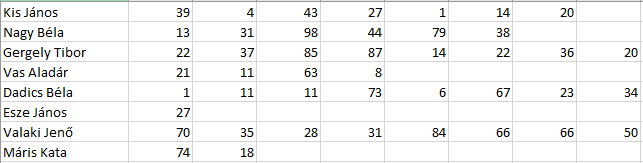
Egy munkafolyamat teljesítését ellenőrzik. Óránként nézik az darabszámot és rögzítik az adatokat. A feladat önkéntes, így bárki bármikor abbahagyhatja, ekkor az eddigi eredményeit veszik figyelembe.

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



Feladatok

|  |  |
| --- | --- |
| Adatok beolvasása | **adatokBe**("adatok.csv"); |
| Táblázat kiírása (tesztelni a beolvasást, …) | **tblKi**(); |
| Név, délelőtti, délutáni összegek | **szummaDe\_Du**(); |

f

|  |  |  |
| --- | --- | --- |
| import java.io.\*;  import java.util.\*; | |  |
| static RandomAccessFile *f*;  static ArrayList<Szaki> *szakik* = new ArrayList<>(); | | A lista konstruktora |
| eljárások, függvények | | |
| 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 | az elől tesztelő ciklushoz, belépés  **Szaki** osztály konstruktorának hívása | |
| eljárás **tblKi**()  N = *szakik*.mérete();  ciklus i=0-tól N-ig egyesével  Ki: *szakik*.get(i).toString();  ciklus vége  eljárás vége | **Szaki** osztály metódusának hívása | |
| 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 | **Szaki** osztály metódusainak hívása | |

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