

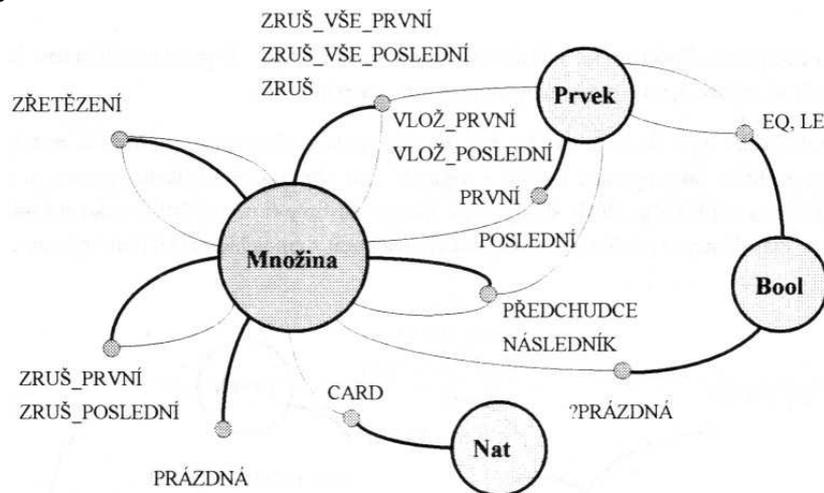
6. Specifikace a implementace datových typů

Úvod

Datový typ je určen přípustnými druhy dat a operacemi, které s daty můžeme provádět.
 (vložit definici strana 107 dole)
 (Bool typ strana 112)

Specifikace vybraných datových typů

Uspořádaná množina

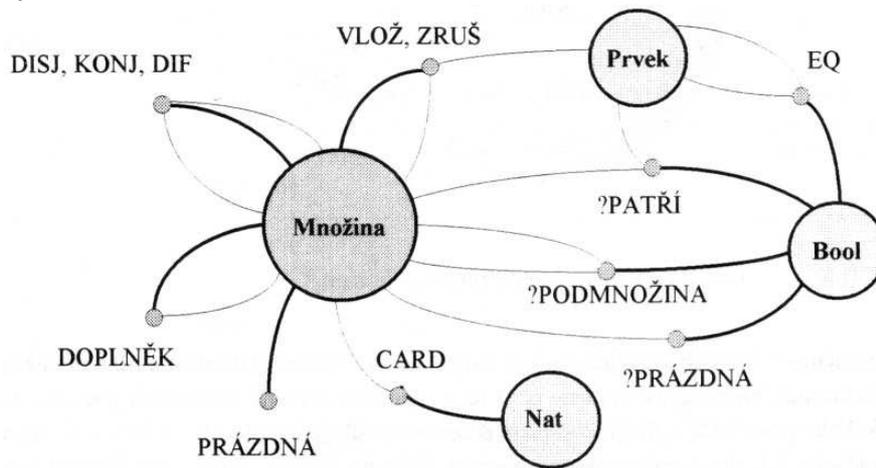


obr. 5.4 Signatura datového typu Uspořádaná množina

Operace datového typu *Uspořádaná množina* jsou:

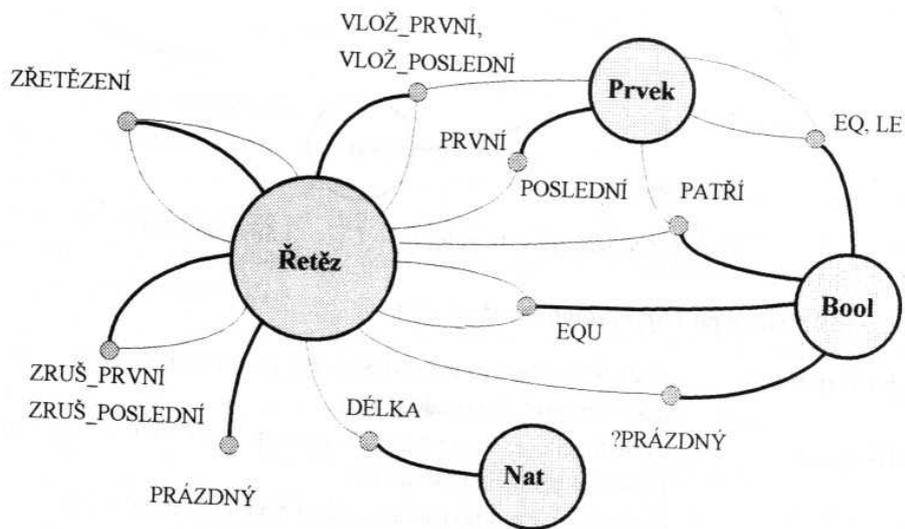
PŘEDCHUDCE (p,m)	výsledkem operace je největší prvek $p_p \in m$ takový, že $p_p \leq p$ (nalezení přímého předchůdce)
NÁSLEDNÍK (p,m)	výsledkem operace je nejmenší prvek $p_n \in m$ takový, že $p \leq p_n$ (nalezení přímého následníka)
PRVNÍ(m)	poskytnutí prvního (minimálního) prvku p v množině m
POSLEDNÍ(m)	poskytnutí posledního (maximálního) prvku p v množině m
ZRUŠ_PRVNÍ (m)	zrušení prvního (minimálního) prvku v množině m
ZRUŠ_POSLEDNÍ (m)	zrušení posledního (maximálního) prvku v množině m
ZRUŠ_VŠE_PRVNÍ(p,m)	zrušení všech prvků $p_i \leq p$
ZRUŠ_VŠE_POSLEDNÍ(p,m)	zrušení všech prvků $p \leq p_i$
ZRUŠ (p,m)	zrušení prvku p v množině m
LE(p1,p2)	porovnání prvků $p1 \leq p2$
EQ(p1,p2)	relace ekvivalence prvků p1, p2
ZŘETĚZ (m1,m2)	zřetězení množin m1, m2, tj. sjednocení takové, že platí $POSLEDNÍ(m1) \leq p_i$ pro všechna $p_i \in m2$ (všechny prvky množiny m2 jsou větší než maximální prvek množiny m1 \Rightarrow následují za posledním prvkem m1).
VLOŽ_POSLEDNÍ(p,m)	zařazení prvku p tak, že $\forall p_i \in m \quad p_i \leq p$ (prvek p je posledním prvkem v m)
VLOŽ_PRVNÍ(p,m)	zařazení prvku p tak, že $\forall p_i \in m \quad p \leq p_i$ (prvek p je prvním prvkem v m)
PRÁZDNÝ (m)	vytvoření prázdné množiny m,
?PRÁZDNÝ(m)	test prázdné množiny m,
CARD (m)	určení mohutnosti množiny m,

Neuspořádaná množina 130

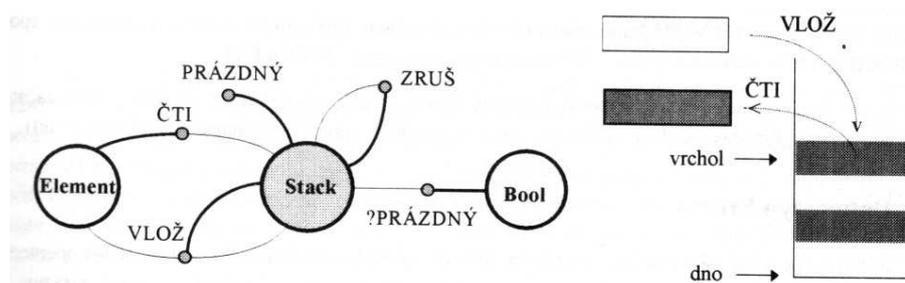


obr. 5.13 Signatura datového typu Neuspořádaná množina

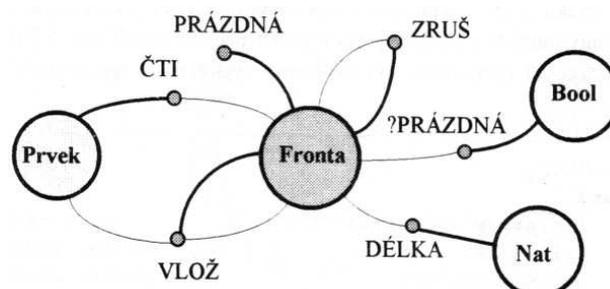
Řetěz (124)



