Dion Kajdomcaj
E9TZ67
kajdomcajdion@gmail.com
Group 1

2. assignment/5. task

11th February 2019

Task

The results of the National Angler's Championship is stored in a text file. Each line of the file contains the identifier of the participant and the championship (strings without whitespace), and the list of the caught fish, which are stored as pairs: (the kind of the fish, the size of the fish). The kind of the fish is a string without whitespace, its size is a natural number. The data in a line are separated by whitespace. The lines of the text file are sorted according to the name of the championship. You can assume that the text file is correct. An example for a line of the text file: James BigLakeChampionship Tuna 50 Salmon 20 Sardine 5 Tuna 100 Which championship had the most participants?

Give an angler who caught at least two catfishes on every competition he participated in.

Plan of the main program

A = (t : Enor(res),maxcount:Int,maxname:String , fname: String)

res = ***rec****(name:String, count:Int)*

Pre =(t = t’,maxcount=maxcount’,maxname=maxname’, fname=fname’)

Post=((maxname,maxcount)=MAX e.count) eϵt’

Algorithm: Maximum Search

```
t.first()
maxcount:=t.current().count
maxname:=t.current().name
t.next()
¬t.end()
    maxcount < t.current().count
        maxcount := t.current().count
        maxname:= t.current().name
    t.next()
```

Analogy:

t : enor ~ t:enor(res)
max : H ~ maxcount: Z

Elem : S ~ maxname: S

Enumerator

enor(res)	*first(), next(), current(), end()*

curr : *res*
end :L
Sx:Status
Dx:InData
F:infile
read():void ~
sx,dx,f:read
	first() ~*read(), next()*
	next() ~ *see below*
	current() ~ *curr*
	end() ~ *end*

Operation *next()* of *Enor(*res*)* has to solve the following problem:

Get the next Championship and its summation(count).For this purpose, the championships have to be enumerated with Input-results (on which competition which angler how many catfish he caught). It results in a *Input.*

A^{next} *= (x:infile(InData), end: L, curr:res,sx:Status,dx:InData)*

InData=rec(name:S,chname:S,vec:vector(Fish)), where Fish=rec(name:S,size:Z)

Pre^{next} = (*t = t', sx=sx', dx=dx'*) curr.name=dx'.name

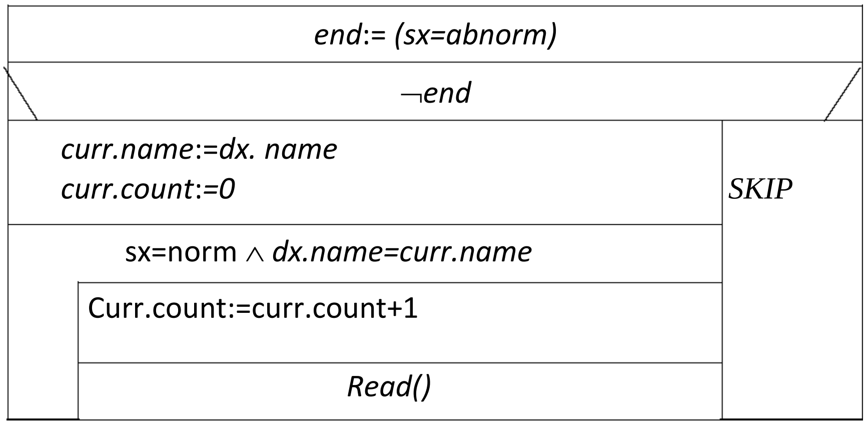
$Post^{next}$ = (*(curr.name,curr.count)=*$\sum_{curr.name=dx'.name} 1$)

Algorithm pattern:summation(counting)

Analogy

t:Enor(E) ~ x:infile(InData), sx,dx,x:read
cond: dx.name=curr.name
f(E) ~ 1
s ~ curr.count
H,+,0 ~ Z,+,0

next()



```
end:= (sx=abnorm)
¬end
    curr.name:=dx. name
    curr.count:=0
    sx=norm ∧ dx.name=curr.name
        Curr.count:=curr.count+1
        Read()
else
    SKIP
```

Testing plan

Two algorithmic patterns are used in the solution: maximum search and counting.

1.Test cases for maximum search when there is an empty file.
2.Test cases when the file doesn’t exist.
3.Test cases for maximum search when the maximum is at the beginning.
4.Test cases for maximum search when the maximum is in the middle.
5.Test cases for maximum search when the maximum is at the end.
6.Test cases when the file has only one championship.