

Arrays

Az `java.util.Arrays` osztály tömbökkel kapcsolatos műveleteket tartalmaz:

- tömb lemásolása (`copyOf`)
- (bináris) keresés rendezett tömbben (`binarySearch`)
- tömb feltöltése megadott elemmel (`fill`)
- tömb elemeinek összehasonlítása (`equals`)
- stringgé alakítás (`toString`)
- és ami nekünk most fontos: rendezés (`sort`)

Arrays.sort

Amennyiben primitív típusból (minden, ami nem osztály és nem tömb: `int`, `short`, `double`, `float`, ...) álló tömböt kap paraméterül, akkor a természetes, nagyság szerinti sorrendet veszi alapul. Ha objektumokról van szó, akkor pedig

- az objektumnak meg kell valósítania a `Comparable` interface-t
- a sorrendezés az objektumok `compareTo` metódusa szerint történik

Például:

```
Diak[] diakok = new Diak[2];
diakok[0] = new Diak("Kiss Klotild", 1994, 4.5);
diakok[1] = new Diak("Nagy Elemér", 1995, 3.4);
Arrays.sort(diakok);
```

Bináris keresés

A bináris keresés rendezett tömbök esetén használható, azt rövidíti le jelentősen.

A szótárban sem úgy keresünk, hogy kinyitjuk az első oldalon, és addig nézzük a szavakat, míg meg nem találjuk a keresettet.

A szótárban úgy keresünk, hogy kinyitjuk középen, elolvassuk a középső oldalon található egyik szót, és ha az, amit keresünk, ennél kisebb, akkor a szótár első felében van a szó, ha ennél nagyobb, akkor a hátsó felében van a szó. Ezután ugyanezt a módszert alkalmazzuk a félszótárra is addig, míg meg nincs a szó.

Emberként azért ennél okosabbak is tudunk lenni, és nem középen nyitjuk ki a szótárt, ha az asztalt keressük, amiről tudjuk, hogy valószínűleg az elején van. (A példa annyiból nem tökéletes, hogy a kinyitott oldalon is számos szó található).

A bináris kereséssel minden alkalommal felezzük a szótár méretét. Ami azt jelenti, hogy egy 1.000.000 elemet tartalmazó szótárban (tömbben) 20 lépésből megvan a szó, szemben a lineáris keresés átlagos 500.000 lépésével. Jó, nem?

A vájt fülűek kedvéért: a lépésszám az elemek számának logaritmusával arányos.

```
Diak[] diakok = new Diak[1000000];
diakok[0] = ...;
...
Arrays.sort(diakok);
int index = Arrays.binarySearch(diakok, klotild);
```

A fenti példában `klotild` egy `Diak` osztályú objektum, amit a `diakok` tömbben keresünk.

A `compareTo` metóduson kívül legyen megfelelően elkészítve az `equals()` metódus is! (Ha a `compareTo()` 0-t ad vissza, az `equals()` adjon vissza `true`-t!)

