

A személyi szám 11 jegyű.

Az ország állampolgárainak van egyedi azonosítójuk. Ez a személyazonosító jel.
Ez a következőképpen néz ki.

A személyazonosító jel 11 jegyű.

Az első jegy a személy nemét jelöli, az alábbi táblázat alapján.

1999.12.31-ig született		1999.12.31 után született	
férfi	nő	férfi	nő
1	2	3	4

A 2 - 7 számjegyek a születési év két utolsó jegyét, a születési hónapot és napot tartalmazza. A 8 - 10 számjegyek az azonos napon születettek születési sorszáma. A 11. jegy az első tíz jegyből generált ellenőrző szám.

Feladatok:

- Olvassa be a *szemszam.txt* fájl adatait - tabulátorokkal tagolt rekordok.

Rekord szerkezet: **név, személyszám első 10 számjegye.**

Pl.:

Karam Ella	4080107199
Para Zita	2490709322
Seft Elek	1340110416

- Készítsen a fájl *nőkAZ.txt* néven a nők névsoráról

- Írja ki a képernyőre a minta szerint, ki mikor született.

- +■ Ki a legidősebb?

Személy	AppSzemélyszám
<ul style="list-style-type: none">- név: String- személyszám: String- születésidátum: String // <i>éééé.hh.nn</i>- neme: char // <i>f/n</i>	<pre>személyek <Személy> :lista nők<String>: lista</pre>
<pre>Személy(sor: String) tmp: String[] this.név this.személyszám setSzületésidátum(személyszám) setNeme (személyszám)</pre>	<pre>adatokBe(path: String) személyek.add(new Személy(sor)) nőiNévsor() rendezés() adatokKi() rendezés() csere(i, j) csere(i: int, j: int) adatokKi(path: String) név_szüldátum() main() adatokBe(path) nőiNévsor() név_szüldátum()</pre>
<ul style="list-style-type: none">- setSzületésidátum(személyszám): String- setNeme(személyszám): char+ getNév(): String+ getNeme(): char+ getSzületésidátum(): String+ getKor(): String // <i>a legidősebbhez</i>	

Személy

- **setSzületésidátum**(személyiszám)
elkészíti a születésidátumot *éééé.hh.nn* formtumban, és átadja a **this.születésidátum** mezőnek
 - **setNeme**(személyiszám)
meghatározza az illető nemét és átadja a **this.neme** mezőnek
 - + **getNév**()
visszaadja a **név** mező értékét
 - + **getNeme**()
visszaadja a **neme** mező értékét
 - + **getSzületésidátum**()
visszaadja a **születésidátum** mező értékét
 - + **getKor**()
visszaadja a **(születésidátum + személyiszám)** utolsó 3 mezőjének értékét
-

AppSzemélyiszám

- adatokBe**(path: String)
1. beolvassa az adatokat a path-ból
 2. feltölti a *személyek* listát
 3. átadja a **Személy** konstruktorának az objektumokhoz szükséges adatokat
- nőiNévsor**(path: String)
1. kiválogatja a *nők* listába a nőket
 2. névsorba rendezi *nők* a listát
 3. kiírja a path fájlba a rendezett *nők* listát
- név_szüldátum**()
- kiírja a monitorra a *személyek* listából a neveket és a formázott születésidátumokat
- legidősebb()***// a legidősebb +feladat*
- kiírja a monitorra a *személyek* listából a legidősebb nevét és a formázott születésidátumát

Személy

```
függvény setNeme(személyszám: String): char
|
|   neme = 'n' //nő
|   char c = személyszám első karaktere
|   ha (c = '1' vagy c = '3')
|       akkor neme = 'f' //férfi
|   ha vége
|       return neme
függvény vége

függvény setSzületésidátum (személyszám: String): String
|
|   születésidátum= "19" //1900, ha 3 vagy 4 akkor 2000
|   char c = személyszám első karaktere
|   ha (c = '3' vagy c = '4')
|       akkor születésidátum= "20"
|   ha vége
|       év = születésidátum + személyszám 2.-3. karaktere + "."
|       hó = személyszám 4.-5. karaktere + "."
|       nap = személyszám 6.-7. karaktere
|       születésidátum = év + hó + nap
|       return születésidátum //pl: 1917.08.06
függvény vége
```

AppSzemélyszám

```
eljárás adatokBe(path: String)
|
|   f = fájl megnyitása olvasásra (path)
|   sor = f.egysor beolvasása
|   ciklus amíg (sor nem üres)
|       személyek.add(new Személy(sor)) // konstruktor hívása
|       sor = f.egysor beolvasása
|   ciklus vége
|   f.zárása()
eljárás vége

eljárás nőiNévsor(path: String)
|
|   int N = személyek.size()
|   ciklus i=1-től N-ig egyesével
|       ha (személyek.get(i).getNeme() == 'n' {
|           nők.add(személyek.get(i).getNév()) // felfűzés a <nők> listára
|       }
|   ha vége
|   ciklus vége
|   rendezés() // <nők> névsora
|   adatokKi(f_neve) // <nők> névsora fájlba
eljárás vége

eljárás rendezés()
|
|   int N = nők.size()
|   tetszőleges rendezése a <nők> listának a csere eljárással
|   nők.get(j).compareTo(nők.get(minIndex)) < 0 // névsor feltétele
eljárás vége

eljárás adatokKi(path: String)
|
|   f = fájl megnyitása írásra (path)
|   int N = nők.size()
|   ciklus i=1-től N-ig egyesével
|       f.writeBytes(nők.get(i) + "\n") // egy sor + soremelés kiírása
|   ciklus vége
|   f.zárása()
eljárás vége
```