Java

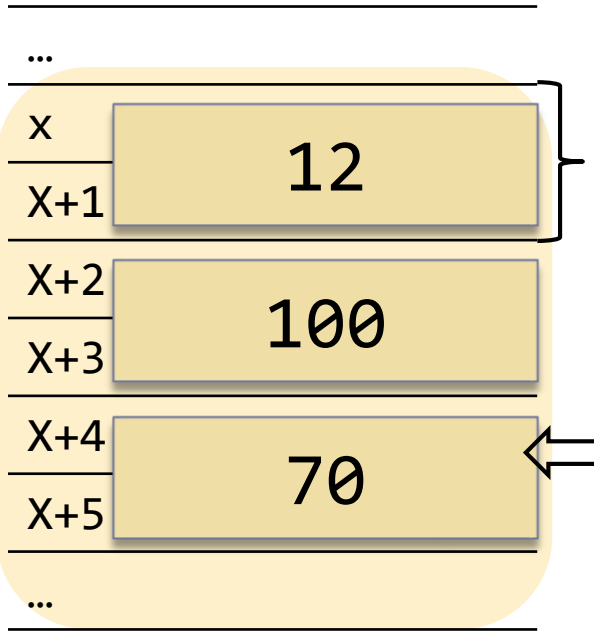
Tömbök, listák

A tömb adatszerkezet jellemzői

- Azonos típusú értékeket tartalmazhat
- Az egyes értékeket sorszámukkal, (indexükkel) azonosíthatjuk
- A sorszámozás 0-val kezdődik
- Létrehozás (deklaráció)

```
short[]  t   =   new     short[3]
```

- Az elemeket szigorúan egybefüggő területre helyezi el a program (*oprendszer*).
- **Bármelyik** elem helyét egyszerű matematikával azonos idő alatt képes kiszámolni
- Nem módosítható a mérete
- Csak egyszerű adattípust kezel



short helyfoglalás 2 byte

összesen: 6 byte

pl. a 3. elem helyének meghatározása:

felső bájt: x(=0) + (3-1)*2

A lista adatszerkezet jellemzői

Olyan adatszerkezet, amely

- nagyszámú, azonos típusú elem tárolására alkalmas
- bármelyik eleme lekérdezhető *lista.get(i)*
- elemeit kitörölhetjük *lista.remove(n)*
- újat vehetünk fel *lista.add()*
- nem feltétlenül ismert előre a sorozat elemszáma, de lekérdezhető *lista.size()*
- *Objektumokat (rekordot tárol)*
- …

A lista adatszerkezet fontosabb metódusai

`lista.add("Veg Eta")`	hozzáadja a rekordot a lista végéhez
`lista.add(3,"Am Erika")`	beszúrja a rekordot a 3. elem elé (*rendezések*)
`lista.set(3,"Bac Ilus")`	átírja a 3. elemet Bac Ilus-ra
`lista.get(1)`	a lista 1. elmének értéke
`lista.remove(1)`	törli az 1. elemet
`lista.size()`	a lista elmeinek száma

A listaelem szerkezete

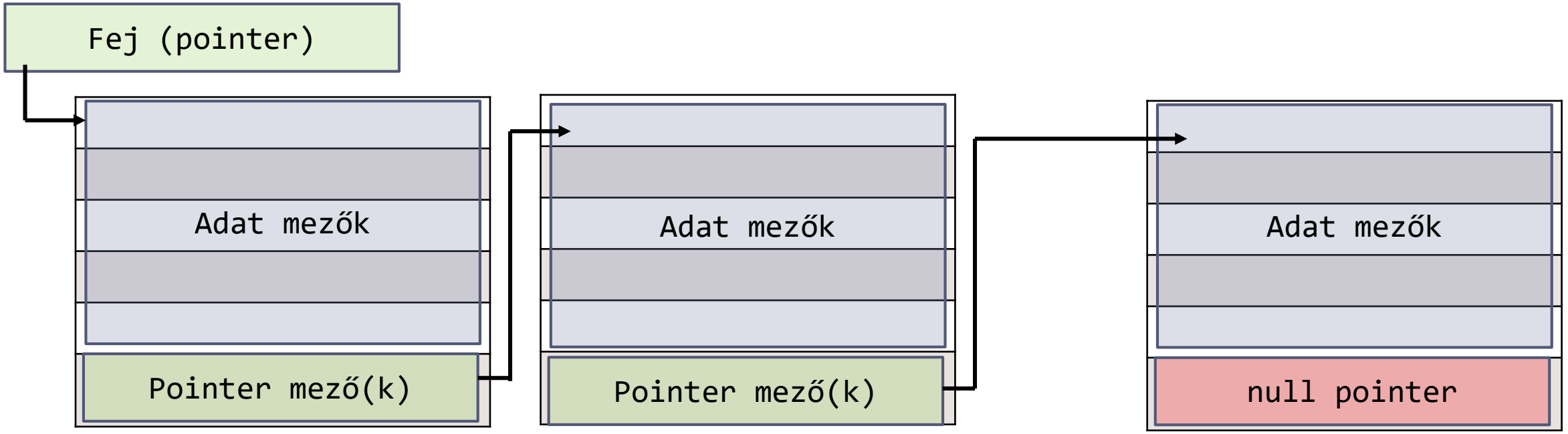
A listaelem feladata kettős:

- helyet kell biztosítania a sokaság egy eleme számára
- biztosítania kell a következő elem elérését, hiszen a memória tetszőleges területén lehet.

A listaelem funkcionálisan két részből áll:

1. a tárolandó sorozat egy eleme
2. mutató a következő elem eléréséhez.

A listaelem, láncolt lista szerkezete



Olyan homogén, dinamikus, szekvenciális elérésű adatszerkezet, amelynek minden eleme azt az információt **is** tárolja, hogy a következő eleme hol van a számítógép memóriájában.

A lista első elemének eléréséhez csupán egy mutató is elegendő. Ennek ismerete a teljes lista ismeretét jelenti, mert így hozzáférhetünk az elemben tárolt minden adathoz, ahhoz is hogy hol található a következő elem.