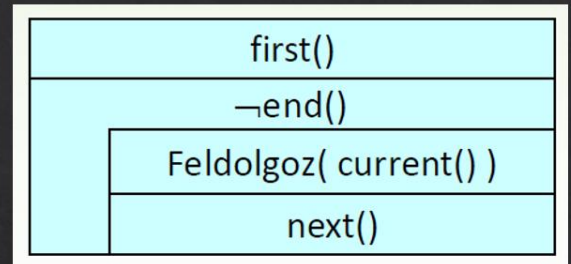


4. OEP Gyakorlat

Felsoroló (3. előadás)



- ❑ Egy gyűjtemény feldolgozásához szükségünk van a gyűjtemény felsorolására (bejárására).
- ❑ Erre a felsorolásra úgy tekinthetünk, mint a gyűjtemény elemeiből képzett **véges sorozatra**, amely az alábbi **műveletekkel** rendelkezik:
 - **first()** : rááll a sorozat első elemére, azaz elkezdi a felsorolást
 - **next()** : rááll a sorozat soron következő elemére, azaz folytatja a felsorolást
 - **$l := \text{end}()$ ($l:\mathbb{L}$)** : megmutatja, hogy a sorozat, azaz a felsorolás végére értünk-e
 - **$e := \text{current}()$ ($e:E$)**: visszaadja a sorozat, a felsorolás aktuális elemét

Szekvenciális input-, output fájl

Szekvenciális fájlok

- A **szekvenciális inputfájl** elnevezést olyan sorozatra értjük, amelynek mindig az első elemét lehet kiolvasni.
- Az olvasás egy $st, e, x := \text{read}(x)$ értékadás, amelyet az $st, e, x : \text{read}$ szimbólummal rövidítünk. Ebben
 - $x : \text{infile}(E)$ a szekvenciális inputfájltra hivatkozó változó
 - $e : E$ a fájl elejéről kiolvasott (leszakított) elemet tartalmazó változó
 - $st : \text{Status} = \{\text{abnorm}, \text{norm}\}$ az olvasás sikerességét jelző változó

```

if x = <> then
  st := abnorm
else
  st := norm;
  e := x1;
  x := <x2, ..., x|x|>
endif

```

- A **szekvenciális outputfájl** elnevezést egy olyan sorozatra használjuk, amelynek végéhez lehet új elemet hozzáírni, és kezdetben az $x := \langle \rangle$ értékadással üresre lehet állítani.
- Az írás egy $x := \text{write}(x, e)$ értékadás, amelyet az $x : \text{write}(e)$ jelöl majd:
 - $x : \text{outfile}(E)$ a szekvenciális outputfájltra hivatkozó változó
 - $e : E$ a fájl végéhez hozzáírt elem

$x := x \oplus \langle e \rangle$

Annak a felsoroló objektumnak a típusát, amely E-beli elemeket sorol fel, az $\text{enor}(E)$ jelöli.

Szekvenciális input fájl felsorolója

E-beli értékeket tartalmazó szekvenciális inputfájl elemeinek felsorolása

$\text{enor}(E)$				
E^*	$\text{first}()$	$\text{next}()$	$l := \text{end}()$ $l : \mathbb{L}$	$o := \text{current}()$ $o : E$
$x : \text{infile}(E)$ $e : E$ $st : \text{Status}$	$st, e, x : \text{read}$	$st, e, x : \text{read}$	$l := st = \text{abnorm}$	$o := e$

