

Egy munkafolyamat teljesítését ellenőrzik. Óránként nézik az darabszámot és rögzítik az adatokat. A feladat önkéntes, így bárki bármikor abbahagyhatja, ekkor az eddigi eredményeit veszik figyelembe.

A mért eredményeket az `adatok.csv` pontosvesszővel tagolt állományba tárolják.

Kis János	39	4	43	27	1	14	20	
Nagy Béla	13	31	98	44	79	38		
Gergely Tibor	22	37	85	87	14	22	36	20
Vas Aladár	21	11	63	8				
Dadics Béla	1	11	11	73	6	67	23	34
Esze János	27							
Valaki Jenő	70	35	28	31	84	66	66	50
Máris Kata	74	18						

FELADATOK

Adatok beolvasása	<code>adatokBe("adatok.csv");</code>
Táblázat kiírása (tesztelni a beolvasást, ...)	<code>tblKi();</code>
Név, délelőtti, délutáni összegek	<code>szummaDe_Du();</code>

f

<pre>import java.io.*; import java.util.*; static RandomAccessFile f; static ArrayList<Szaki> szakik = new ArrayList<>();</pre>		A lista konstruktora
eljárások, függvények		
<pre>eljárás adatokBe(String f_neve) olvasás f = new RandomAccessFile(f_neve, "r"); sor = f.readLine(); <-----> ciklus amíg (sor != null) szakik.add(new Szaki(sor)); sor = f.readLine(); ciklus vége f.bezárása(); olvasási hiba kezelése eljárás vége</pre>	az elől tesztelő ciklushoz, belépés Szaki osztály konstruktorának hívása	
<pre>eljárás tblKi() N = szakik.mérete(); ciklus i=0-tól N-ig egyesével Ki: szakik.get(i).toString(); ciklus vége eljárás vége</pre>	Szaki osztály metódusának hívása	
<pre>eljárás szummaDe_Du() N = szakik.size(); ciklus i=0-tól N-ig egyesével Ki: szakik.get(i).getNev(), szakik.get(i).SzummaDe(), szakik.get(i).SzummaDu() ciklus vége eljárás vége</pre>	Szaki osztály metódusainak hívása	

Szaki	
private String nev;	név tárolása
private ArrayList<Integer> db new ArrayList<>();	napi teljesítmény listája
konstruktor	
<pre>public Szaki(String sor){ String[] tmp = sor.split(";"); this.nev = tmp[0]; this.fillPontok(tmp); }</pre>	<p>az adatok.csv egy sora jön a konstrukcióba szeletelés a segéd (tmp) tömbbe a tmp[0] elem név feltöltjük az adatokat a listába</p>
metódusok	
<pre>+ függvény String getNev() return nev; függvény vége</pre>	a név mező elérését biztosítja a külső program számára (főprogram)
<pre>- eljárás fillPontok(String[] tmp) N = tmp.hossza; ciklus i=1-től N-ig egyesével db.add(egészszám(tmp[i])); ciklus vége eljárás vége</pre>	<p>a teljes tmp[]-t kapja meg hány bejegyzése van</p> <p>hozzáfűzés az add() metódussal</p>
<pre>+ függvény int szummaDe() { s = 0; N = db.mérete(); ha (N > 3) akkor: N = 4; ha vége ciklus i=0-tól N-ig egyesével s = s + db.get(i); ciklus vége return s; függvény vége</pre>	<p>a délelőtti összeg [1;4] óra</p> <p>csak a délelőtti adatok kelljenek</p> <p>a listából a get()-el lehet adatot lekérni</p>
<pre>+ függvény int szummaDu() { s = 0; N = db.mérete(); ha (N > 4) akkor: ciklus i=4-től N-ig egyesével s = s + db.get(i); ciklus vége ha vége return s; függvény vége</pre>	<p>a délutáni összeg [5;8] óra</p> <p>csak a délutáni adatok kelljenek</p> <p>a listából a get()-el lehet adatot lekérni</p>