

# Adatszerkezetek

## 05. Dinamikusan láncolt listák

Vekov Géza

2023. március 22.



Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

Láncolt lista AAT

# Egyszeresen láncolt lista - absztrakt verzió

Adatszerkezetek

Vekov Géza

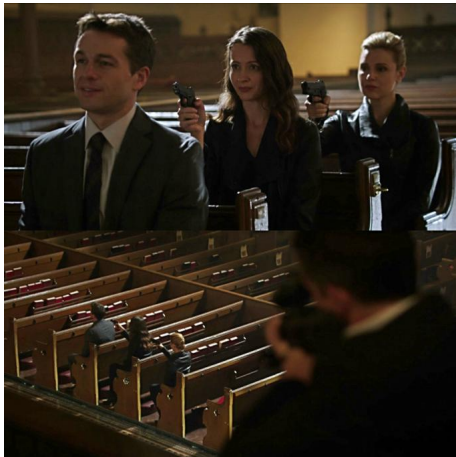
Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT



1

---

<sup>1</sup><https://files.catbox.moe/zty4sc.jpg>

# Láncolt lista AAT

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Leírás

- *Homogén*: azonos elemtípus
- *Lineáris*: az elemek a láncolási sorrendben követik egymást
  - A tömböknél a linearitást az indexek egymásutánisága határozta meg, itt a sorba rendezettséget a mutatók valósítják meg.
- Lehet *rendezett* vagy *rendezetlen*
- A *láncolástól függően* lehet: egyszeresen láncolt, kétszeresen láncolt, ciklikus, stb.

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

Egyszeresen láncolt lista AAT: listaelem

# Egyszeresen láncolt lista AAT

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Elemtípus

```
ADAT adatmezo;  
MUTATO kov;
```

```
typedef struct ELEM {  
    ADAT adatmezo;  
    struct ELEM * kov;  
} ELEM;
```

## Leírás

- *adatmező*: a tárolt adatok típusa
- *köv*: az aktuális listaelem után *következő* listaelem címét tartalmazó mutató
- bármely listaelem lehet a lista feje
- amennyiben valamely *köv* mező értéke elvesztlődik, a lista oda kapcsolódó része "elvesztlődik", elveszítjük a hozzáférést.

# Egyszeresen láncolt lista AAT: elem-műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Elem létrehozása: **createNode(adat)**

- **előfeltétel:** -
- **utófeltétel:** létrejön egy listaelem, melynek az *adat*-mezője a megadott adat és a *köv*-mezője *NULL*. Téríti a létrehozott elem címét.

## Elem felszabadítása: **destroyNode(E)**

- **előfeltétel:** -
- **utófeltétel:** felszabadítja a megadott elem tárhelyét

# Egyszeresen láncolt lista AAT

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

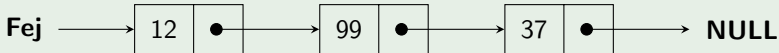
Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Példa: háromelemű egyszerűen láncolt lista

- a 12-es adatot tartalmazó elem a **listafej**
- a listához a listafejen keresztül férünk hozzá
- minden listaelem tárol egy mutatót, mely a **következő listaelem címét** tartalmazza
- a 37-es adatot tartalmazó elem következője *null*, ez a lista vége





# Egyszeresen láncolt lista AAT: szerkezet

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Szerkezet

- Listafej - elemtípusnak megfelelő típusú

## Megjegyzés

- A listafejen keresztül érjük el a lista valamennyi elemét.

# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Létrehozás: **create()**

- **előfeltétel:** -
- **utófeltétel:** létrejön egy üres lista a Fej értéke *NULL*

## Felszabadítás: **destroy(Fej)**

- **előfeltétel:** -
- **utófeltétel:** felszabadítunk minden elemet, melyek a lista részét képezik

# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás: **insert(adat)**

Beszúrási lehetőségek:

- üres listába
- lista elejére
- adott  $p$  elem elé
- adott  $p$  elem után
  
- **előfeltétel:** -
- **utófeltétel:** egy új elem bekerül a listába a megadott helyre.

# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

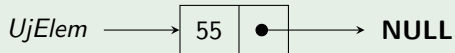
## Beszúrás üres listába

Műveletek:

1 **UjElem**  $\leftarrow$  **nodeCreate(adat)**

2 **Fej**  $\leftarrow$  **UjElem**

**Fej**  $\longrightarrow$  **NULL**



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

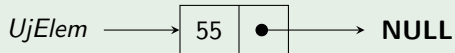
## Beszúrás üres listába

Műveletek:

1 **UjElem**  $\leftarrow$  **nodeCreate(adat)**

2 **Fej**  $\leftarrow$  **UjElem**

**Fej**  $\longrightarrow$  **NULL**



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

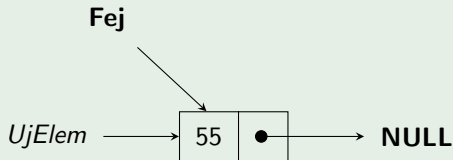
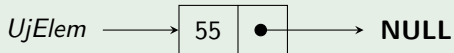
## Beszúrás üres listába

Műveletek:

1  $UjElem \leftarrow \text{nodeCreate(adat)}$

2  $Fej \leftarrow UjElem$

$Fej \longrightarrow \text{NULL}$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

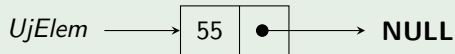
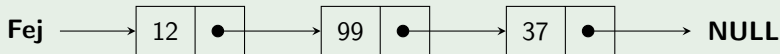
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás lista elejére

Műveletek:

- 1 **UjElem**  $\leftarrow$  **nodeCreate(adat)**
- 2 **UjElem.köv**  $\leftarrow$  **Fej**
- 3 **Fej**  $\leftarrow$  **UjElem**



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

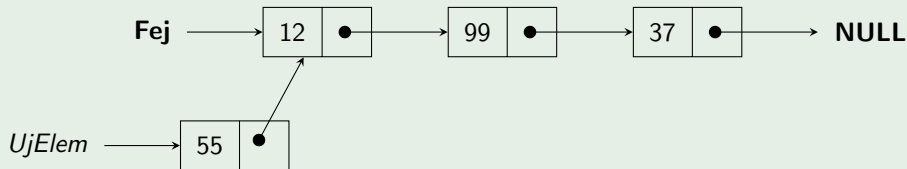
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás lista elejére

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $UjElem.köv \leftarrow Fej$
- 3  $Fej \leftarrow UjElem$





# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

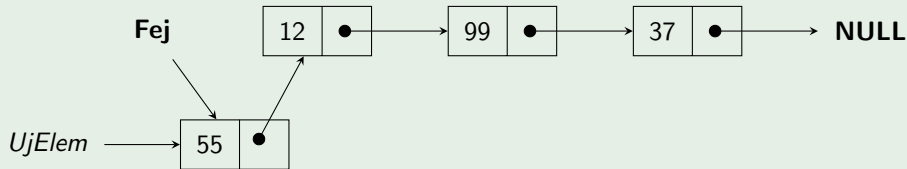
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás lista elejére

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $UjElem.köv \leftarrow Fej$
- 3  $Fej \leftarrow UjElem$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

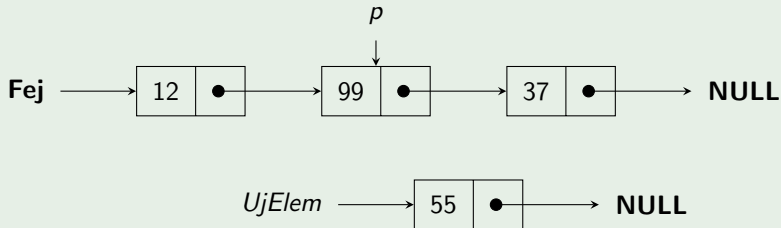
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás $p$ után

