

Gregorics Tibor

2. beadandó/0.feladat

2013. február 11.

NEPTUN

gt@inf.elte.hu

0.csoport

Feladat

Egy szöveges állományban egy kurzusra jelentkezett hallgatók félévközben elért eredményeit tároljuk. Egy sorban szóközzel elválasztva szerepel a hallgató neve (sztring), Neptun kódja (sztring), továbbá a félévközi zárthelyiken (kettő ilyen van) elért pontszámai (0 és 30 közötti egész számok), ezt követően a vizsgán szerzett pontszáma (0 és 60 közötti egész szám), a nyilatkozata arról, hogy elfogadja-e a megajánlott vizsgajegyet (ennek feltétele, hogy a két zárthelyin legalább 40 pontot érjen el), és nyilatkozata arról, hogy a vizsga értékelésbe beszámítsanak-e a félévközben írt zárthelyik. Számítsuk ki a hallgatók vizsgajegyét! Ha valaki elfogadja a megajánlott jegyet, akkor ez a zárthelyik összpontszáma tizedének egész része. Ekkor nem vizsgázhat, tehát a vizsgapontszáma 0. Ezt majd ellenőrizni kell! Ha valaki vizsgázik, de beszámíttatja a zárthelyik eredményét, akkor ki kell számolni a zárthelyik összpontszámának felét, amelyhez hozzáadjuk a vizsgapontszámot, és az így kapott érték tizedének egész része lesz a vizsgajegy. Ekkor a vizsgapontszám nem lehet 30-nál több, amit majd ellenőrizni kell! Egyébként a vizsgajegy a vizsgapontszám tizedének egész része.

Például az input fájl adatai:

Gipsz Jakab	ABCDEF	17	23	0	I	I
Kabor Bors Orsolya	BCDEFG	10	10	24	N	I
Zabos Zsolt Zsombor	CDEFGH	10	30	55	N	N

Specifikáció

$$A = (x: \text{Student}^*, y: \text{Result}^*)$$

$$\text{Student} = \text{rec}(\text{name: String}, \text{test1: } \mathbb{N}, \text{test2: } \mathbb{N}, \text{exam: } \mathbb{N}, \text{accept: } \mathbb{L}, \text{include: } \mathbb{L})$$

$$\text{Result} = \text{rec}(\text{name: String}, \text{mark: } \mathbb{N})$$

$$E_f = (x = x')$$

$$U_f = (y = \bigoplus_{e \in x'} \langle (e.\text{name}, \lfloor f(e) / 10 \rfloor) \rangle) \quad f: \text{Hallgató} \rightarrow \text{Eredmény}$$

$$f(e)$$

$$= \left\{ \begin{array}{ll} \frac{e.\text{test1} + e.\text{test2}}{2} + e.\text{exam} & \text{ha } e.\text{accept} \wedge e.\text{test1} + e.\text{test2} \geq 40 \\ \frac{e.\text{test1} + e.\text{test2}}{2} + e.\text{exam} & \text{ha } (\neg e.\text{accept} \vee e.\text{test1} + e.\text{test2} < 40) \wedge e.\text{include} \\ e.\text{exam} & \text{ha } (\neg e.\text{accept} \vee e.\text{test1} + e.\text{test2} < 40) \wedge \neg e.\text{include} \end{array} \right\}$$

Hibajelzést kell adni, ha $e.\text{accept}$ és $e.\text{exam} > 0$, illetve ha $e.\text{include}$ és $e.\text{exam} > 30$.

Absztrakt program

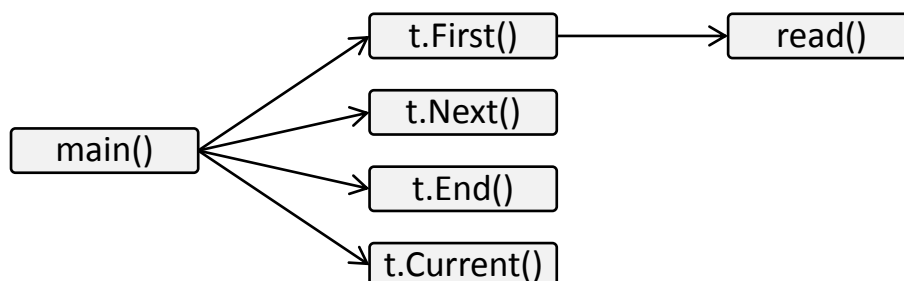
összegzés		
t	\sim	x sorainak felsorolója
$+, 0$	\sim	$\oplus, < >$
$f(e)$	\sim	$\langle (e.name, \lfloor f(e) / 10 \rfloor) \rangle$

t.First()		
$\neg t.End()$		
$e.accept \wedge e.exam > 0$	$e.include \wedge e.exam > 30$	<i>else</i>
<i>Hiba</i>	<i>Hiba</i>	$y.write(t.Current().name, \lfloor f(t.Current) / 10 \rfloor)$
t.Next()		

Implementáció

Az x szekvenciális inputfájl sorait bejáró felsorolót a SeqInFile osztállyal, a Student típust egy struktúrával valósítjuk meg, a Result típust viszont egyáltalán nem kell.

A SeqInFile osztály First() és Next() művelete egy $st, e, x: read$, amely a szöveges állomány soron következő sora alapján állítja elő az e Student típusú rekordot. Ha nincs soron következő sor, akkor az st státuszt $abnorm$ -ra állítja. Ez a státusz az End(), a beolvasott rekord a Current() segítségével kérdezhető le.



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

main.cpp	seqinfile.h - seqinfile.cpp
main()	class SeqInFile First() Next() Current() End()

Tesztelés

Az összefűzés (összegzés) programozási tétel tesztesei:

intervallum hossza szerint

Üres fájl

Egyetlen sorból álló fájl

Két sorból álló fájl

Sok sorból álló fájl

intervallum eleje és vége szerint

Két sorból álló fájl

tételre jellemző esetek

<elég a fenti néhány eset>

Az f függvény tesztesei:

Hibás adatok ($e.\text{accept} \wedge e.\text{exam} > 0$) egy sorban

Hibás adatok ($e.\text{include} \wedge e.\text{exam} > 30$) egy sorban

Aki elfogadta a megajánlott jegyet ($e.\text{accept} \wedge e.\text{test1} + e.\text{test2} \geq 40$)

Aki el akarta fogadni a megajánlott jegyet ($e.\text{accept} \wedge e.\text{test1} + e.\text{test2} < 40$)

Aki nem fogadta el a megajánlott jegyet ($\neg e.\text{accept} \wedge e.\text{test1} + e.\text{test2} \geq 40$)

Aki nem fogadta el a megajánlott jegyet ($\neg e.\text{accept} \wedge e.\text{test1} + e.\text{test2} < 40$)

Aki beszámította a zh-t a vizsgába

($\neg e.\text{accept} \vee e.\text{test1} + e.\text{test2} < 40$) $\wedge e.\text{include}$

és aki nem

($\neg e.\text{accept} \vee e.\text{test1} + e.\text{test2} < 40$) $\wedge \neg e.\text{include}$

First() ill. Next() művelet tesztesei:

A név egy, kettő, három, ill. négy tagú.