

Egy triatlon versenyen a versenyzőknek a verseny folyamán egymás után kell először úszniuk, kerékpározniuk majd futniuk. Az győz, aki a legrövidebb idő alatt fejezi be a versenyt.

Az egyes versenyzők adatai és időeredményei a `triathlon.txt` fájlban találhatók.

Az **első sorban a versenyzők száma**, a következő sorokban a versenyzők adatai a következők szerint szerepelnek:

Név

Úszás idő

Kerékpár idő

Futás idő

Például:

```
Gipsz Jakab
1345
2312
7988
```

Az elért időeredményeket másodpercekben tároljuk.

1. Olvassa be a `triathlon.txt` fájlból az adatokat!
2. Írja ki az összesített időeredmények alapján az első három versenyző nevét és eredményét!
3. Írja ki a képernyőre az első helyezett nevét és azt, hogy mekkora volt az átlagsebessége (km/h-ban) az úszásban, kerékpározásban és futásban, ha a távok a következők voltak:
Úszás: 1,5 km,
Kerékpározás: 40 km,
Futás: 10 km!
4. Konvertálja át a versenyzők végső időeredményeit `óó:pp:ss` formátumra . Mindegyik értéket két számjeggyel jelölje!
A versenyzők nevét és az átkonvertált, összesített időeredményét (rangsorban) írja a név mellett szerepeljen az időeredmény, például: *Gipsz Jakab 03:14:05!*

FŐPROGRAM (TRIATLON OSZTÁLY)

ADATSZERKEZET

```
static ArrayList<Versenyző> versenyzők = new ArrayList<>(); // lista, ha új versenyző jön felvehető...
```

METÓDUSOK

adatokBe ("triathlon.txt");

az adatokat a String[4] tömbbe olvassa be a program és adja át a **Versenyző(tmp)** konstruktor hívással a **versenyzők<>** listának

rendezés ()

minimum vagy buborékos, ..., összehidők szerint Z-A a **Versenyző** osztály **getSzummaPontok()** metódusával

csere(int i, int j)

felcseréli a **versenyzők** lista két elemét $i \leftrightarrow j$

elsoHárom ();

a rendezett adatok első 3 elemét írja ki (ciklus)

sebességek();

a rendezett lista 1. helyezettjének eredményeit kiírja

pl.: **double uszas=(double)3.6*1500/((versenyzok.get(0).getPontok()[0]));**

konvertáltidők ();

a **Versenyző** osztály konvertált adatait kéri le és írja ki a **getNév()** és a **getKonvertál()** metódusával

VERSENYZŐ OSZTÁLY

ADATSZERKEZET (adattagok)

- név;
- **pontok** int[3]; // tömbben tároljuk az adatokat mindenkié ugyan annyi

KONSTRUKTOR

```
+ Versenyző(String [] tmp);
    név = tmp[0];
    fillSportágak(tmp);
```

METÓDUSOK

- **fillSportágak(String [] tmp) ()**
feltölti a pontokat egészszámként a **pontok** tömbbe

+ **getSzummaPontok()**

összegzi a **pontok** tömböt
pl.: **return pontok[0]+pontok[1]+pontok[2];**

+ **getNév()**

visszaadja a Név mező értékét

+ **getPontok()**

visszaadja a **pontok** mező értékét

+ **getKonvertál()**

oo:pp:ss formátumban visszaadja a **getSzummaPontok** értékét

FŐPROGRAM (TRIATLON OSZTÁLY)

eljárás adatokBe(f_neve)

```
f: fájl megnyitása olvasásra;  
N = egészszám (f.egysor_olvasása); // az 1. sorban van a méret  
String[] tmp = new String[4];  
ciklus j=0-tól N-ig egyesével  
    ciklus i=0-tól 4-ig egyesével  
        sor = f.readLine();  
        tmp[i] = sor; // a tmp [4] feltöltése  
    ciklus vége  
versenyzők.add(new Versenyző(tmp)); // adatok küldése a konstruktornak  
ciklus vége  
f.bezárása();
```

olvasási hiba kezelése

eljárás vége

eljárás rendezés()

```
N = versenyzők.mérete();  
ciklus i=0-tól N-ig egyesével  
    minIndex = i; // legyen ő a minimum indexe  
    ciklus j = i + 1-től N-ig egyesével  
        ha (versenyzők.get(j).getSzummaPontok() < versenyzők.get(minIndex).getSzummaPontok())  
            minIndex = j; // új lett a minimum indexe  
        ha vége  
    ciklus vége  
csere(i, minIndex);  
ciklus vége
```

eljárás vége

VERSENYZŐ OSZTÁLY

```
+ függvény String getKonvertal() {  
    x = getSzummaPontok();  
    óra = x / 3600;  
    perc = (x - óra * 3600) / 60;  
    mperc = x - (óra * 3600) - (perc * 60);  
    s=String.format("%02d:%02d:%02d", óra, perc, mperc);  
    return s;  
függvény vége
```