Műveletek:

- 1 **UjElem**  $\leftarrow$  **nodeCreate(adat)**
- 2 **UjElem.köv**  $\leftarrow$  **p.köv**
- 3 **p.köv**  $\leftarrow$  **UjElem**



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

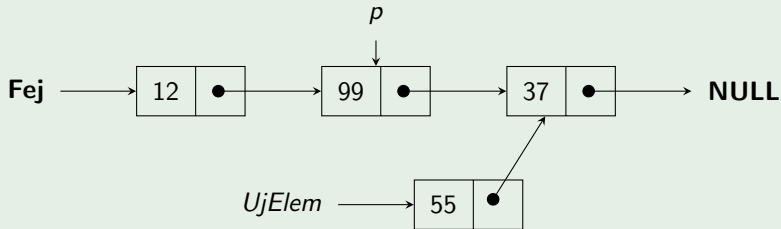
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás $p$ után

Műveletek:

- 1  $UjElem \leftarrow \text{nodeCreate(adat)}$
- 2  $UjElem.köv \leftarrow p.köv$
- 3  $p.köv \leftarrow UjElem$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

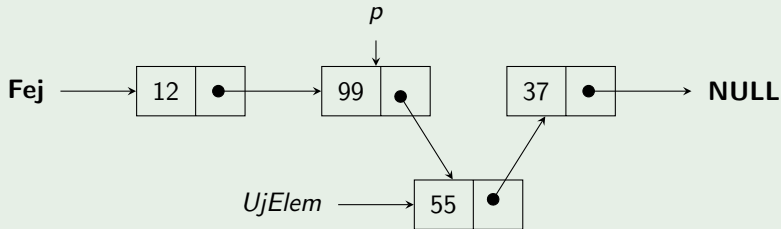
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás $p$ után

Műveletek:

- 1  $UjElem \leftarrow \text{nodeCreate(adat)}$
- 2  $UjElem.köv \leftarrow p.köv$
- 3  $p.köv \leftarrow UjElem$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

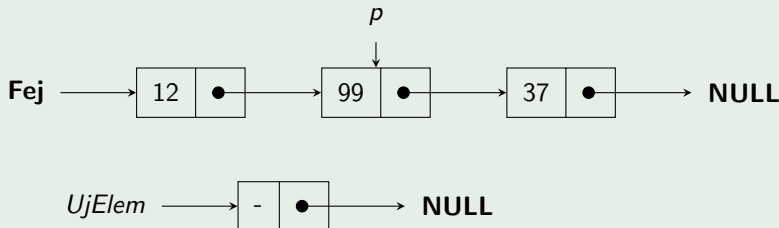
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás $p$ elé

Műveletek:

- 1 **UjElem**  $\leftarrow$  **nodeCreate(-)**
- 2 **UjElem.mezők**  $\leftarrow$  **p.mezők**
- 3 **p.köv**  $\leftarrow$  **UjElem**
- 4 **p.adat**  $\leftarrow$  **adat**



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

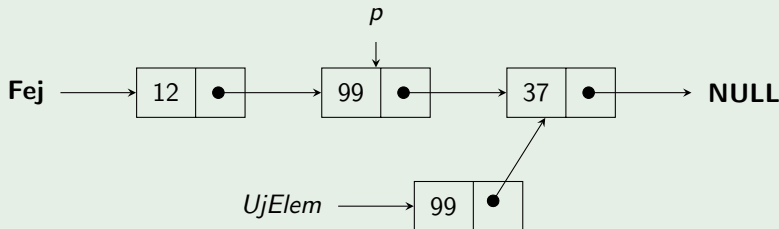
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás $p$ elé

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(-)$
- 2  $UjElem.mezők \leftarrow p.mezők$
- 3  $p.köv \leftarrow UjElem$
- 4  $p.adat \leftarrow adat$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