ez a felsorolás „elfogyasztja” a felsorolt fájlt

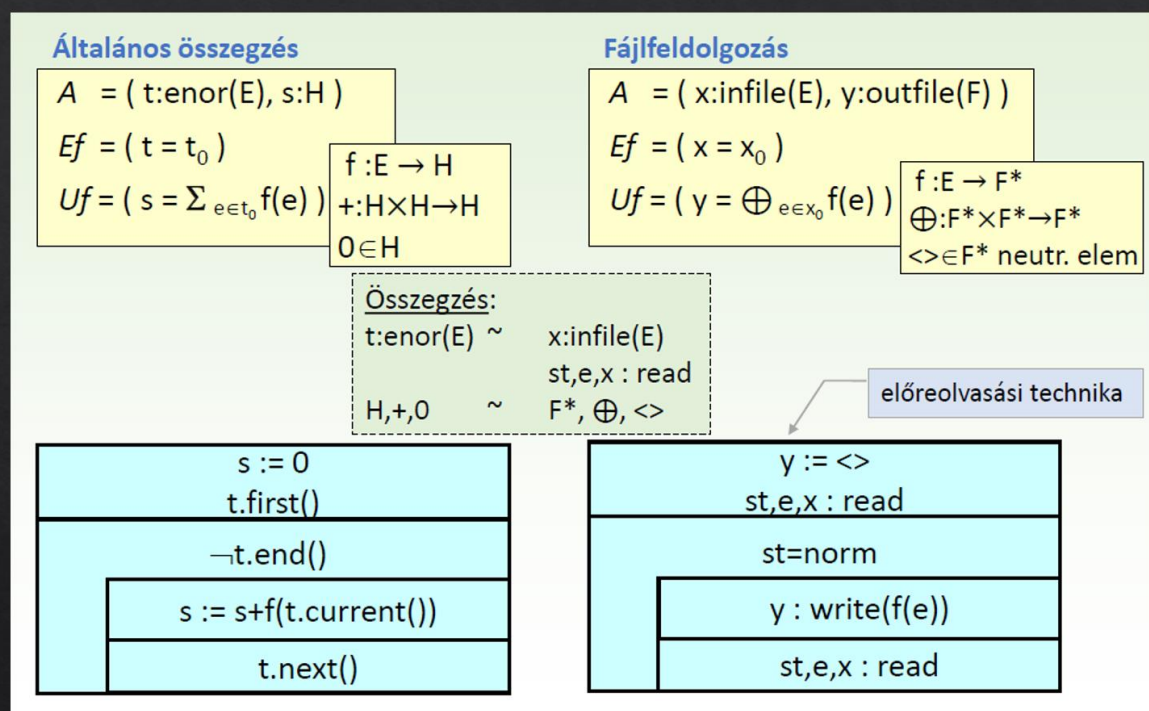
SeqInFileEnumerator	
$x : \text{infile}(E)$ $e : E$ $st : \text{Status}$	
$+ \text{first}()$	$: \text{void}$
$+ \text{next}()$	$: \text{void}$
$+ \text{end}()$	$: \text{bool} \{\text{query}\}$
$+ \text{current}()$	$: E \{\text{query}\}$

$st, e, x : \text{read}$
 $sf, e, x : \text{read}$
 $\text{return } st = \text{abnorm}$
 $\text{return } e$

előre olvasási stratégiára

$st, e, x : \text{read}$
$st = \text{norm}$
$\text{Feldolgoz}(e)$
$st, e, x : \text{read}$

Összegzés fájlkezeléshez



◆ Egy szekvenciális inputfájlban egyes kaktuszfajtákról ismerünk néhány adatot: név, őshaza, virágszín, méret.

◆ Számoljuk meg a piros virágú kaktuszokat!

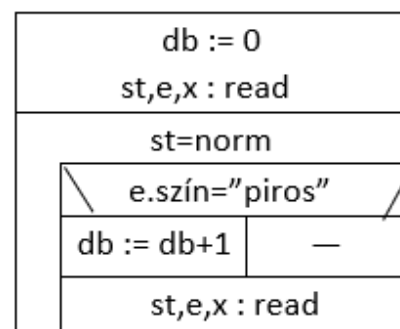
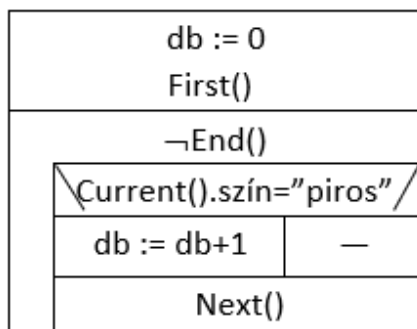
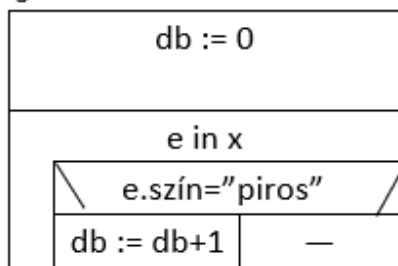
Specifikáció:

$A = (x: \text{infile}(\text{Kaktusz}), db: \mathbb{N})$
 $\text{Kaktusz} = \text{rec}(\text{név}: \mathbb{S}, \text{szín}: \mathbb{S}, \text{ős}: \mathbb{S}, \text{méret}: \mathbb{N})$
 $Ef = (x = x_0)$
 $Uf = (db = \sum_{\substack{e \in x_0 \\ e.\text{szín} = \text{"piros"}}} 1)$

Számlálás

$t: \text{enor}(E) \sim x: \text{infile}(\text{Kaktusz}) \ (st, e, x: \text{read})$
 $\text{felt}(e) \sim e.\text{szín} = \text{"piros"}$
 $c \sim db$

Algoritmus:



◆ Van-e piros virágú kaktusz?