Zásobník (125)

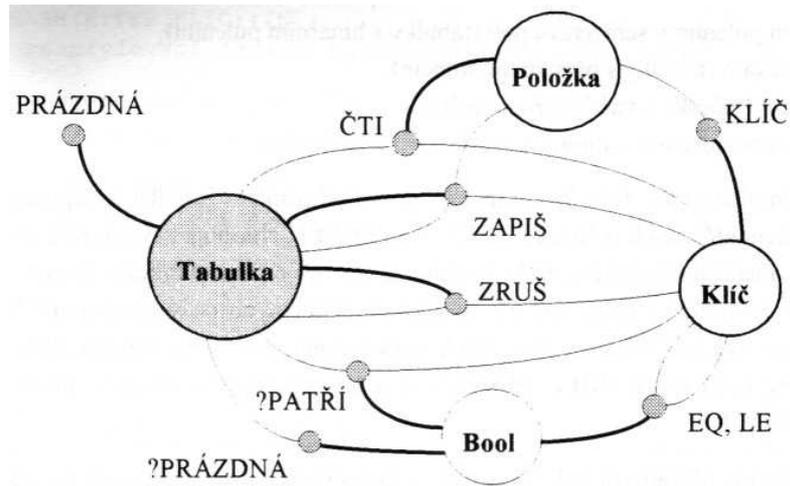


Fronta (127)



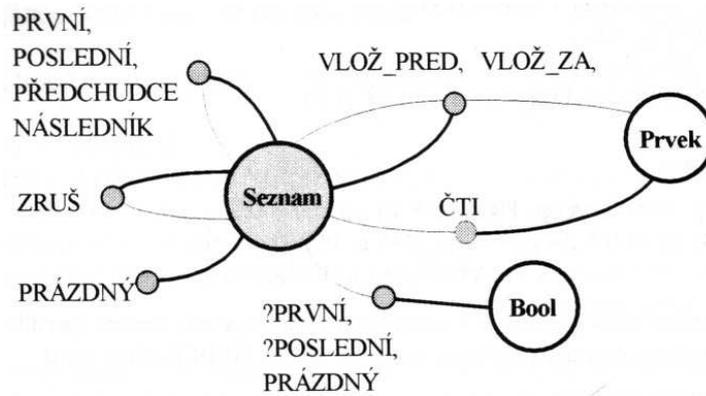
obr. 5.9 Signatura datového typu Fronta

Tabulka (143)



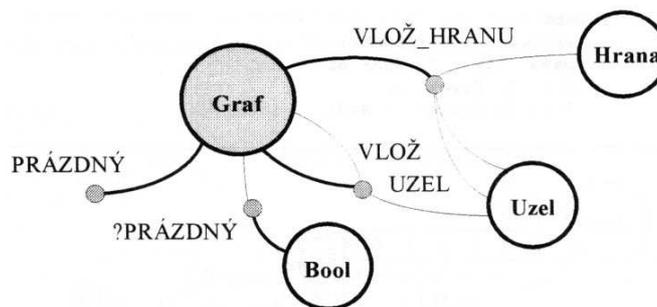
obr. 5.23 Signatura datového typu Tabulka

Seznam(147)



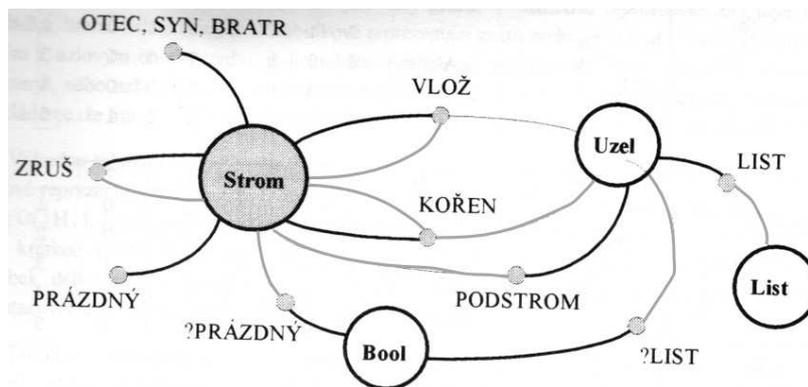
obr. 5.24 Signatura datového typu lineární Seznam