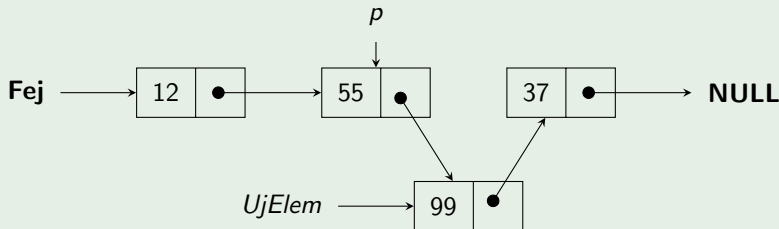
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás $p$ elé

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(-)$
- 2  $UjElem.mezők \leftarrow p.mezők$
- 3  $p.köv \leftarrow UjElem$
- 4  $p.adat \leftarrow adat$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Törlés: **delete(p)**

Törlési lehetőségek:

- első elem
  - utolsó elem
  - nem első és nem utolsó
- 
- **előfeltétel:** Az adott  $p$  cím egy létező listaelem címe.
  - **utófeltétel:** Az adott  $p$  című mező törlődik a listából.



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

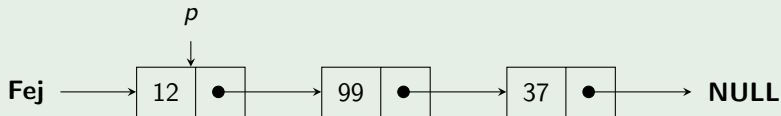
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - első elem törlése

Műveletek:

- 1 Fej  $\leftarrow$  Fej.köv
- 2 destroyNode( $p$ )



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

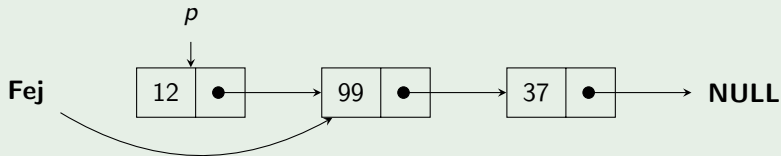
Rendezett lista  
AAT

## Adott $p$ elem törlése - első elem törlése

Műveletek:

1  $\text{Fej} \leftarrow \text{Fej.köv}$

2  $\text{destroyNode}(p)$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

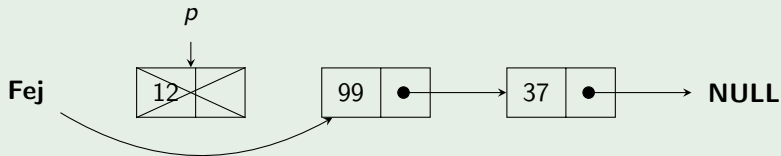
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - első elem törlése

Műveletek:

- 1  $\text{Fej} \leftarrow \text{Fej.köv}$
- 2 **destroyNode(p)**



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

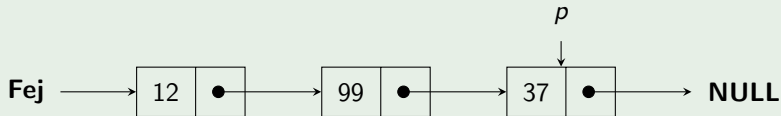
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - utolsó elem törlése

Műveletek:

- 1 Meg kell keresni azt a  $q$  elemet, ami a  $p$  előtt van.
- 2  $q.köv \leftarrow \text{NULL}$
- 3 `destroyNode(p)`



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

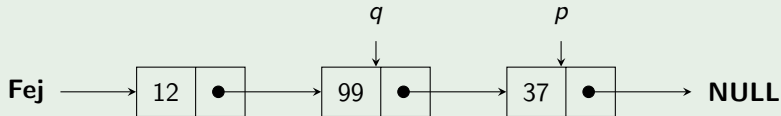
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - utolsó elem törlése