Specifikáció:

$A = (x:infile(Kaktusz), l:\mathbb{L})$

$Kaktusz = rec(név:\mathbb{S}, szín:\mathbb{S}, ősz:\mathbb{S}, méret:\mathbb{N})$

$Ef = (x=x_0)$

$Uf = (l = \text{SEARCH}_{e \in x_0} e.szín = "piros")$

Linker

$t:enor(E) \sim x:infile(Kaktusz) \ (st,e,x:read)$

$felt(e) \sim e.szín = "piros"$

Algoritmus:

$l := \text{hamis}$	
$\neg l \wedge e \text{ in } x$	
$e.szín = "piros"$	
$l := \text{igaz}$	—

$l := \text{hamis}$ $\text{First}()$	
$\neg l \wedge \neg \text{End}()$	
$\text{Current}().szín = "piros"$	
$l := \text{igaz}$	$\text{Next}()$

$l := \text{hamis}$ $st, e, x : \text{read}$	
$\neg l \wedge st = \text{norm}$	
$e.szín = "piros"$	
$l := \text{igaz}$	$st, e, x : \text{read}$

◆ Válogassuk ki egy szekvenciális outputfájlba a piros virágú kaktuszos neveit!

Specifikáció:

$A = (x:infile(Kaktusz), y:outfile(\mathbb{S}))$

$Kaktusz = rec(név:\mathbb{S}, szín:\mathbb{S}, ősz:\mathbb{S}, méret:\mathbb{N})$

$Ef = (x=x_0)$

$Uf = (y = \bigoplus_{e \in x_0} \langle e.név \rangle)$

$e.szín = "piros"$

Összegzés (kiválogatás)

$H, +, 0 \sim \mathbb{S}^*, \bigoplus, \langle \rangle$

$t:enor(E) \sim x:infile(Kaktusz) \ (st,e,x:read)$

$f(e) \sim \langle e.név \rangle \text{ ha } e.szín = "piros"$

$s \sim y$

Algoritmus:

y:= <>		st:Status e:Kaktusz
st, e, x:read		
st = norm		
e.szín="piros"		
y:write(e.név)	—	
st, e, x:read		

- ◊ Válogassuk ki egy szekvenciális outputfájlba a piros virágú kaktuszok, egy másikba a mexikói őshazájú kaktuszok neveit!

◊ Specifikáció:

- ◊ $A = (x:\text{infile}(\text{Kaktusz}), y, z : \text{outfile}(\mathbb{S}))$
 $\text{Kaktusz} = \text{rec}(\text{név}:\mathbb{S}, \text{szín}:\mathbb{S}, \text{ős}:\mathbb{S}, \text{méret}:\mathbb{N})$
- ◊ $Ef = (x = x_0)$
- ◊ $Uf = (y = \bigoplus_{e \in x_0} \langle e.\text{név} \rangle \wedge z = \bigoplus_{\substack{e \in x_0 \\ e.\text{szín} = \text{"piros"}}} \langle e.\text{név} \rangle \wedge \bigoplus_{\substack{e \in x_0 \\ e.\text{ős} = \text{"Mexikó"}}} \langle e.\text{név} \rangle)$

- ◇ $t:enor(E) \sim x:infile(Kaktusz) \quad (st,e,x:read)$
- ◇ $f_1(e) \sim \langle e.név \rangle \text{ ha } e.szín = \text{"piros"}$
- ◇ $f_2(e) \sim \langle e.név \rangle \text{ ha } e.ős = \text{"Mexikó"}$
- ◇ $s \sim y, z$
- ◇ $H, +, 0 \sim \mathbb{S}^*, \oplus, \langle \rangle$

Algoritmus:

$y, z := \langle \rangle, \langle \rangle$ $st, e, x:read$	
$st = norm$	
$e.szín = \text{"piros"}$	
$y:write(e.név)$	—
$e.ős = \text{"Mexikó"}$	
$z:write(e.név)$	—
$st, e, x:read$	

st:Status
e:Kaktusz

◆ Melyik a legnagyobb piros virágú kaktusz neve?

Specifikáció:

$A = (x:\text{infile}(\text{Kaktusz}), l:\mathbb{L}, \text{név}:\mathbb{S}) \quad \text{Kaktusz} = \text{rec}(\text{név}:\mathbb{S}, \text{szín}:\mathbb{S}, \text{ős}:\mathbb{S}, \text{méret}:\mathbb{N})$

$Ef = (x=x_0)$

$Uf = (l, \text{max}, \text{elem}) = \mathbf{MAX}_{e \in x_0} e.\text{méret} \wedge l \rightarrow \text{név} = \text{elem}.\text{név} \quad \text{max}:\mathbb{Z}, \text{elem}:\text{Kaktusz}$
 $e.\text{szín} = \text{"piros"}$

Algoritmus:

l := hamis st, e, x : read				e:Kaktusz st:Status
st=norm				
e.szín≠"piros"	e.szín="piros" ∧ l		e.szín="piros" ∧ ¬l	
—	e.méret > max		l, max, név := igaz, e.méret, e.név	
	max, név := e.méret, e.név	—		
st, e, x : read				

Megjegyzés: A specifikáció szerint egy extra elágazás is kell a név változó értékének megadásához, de itt ezt beépült a feltételes maximum keresés ciklusába, hiszen elég a kaktusz (elem) helyett annak nevét (név) nyilvántartani.