Graf (152)



obr. 5.31 Signatura datového typu Neorientovaný graf

Strom



obr. 5.32 Signatura datového typu Strom

Implementace datových typů

- s konstantní délkou (staticky = pole)
- s proměnlivou délkou (dynamicky = spojové seznamy)

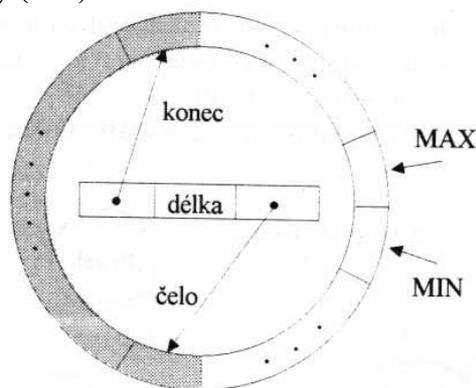
Množina = pole prvků s definovanými operacemi

Řetěz = umí vkládat/mazat pouze ze začátku nebo konce řetězu

Zásobník = vkládá se pouze na konec a z konce se také odebírá

Fronta = spojový seznam (z jednoho konce se odebírá a druhým koncem se přidává)

- Kruhová reprezentace fronty (129)



Tabulka = tabulka(řádky) struktur(sloupce) = (např. pole záznamů)

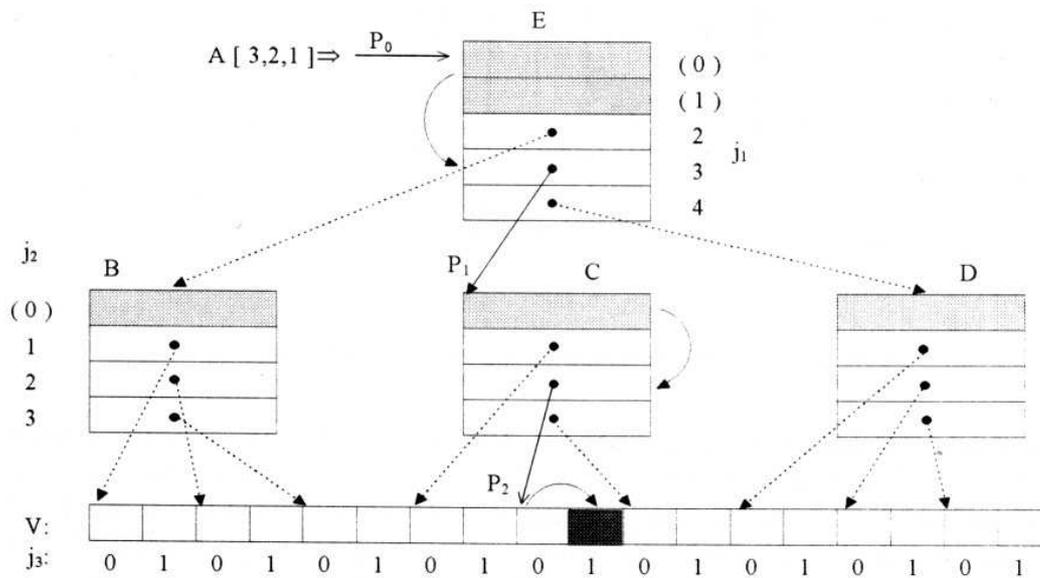
Seznam = tabulka nebo spojový seznam

Graf = záleží, co všechno nás zajímá

- stačí matice sousednosti
- incidenční matice
- tabulka hran
- seznam uzlů a hran ...

Vícerozměrné pole

- jako vícerozměrné pole (array)
- více spojových seznamů (např. Soustava Iliffových vektorů)
Soustava iliffových vektorů (138)



- jeden spojový seznam a přepočítávání pozice

Operační a paměťová složitost implementace

- paměťová záleží na:
 - typech položek
 - plánovaném rozsahu
 - překladači
 - konkrétním návrhu
- operační:
 - záleží na konkrétní implementaci
 - můžeme vytvářet pomocné struktury nad spojovými seznamy (např. Binární stromy)