Műveletek:

- 1 Meg kell keresni azt a  $q$  elemet, ami a  $p$  előtt van.
- 2  $q.köv \leftarrow \text{NULL}$
- 3 `destroyNode(p)`



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

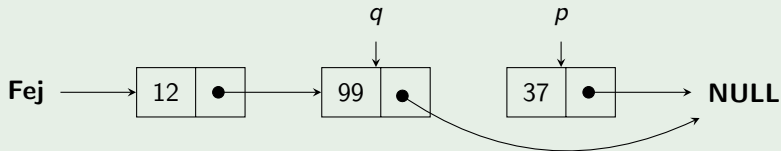
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - utolsó elem törlése

Műveletek:

- 1 Meg kell keresni azt a  $q$  elemet, ami a  $p$  előtt van.
- 2  $q.köv \leftarrow \text{NULL}$
- 3  $\text{destroyNode}(p)$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

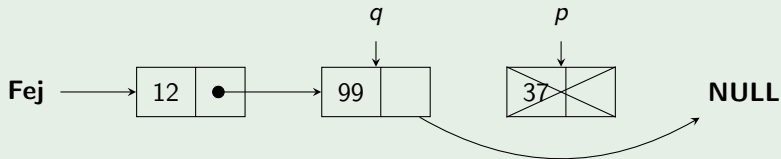
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - utolsó elem törlése

Műveletek:

- 1 Meg kell keresni azt a  $q$  elemet, ami a  $p$  előtt van.
- 2  $q.köv \leftarrow \text{NULL}$
- 3 **destroyNode( $p$ )**



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

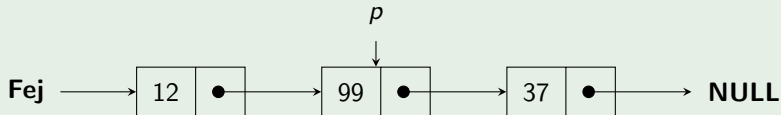
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - "belső" elem törlése

Műveletek:

- 1  $q \leftarrow p.\text{köv}$
- 2  $p.\text{mezők} \leftarrow q.\text{mezők}$
- 3  $\text{destroyNode}(q)$





# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

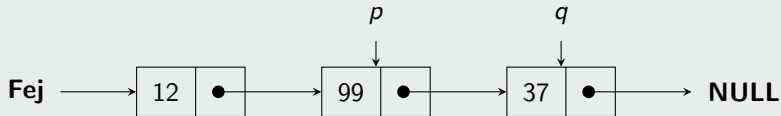
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - "belső" elem törlése

Műveletek:

- 1  $q \leftarrow p.köv$
- 2  $p.mezők \leftarrow q.mezők$
- 3  $destroyNode(q)$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

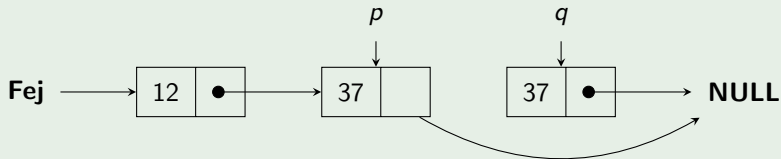
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - "belső" elem törlése

Műveletek:

- 1  $q \leftarrow p.\text{köv}$
- 2  $p.\text{mezők} \leftarrow q.\text{mezők}$
- 3  $\text{destroyNode}(q)$



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

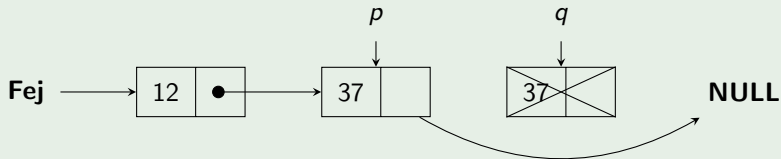
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Adott $p$ elem törlése - "belső" elem törlése

Műveletek:

- 1  $q \leftarrow p.\text{köv}$
- 2  $p.\text{mezők} \leftarrow q.\text{mezők}$
- 3 **destroyNode( $q$ )**



# Egyszeresen láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Bejárás

- **előfeltétel:** -
- **utófeltétel:** Valamennyi elemet érinti a lista első elemével kezdődően. Figyelem, az utolsó érvényes elem az utolsó elem, nem pedig a NULL.