◆ Igaz-e, hogy minden szám páros egy egész számok szekvenciális inputfájljában?

Specifikáció:

$A = (x:\text{infile}(\mathbb{Z}), l:\mathbb{L})$

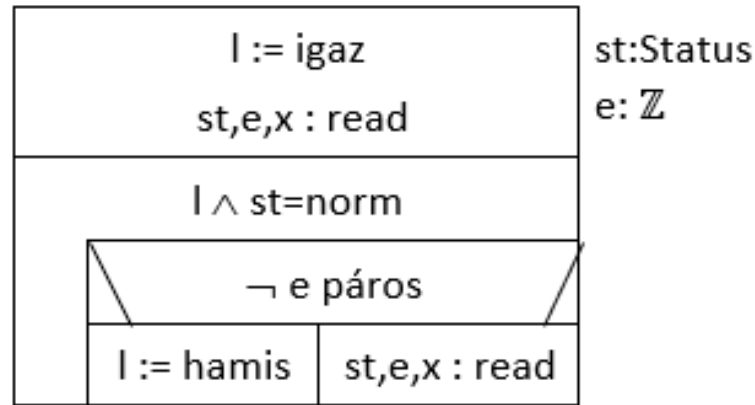
$Ef = (x=x_0)$

$Uf = (l = \mathbf{VSEARCH}_{e \in x_0} e \text{ páros})$

$t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z})$ felsorolója ($st, e, x:\text{read}$)

$\text{felt}(e) \sim e \text{ páros}$

Algoritmus:



◇ Ismerjük egy horgásznek egy horgászversenyen elért eredményét: a halfogásainak sorozatát. Egy fogás egy időpontból, a kifogott hal fajtájának nevéből, a hal súlyából (kg) és hosszából (cm) áll. Hány kilogrammnyi legalább 50 cm-es pontyot fogott összesen?

Specifikáció:

$A = (x: \text{Fogás}^*, s: \mathbb{R}) \quad \text{Fogás} = \text{rec}(\text{idő}: \mathbb{S}, \text{hal}: \mathbb{S}, \text{súly}: \mathbb{R}, \text{hossz}: \mathbb{R})$

$Ef = (x = x_0)$

$Uf = (x = x_0 \wedge s = \sum_{e \in x} e.\text{súly} \mid e.\text{hal} = \text{"ponty"} \wedge e.\text{hossz} \geq 0.5)$

Összegzés (feltételes összegzés)

$H, +, 0 \sim \mathbb{R}, +, 0$

$t: \text{enor}(E) \sim e \text{ in } x$

$f(e) \sim e.\text{súly} \text{ ha } e.\text{hal} = \text{"ponty"}$

$\wedge e.\text{hossz} \geq 0.5)$

Algoritmus:

s := 0.0		
e in x		
e.hal="ponty" ∧ e.hossz ≥ 0.5		
s := s + e.súly	—	

e:Fogás

- ◆ Egy szekvenciális fájlban egy bank számlatulajdonosait (azonosító, egyenleg) tartjuk nyilván. Adjuk meg annak azonosítóját, akinek nincs tartozása, de a legkisebb a számlaegyenlege.

Specifikáció:

$A = (x:infile(\text{Ügyfél}), l:\mathbb{L}, az:\mathbb{S}) \quad \text{Ügyfél} = rec(az:\mathbb{S}, egy:\mathbb{Z})$

$Ef = (x=x_0)$

$Uf = (l, min, elem) = \text{MIN}_{e \in x_0} e.egy \wedge l \rightarrow az = elem.az \quad min:\mathbb{Z}, elem:\text{Ügyfél}$
 $e.egy \geq 0$

Felt. min. ker.

$t:enor(E) \sim x:infile(\text{Ügyfél})$ felsorolója (st,e,x:read)

$f(e) \sim e.egy$

$felt(e) \sim e.egy \geq 0$

$H, > \sim \mathbb{Z}, <$

Algoritmus: Elég a legkisebb egyenlegű ügyfél helyett (elem) annak azonosítóját (az) nyilvántartani.

l := hamis st, e, x : read			e:Ügyfél st:Status
st=norm			
e.egy<0	e.egy≥0 ∧ l		l, min, az := igaz, e.egy, e.az
—	e.egy < min		
	min, az := e.egy, e.az	—	
st, e, x : read			