkovna abstrakcija

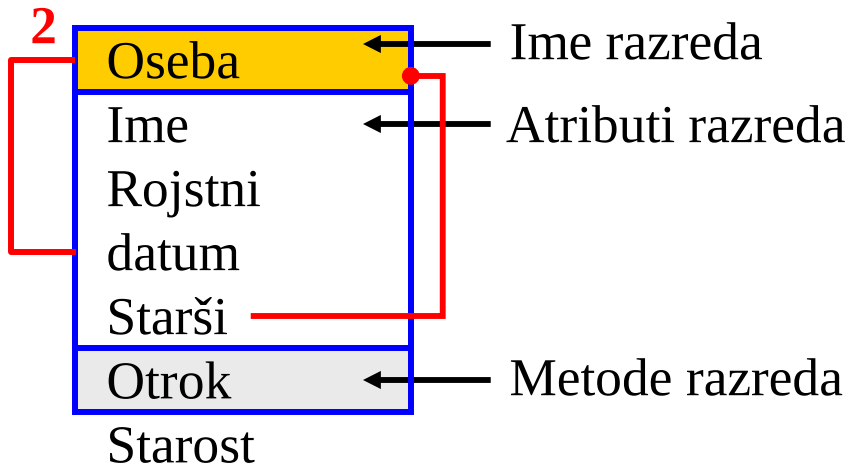


Zapis razreda:

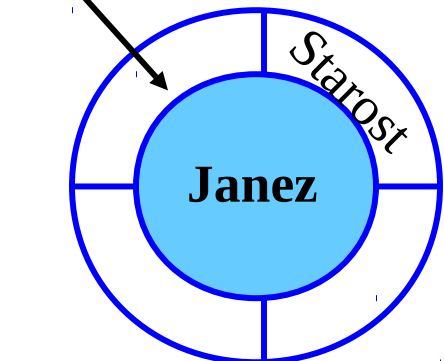


Vgnezdenje:

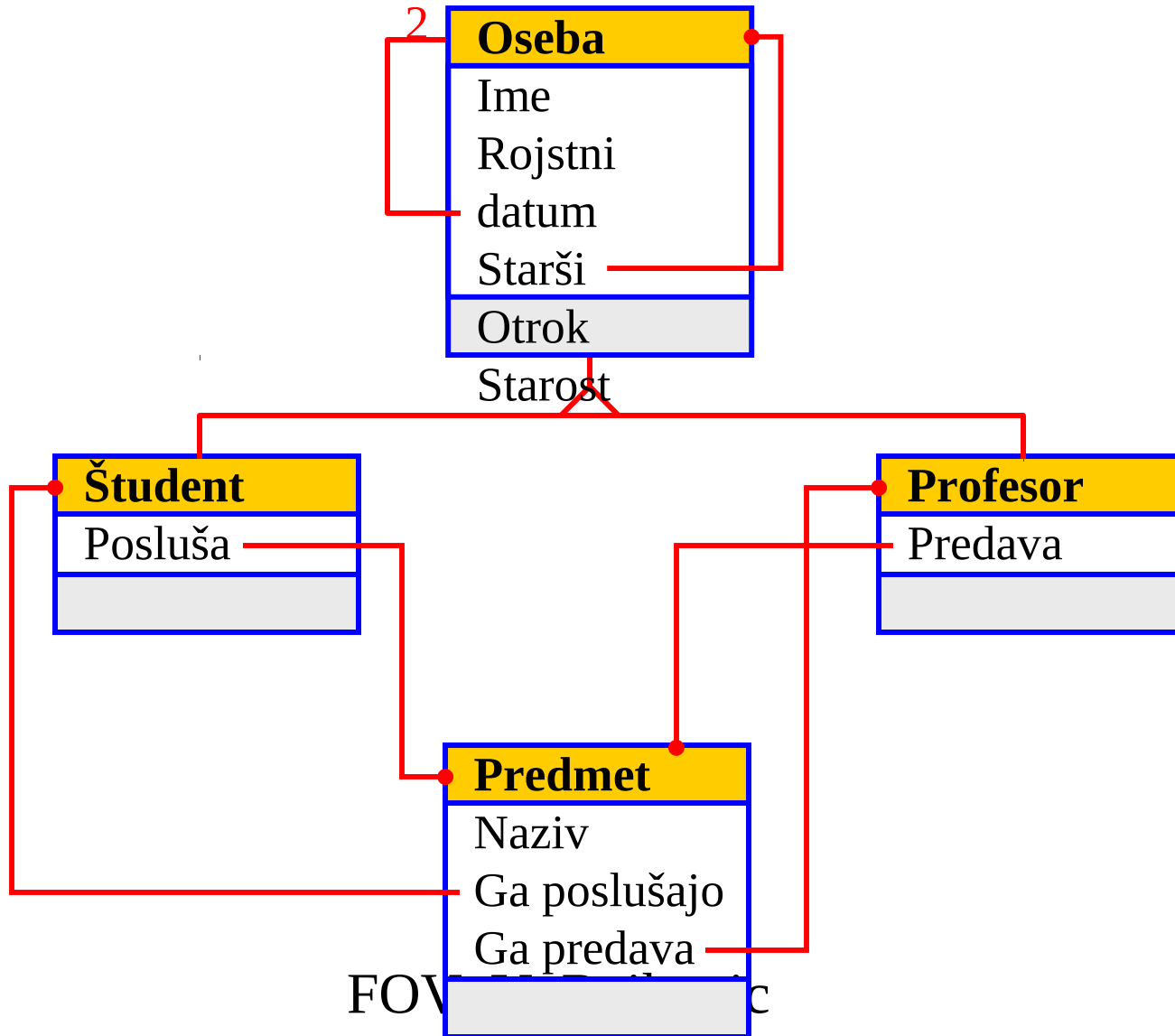
Zapis razreda:



Poljubno število
definiranih metod



Dedovanje - ponovna uporaba:



Poklici:



- **podatkovni analitik** mora ugotoviti:
 - entitete in njihove atribute (lastnosti)
 - kako jih lahko grupiramo
 - kako so entitete med seboj povezane
 - kakšne so zahteve po zasebnosti
 - ...
- **skrbnik (administrator) BP**
 - koordinira delo v zvezi z življenjem BP
 - sledi delovanju BP
 - načrtuje morebitne spremembe
 - pomaga uporabnikom
 - vodi komisijo za BP
 - ...
- **komisija za BP** je koordinativno telo, v katerem sodelujejo predstavniki različnih vrst uporabnikov

Kriteriji za izbor SUBP:



1. zunanje karakteristike

1.1 nabavne karakteristike

- finančni pogoji
- šolanje

1.2 ostale zunanje karakteristike

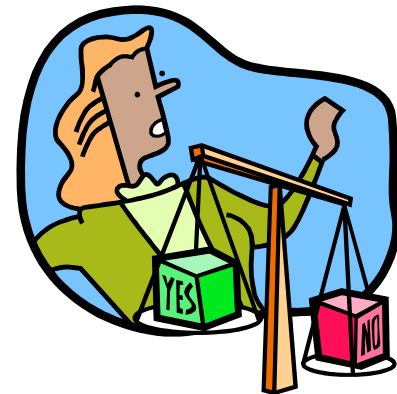
- razvoj
- reference
- izkušnje

2. notranje karakteristike

2.1 jeziki za delo s SUBP

- definicijski jeziki
- SQL

Kriteriji za izbor SUBP:



2.2 varovanje podatkov

- sočasno delo
- zaščita pred nepooblaščenno uporabo
- zaščita pred uničenjem

2.3 organizacija podatkov

- logični model
- fizični model
- standardizacija
- podatkovni slovar

2.4 učinkovitost in zahtevnost

- računalniška
- časovna

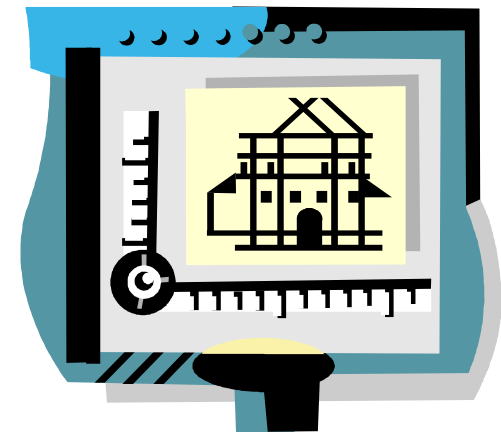
3. ostale karakteristike



Razvoj IS in BP:

Osnovni koncept:

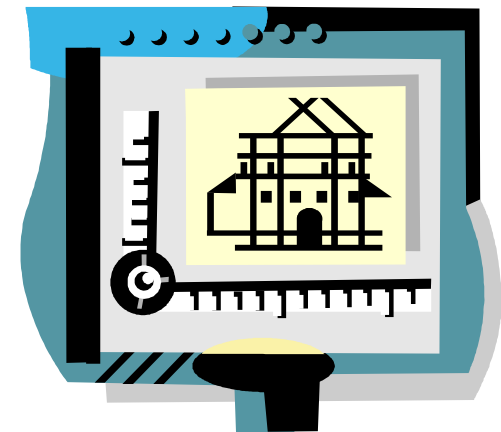
1. strateške usmeritve poslovnega sistema
2. obstoječe stanje informatike
3. analiza sistema
 - 3.1 analiza poslovnega sistema
 - 3.2 analiza organiziranosti
 - 3.3 analiza arhitekture IS
4. podatkovni model
5. opredelitev procesov
6. tehnološki projekt
7. plan razvoja IS
8. kadri in organiziranost informatike
9. finančna sredstva po fazah



Razvoj IS in BP:

Udejanjanje BP:

1. analiza podatkov
2. konceptualni model podatkov
3. ocena transakcijske intenzivnosti
4. izbira SUBP-ja
5. izdelava podatkovnega slovarja
6. vzpostavitev BP
7. skrbništvo nad BP
8. izobraževanje uporabnikov



Smeri razvoja:

- **integralni poslovni dokument**
- **podatkovna skladišča - DWH**
- **javne baze podatkov**
- **multimedijske baze**
- **objektne baze**
- **CAD-CAM baze**