# Egyszeresen láncolt lista AAT: másképp

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Egyszerűsítsünk

- Használjunk két strázsaelemet, amik mindig a lista részei.
- Egyik strázsaelem a lista első eleme előtt legyen, a másik az utolsó után.

## Mi történik?

- Egyszerűbb lesz valamely művelet?
- Lehetne még egyszerűbben?

# Egyszeresen láncolt lista AAT: strázsával

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Szerkezet

```
typedef struct LISTA
    ELEM * strazsa1, strazsa2;
    LISTA;
```

# Egyszeresen láncolt lista AAT: strázsával

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Tulajdonságok

- egyik strázsa mindig az első elem előtt van
- második strázsa mindig az utolsó elem után van
- a listának mindig van legalább két eleme
- amikor a lista fejére (első elemére) hivatkozunk, akkor gyakorlatilag a  $strazsa_1 \rightarrow kov$  elemet kell használni
- üres lista ellenőrzése:  $strazsa_1 \rightarrow kov == strazsa_2$
- meg kell valósítani a *fej* hivatkozást

# Egyszeresen láncolt lista AAT: strázsával

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Létrehozás

- létre kell hozni a két strázsát is

## Beszúrás

- mindig belső elemet szúrunk be
- az első elem beszúrása belső elem beszúrásának felel meg
- a lista "feje" soha nem változik meg, az első strázsazsaelem mindig jó elemre fog mutatni



# Egyszeresen láncolt lista AAT: strázsával

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Törlés

- mindig belső elemet törlünk
- legrosszabb esetben a *strazsa<sub>2</sub>* előtti elemről lehet szó. Ekkor a *strazsa<sub>2</sub>* felülírja az utolsó elemet a listából, és megváltozik a címe

## Megjegyzés

Egy strázsával is meg lehet oldani az egyszeresen láncolt listát, ekkor ez lesz az első és az utolsó elem is egyben.

Gyakorlatilag *körkörösén láncolt listát* hozunk létre.

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Duplán láncolt lista AAT

# Duplán láncolt lista AAT: szerkezet

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Szerkezet

- Listafej, listavég - elemtípusnak megfelelő típusú

## Elemtípus

ADAT adatmezo;  
MUTATO elozo, kov;

```
typedef struct ELEM {  
    ADAT adatmezo;  
    struct ELEM * elozo, kov;  
} ELEM;
```

## Leírás

- *adatmező*: a tárolt adatok típusa
- *kov*: az aktuális listaelem után *következő* listaelem címét tartalmazó mutató
- *elozo*: az aktuális listaelem *előtti* listaelem címét tartalmazó mutató

# Duplán láncolt lista AAT: elem-műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Elem létrehozása: `createNode(adat)`

- **előfeltétel:** -
- **utófeltétel:** létrejön egy listaelem, melynek az *adat*-mezője a megadott adat és a *köv*- és *elozo*-mezője *NULL*. Téríti a létrehozott elem címét.

