

Második beadandó feladat

Dokumentáció

Feladat

Egy szöveges állományban bekezdésekre tördelt szöveg található. Egy bekezdés egy vagy több nem üres sorból áll. A bekezdéseket üres sorok vagy az állomány eleje illetve vége határolja. Gondolatban sorszámozza meg a bekezdéseket és írja ki a konzolra azon bekezdések sorszámát, amelyeknek van olyan sora, amelyben egyáltalán nem szerepel az alma szó sem önmagában, sem valamilyen szóösszetételben! (A szövegben egyik szó sincs több sorra tördelve.) Csak azt a programot fogadjuk el, amelyik felsoroló objektum használatával oldja meg a feladatot. Készítse el a felsoroló osztályát! Tipp: Készítsen olyan felsorolót, amelyik a bekezdéseket, pontosabban azoknak számunkra fontos tulajdonságát sorolja fel!

Specifikáció

$A = (t: \text{Enor}(\text{Bekezdés}), \text{cout}: \text{OutFile}(\mathbb{N}))$

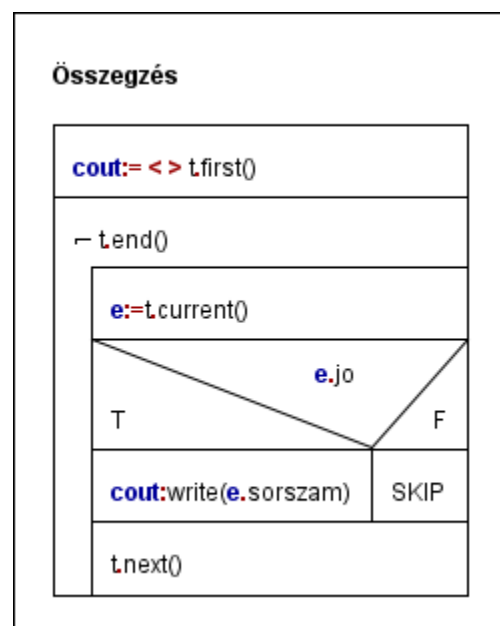
Bekezdés = **rec** (sorszám: \mathbb{N} , jo: L)

$E_f = (t = t')$

$U_f = (\text{cout} = \bigoplus_{\substack{e \in t' \\ e.jo}} \langle e.\text{sorszám} \rangle)$

Absztrakt program

<u>Összegzés</u>		
t	~	x sorainak felsorolója
+, 0	~	$\bigoplus, \langle \rangle$
f(e)	~	e.jo



Felsoroló

Enor(Bekezdés)	first(), next(), current(), end()
F:infile(String),sor:String,st:Státusz	first() ~ akt.sorszam:=0; st, sor, f : read; next()
Akt:Bekezdés	next() ~ ld. külön current() ~ akt
vége: L	end() ~ vége

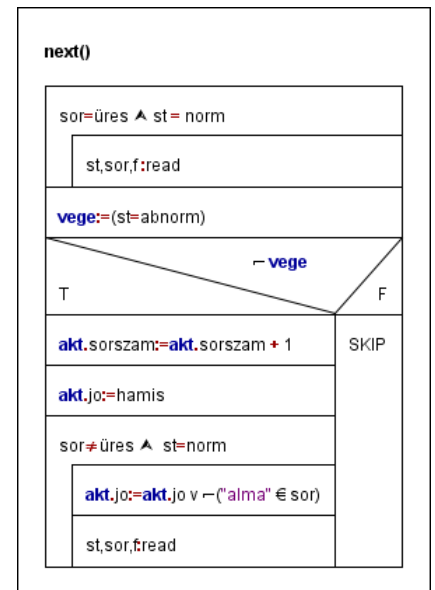
$A^{\text{Next}} = (f:\text{infile}(\text{String}), \text{sor}:\text{String}, \text{st}:\text{Státusz}, \text{vége}:\text{L}, \text{akt}:\text{Bekezdés})$

$Ef^{\text{Next}} = (f=f', \text{sor}=\text{sor}^1, \text{akt}=\text{akt}^1)$

$Uf^{\text{Next}} = (\text{st}^2, \text{sor}^2, f^2 = \text{select}(\text{st}=\text{abnorm} \vee \text{sor} \neq \text{üres})$
 $\text{sor} \in (\text{sor}^1, f^1)$

$\wedge \text{vége} = (\text{st}^2 = \text{abnorm}) \wedge (\neg \text{vége} \rightarrow \text{akt.sorsz} = \text{akt}^1.\text{sorsz} + 1 \wedge$

$\text{akt.jo}, \text{st}, \text{sor}, f = \bigvee_{\substack{\text{sor} \neq \text{üres} \\ \text{sor} \in (\text{sor}^2, f^2)}} \neg ("alma" \in \text{sor})$
 $)$

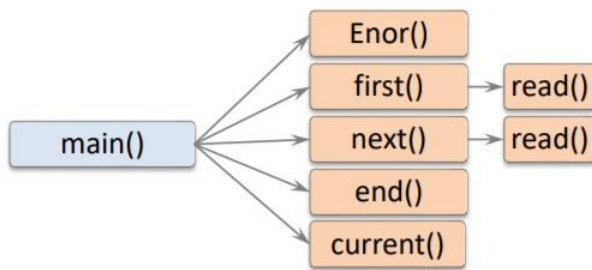


Implementáció

A program több állományból áll: *main.cpp*, *enor.h*, *enor.cpp*.

main.cpp	enor.h	enor.cpp
int main()	struct Bekezdes enum Status class Enor enor() void first() Bekezdes current() vooll end()	void read() void next()

Függvények kapcsolódási szerkezete:



Felsoroló osztálya

```
class Enor{
    private:
        std::ifstream f;
        std::string sor;
        Status st;
        Bekezdés akt;
        bool vege;
        void read();
    public:
        Enor(const std::string &str);
        void first() {read();next();}
        void next();
        Bekezdés current() const {return akt;}
        bool end() {return vege;}
};
```

A szöveges állomány soronkénti olvasását a *getline(f,sor)* utasítás végzi, amely be van ágyazva a *read()* metódusba, illetve beállítsa az olvasás *st* státuszát.

Egy sorban az “alma” szó a keresését a *string* típus *find()* függvénye végzi.

Tesztelési terv

A tesztelést 4 különböző inputfájlon végeztem el:

1. Az első inputfájl üres sorokat tartalmaz, így ellenőrzöm, hogy a program ez esetben nem számol egy bekezdést sem
2. A második inputfájl csak olyan bekezdéseket tartalmaz, melyben minden sor tartalmazza legalább egyszer az “alma” szót, így ellenőrzöm, hogy ez esetben a *jo* mindig hamis
3. A harmadik inputfájl vegyesen tartalmaz olyan bekezdéseket, melyeknek minden sorában van “alma” és nem minden sorában van “alma”. Ez esetben ellenőrzöm, hogy a program

kiválogatja az összes jó bekezdést, illetve, hogy tényleg annyi bekezdést olvas be ahány a fájlban van.

4. A negyedik inputfájl olyan bekezdéseket tartalmaz, melyek legalább egy sorában van alma, de van olyan soruk, amelyben nincs, így leellenőrzöm, hogy mindegyik jó.

Természetesen minden inputfájlban, ahol van az “alma” szó különböző helyeken van a sorban (elején, végén, közepén, szóösszetételben, első sorban, utolsó sorban, középső sorban.)