Dokumentáció

Koller Dávid Dániel

IDJJQ2

OEP 2. beadandó

Feladat

Az Országos Horgász Bajnokság mindegyik versenyén feljegyezték a versenyzők

eredményeit, és egy szöveges állományban rögzítették az adatokat. Ennek minden sora

tartalmazza a versenyző horgász és a verseny azonosítóját (szóközök nélküli sztring),

majd halfajta-súly párok (szóközök nélküli sztring és valósszám párok) formájában az

adott versenyen elért fogásokat. Az adatok szóközökkel vagy tabulátorjelekkel vannak egy

soron belül elválasztva. A szöveges állomány sorait a horgászok szerint rendezték.

Feltehetjük, hogy a szöveges állomány helyesen van kitöltve. Példa az állomány egy

sorára:

JANIBÁ Kiliti0512 Ponty 4.5 Ponty 5.3 Harcsa 9.6

(1) Mennyi volt a legkisebb kifogott ponty súlya? Ki és melyik versenyen fogta? (Nem

biztos, hogy fogtak pontyot a horgászok!)

(2) Igaz-e, hogy minden horgász legalább két versenyen fogott pontyot?

Részprogram #1 megoldása

FőProgram terve:

A = (t : enor(Verseny), I: L, suly:R, horgasz:string, Versenynev:string)

Verseny = rec(horgasz:String, versenynev:String, halak:sorozat(Hal))

Hal = rec(nev:String, suly:R)

Ef = (t = t’ ∧ |t|>0 )

Uf = (l, suly, horgasz, versenynev) = MINe∈t'𝑓(𝑒)

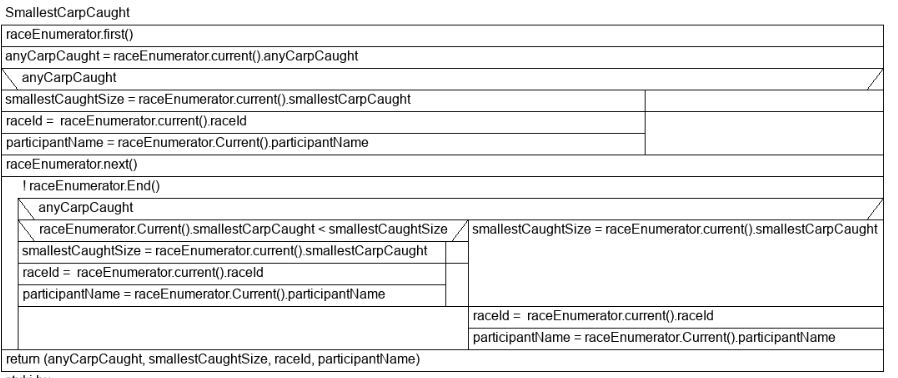
felt(e)

felt : Verseny => L

felt(x) = ∃k∈[x.halak]:k.nev = ’ponty’

f : Verseny => R

∃k∈[x.halak]:k.suly = f(x) ∧ ¬∃jeR: ∃gE[x.halak] ∧ j = g.suly ∧ j<(f(x)



(l => anyCarpCaught, suly => smallestCaughtSize, versenynev => raceId, horgasz => participantName, t=> raceEnumerator)

Versenyek felsorolója

enor(verseny) first(), next(), current(), end()

f: inFile(Sor) first() ~ next()

cur: Verseny next() ~ lásd később

end:L current() ~ cur

end() ~ end

Az enor(Verseny) first() és next() műveletei megegyeznek, és az alábbi feladatot kell megoldaniuk: olvassuk be a szöveges állomány (az f szekvenciális inputfájl) soron következő sorát. Ha ilyen nincs, akkor az end változó értéke legyen igaz. Ha van, akkor ezt a sort sztringek sorozatának tekintve kiszedhetjük belőle a horgász és a verseny azonosítóját, majd az ezeket követő részben kiválasztjuk a „ponty” nevű, legkisebb tömegű halat. Ha találtunk legalább egyet, akkor ezt megjelöljk egy külön erre definiált igaz/hamis kapcsolóval.

A next = (f: infile(Sor), end:𝕃, cur:Verseny) Sor = String\*

Efnext = ( f = f’ )

Ufnext = ( sf, df, f = read(f’) ∧ end=(sf=abnorm)

∧ ¬end → cur.horgász = df1 ∧ cur.verseny = df2

∧ cur.vanPonty = ∃k∈[ cur.halak]:k.nev = ’ponty’

∧∃ k∈[ cur.halak]:k.suly = cur.suly) ∧ ¬∃jeR: ∃gEcur.halak] ∧ j = g.suly ∧ j<cur.suly

Az enor(Horgász) next() műveletnek az alábbi feladatot kell megoldania:

Állítsuk elő a soron következő horgászt, akiről el kell dönteni, hogy vajon minden versenyén fogott-e legalább két pontyot. Ehhez fel tudjuk használni az első részfeladat megoldásához készített verseny felsorolót (enor(Contest)), mely az input fájl egy sorát értelmezi, és megadja, hogy az adott versenyen az adott horgász fogott-e pontyot. A next() művelet végrehajtásakor a Contest felsoroló (ha még nem ért a felsorolás végére) már tartalmazza azt a horgászt, és annak első versenyét, amelyet a next() feldolgoz majd, így, a tt.current() értékét már ismerjük, ehhez nem kell sem a tt.first(), sem a tt.next(). A felsorolás csak addig tart, amíg ugyanazon horgász versenyeit „olvassuk” a tt.next() művelettel.

