|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *OPP* | *Documentation of Task* | | | | | *1.* | | | |  |  |
|  |  |  |
|  |  |  |  | | | | |  | | | |  |  |
|  |  | |  | | | | |  | | | | |  |
|  | **ARBAB ALI** | | **2. assignment/3. task** | | | | | 12th April 2020 | | | | |  |
|  |  | | | | | |  | |  |  |
|  |  | | |  |  |
|  |  | |  | | | | |  | | | |  |  |
|  |  |  |  | | | | |  | | | |  |  |

**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

List all the championships where tuna has been caught.

**Plan of the main program**

A = (t : Enor(Champ), l:L, ID : StringN )

champ =**rec**(champ:String, tuna:bool , counter:N)

Pre =( t = t’)

Post =( 𝑙,𝐼𝐷=𝑺𝑬𝑨𝑹𝑪𝑯 (𝑒.tuna))

𝑒∈𝑡′

|  |  |  |
| --- | --- | --- |
| l:=false ; t.first() | | |
| ~l ^ ~t.end() | | |
|  | l:= current().tuna | |
| t.current().tuna:=True | |
| ID:=t.current().ID |  |
| t.next() |  |

**Enumerator of Champ:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *enor(champ)* | | | *first(), next(), current(), end()* | | |
|  |  | | |  |  | |
|  | *tt* : *enor*(*contest*) | | | *first()* | *~ tt.first(); tt.next()* | |
|  | *act* : *champ* | | | *next()* | *~ see below* | |
|  | *end* : L | | | *current()* ~ *act* | | |
|  |  |  |  | *end()* | ~ *end* | |
|  |  |  |  |  |  |  |

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

Get the next contest or champ of which it is have to be decided whether in that festival tuna’s on all of his contest. For this purpose, the championship have to be enumerated with competition-results (on which competition which championship has been caught ). It results in a

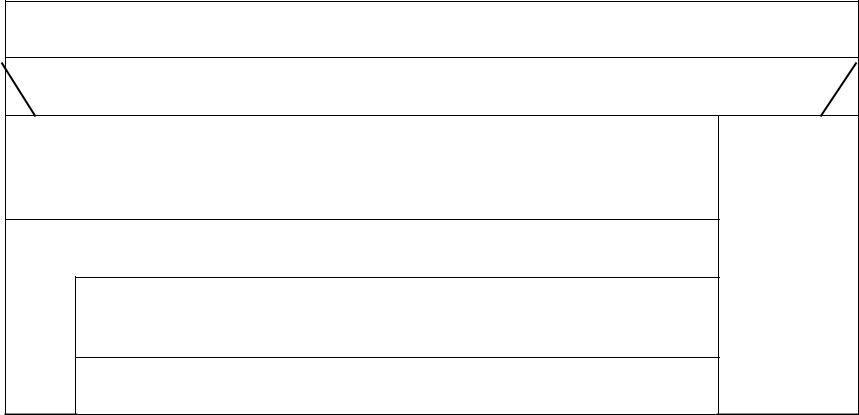
contest=**rec**(angler:String, contest:String, bool:L)) data structure. The first competition of the actual champ is already stored in tt.current(), neither tt.first(), nor tt.next() is needed. The enumeration lasts as long as the same champ’s competitions are read by operation tt.next().

A next= (tt:enor(contest), end:𝕃, act:Angler)

Pre next= (tt=tt1)

Post next= (end= tt.end()∧¬end→𝒂𝒄𝒕.**tuna** =⋀𝒆∈𝒕′(tt.current().tuna)))

***next()***

******

*end*:= *tt.end()*

*~end*

*act.champ*:=*tt.current().champ*

*act.tuna*:*=false* *SKIP*

*tt.end()* ^ *tt.current().contest=act.champ*

*act.tuna*:*= act.tuna OR tt.current().tuna*

*tt.next()*

***Enumerator of contest***

|  |  |  |
| --- | --- | --- |
| *enor(contest)* | *first(), next(), current(), end()* | |
|  |  |  |
| *f* : *infile*(*Line*) | *first()* | *~ see below* |
| *act* : *contest* | *next()* | *~ see below* |
| *end* : L | *current()* ~ *act* | |
|  | *end()* | ~ *end* |
|  |  |  |

Operations first() and next() of Enor(contest) are the same and they have to solve the following problem: Read the next line of the input file f. If there are no more lines, then variable end should be true. If there are more lines, then get the name of the contest and the ID of the contest and find the word „tuna”.

**A** next = (f: infile(Line), end: , act:contest) Line = seq(Word)

**Pre** next = ( f = f’)

**Post** next = ( sf, df, f = read(f’) end=(sf=abnorm) ^

~end → act.angler = “first word of df” ^

act.contest = “second word of df” ^

act.tuna = if word ’tuna’ in df” exist or not )

In the implementation, the two classes of the two above enumerator objects (t and tt) are placed into separate compilation units.

***Testing plan***

A)

1. empty file
2. only 1 championship
3. more championship
4. first line is contest is with tuna
5. only last line is contest is with tuna
6. no tuna in all file
7. more than one contest with tuna in file

B)

1. no contest in whole file
2. one and only contest with tuna in file
3. first contest with tuna is not the first contest in file
4. first contest with tuna, tuna is not caught by both angler
5. second contest with tuna is caught by both angler