



# Kazalci

Andrej Bagon



# Uvod

---

- Podatkovni tip, ki spremeni metode programiranja
- Statičen način programiranja – problem teče iz vrha navzdol
- Dinamičen način programiranja – struktura podatka je dvodelna. Podatki so ločeni od kazalcev



# Prednosti

---

- Velik prihranek pomnilnika
- Realizacija pravih podatkovnih struktur



# Slabosti

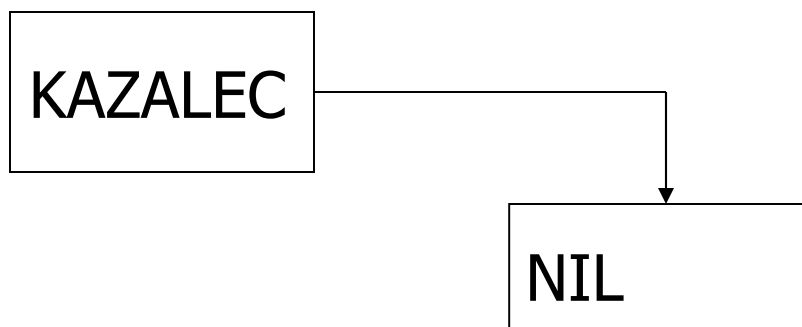
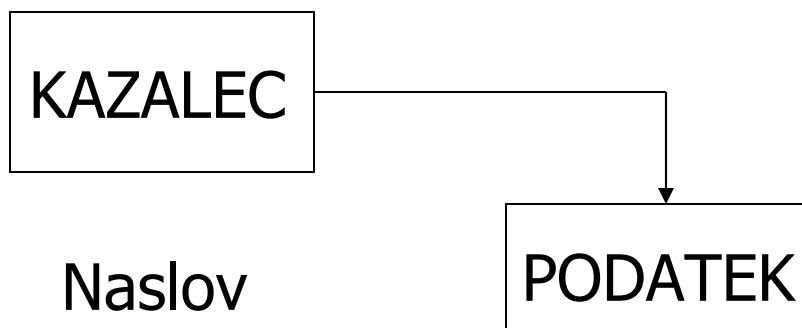
---

- Težka začetna faza
- Včasih so kazalci nerazumljivi



# Predstavitev

---



Inicializiran in  
neinicializiran kazalec



# Deklaracija

---

- Posredno
- Rekurzivno

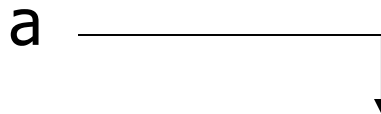


# Deklaracija

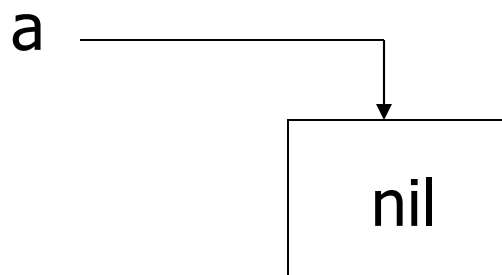
---

Posredno naslavljanje  
var

```
a:^integer; { kazalec na objekt tipa INTEGER }  
b:integer;
```



```
new(a);  
ali  
a := @b;
```



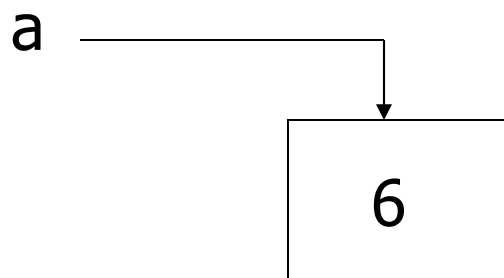
povzroči inicializacijo  
pojavo objekta, ki je  
po strukturi definiran  
posredno preko  
spremenljivke a.



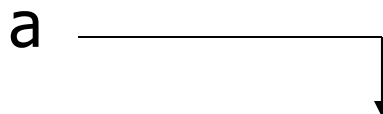
# Deklaracija

---

$a^{\wedge} := 6$



`dispose(a);`



klic povzroči brisanje  
podatka iz pomnilnika –  
obratna operacija od new





# Operacije nad kazalci

---

- `new(a)`
- `dispose(a)`
- `a^:=5;`
- `a := b;`
- `inc(a^);`
- `dec(a^);`



# Deklaracija

---

Rekurzivno naslavljanje

type

kaz = ^vozel;

vozel = record

    stevilo:integer;

    naslednjik:kaz;

end;

var

    a:kaz;

type

    vozel = record

        stevilo:integer;

        nasl:^vozel;

end;

var

    a:vozel;