## Elem felszabadítása: `destroyNode(E)`

- **előfeltétel:** -
- **utófeltétel:** felszabadítja a megadott elem tárhelyét

# Duplán láncolt lista AAT

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

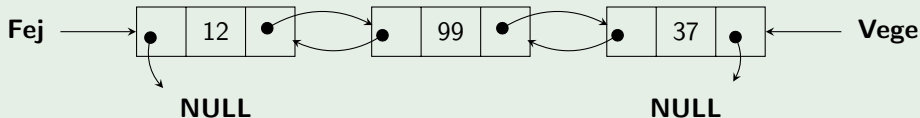
Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Példa: háromelemű duplán láncolt lista

- a 12-es adatot tartalmazó elem a **listafej**, előzője *null*
- a listához a listafejen keresztül férünk hozzá
- minden listaelem tárol egy mutatót, mely **az előző és a következő listaelem címét** tartalmazza
- a 37-es adatot tartalmazó elem következője *null*, ez a lista **vége**



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Létrehozás: **create()**

- **előfeltétel:** -
- **utófeltétel:** létrejön egy üres lista a *Fej* és a *Vége* értéke *NULL*

## Felszabadítás: **destroy(Fej)**

- **előfeltétel:** -
- **utófeltétel:** felszabadítunk minden elemet, melyek a lista részét képezik

# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás: **insert(adat)**

Beszúrási lehetőségek:

- üres listába
- lista elejére/végére
- adott  $p$  elem elé
- adott  $p$  elem után
  
- **előfeltétel:** -
- **utófeltétel:** egy új elem bekerül a listába a megadott helyre. Szükség esetén frissülnek a *Fej* és *Vége* mutatók

# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás üres listába

Műveletek:

1 **UjElem**  $\leftarrow$  **nodeCreate(adat)**

2 **Fej**  $\leftarrow$  **UjElem**, **Vege**  $\leftarrow$  **UjElem**

**Fej**  $\longrightarrow$  **NULL**  $\longleftarrow$  **Vege**





# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

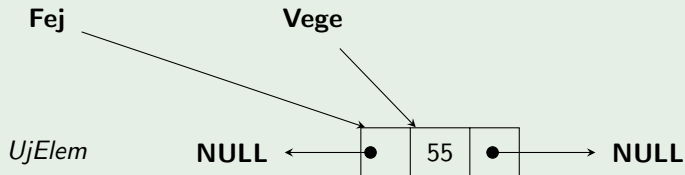
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás üres listába

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $Fej \leftarrow UjElem; Vege \leftarrow UjElem$



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

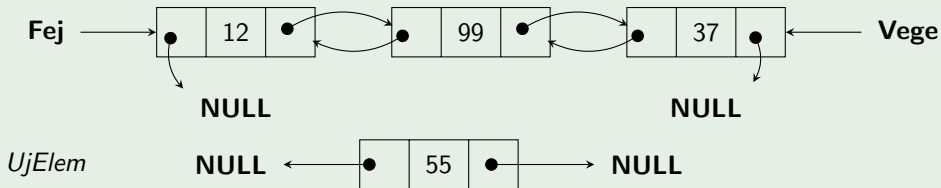
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás lista elejére

Műveletek:

- 1 **UjElem**  $\leftarrow$  **nodeCreate(adat)**
- 2 **UjElem.köv**  $\leftarrow$  **Fej**
- 3 **Fej.elozo**  $\leftarrow$  **Ujelem**
- 4 **Fej**  $\leftarrow$  **UjElem**



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

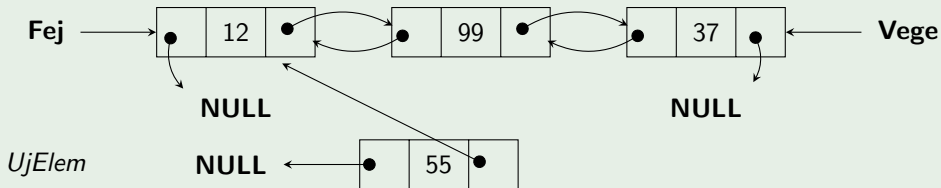
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás lista elejére

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $UjElem.köv \leftarrow Fej$
- 3  $Fej.elozo \leftarrow Ujelem$
- 4  $Fej \leftarrow UjElem$



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

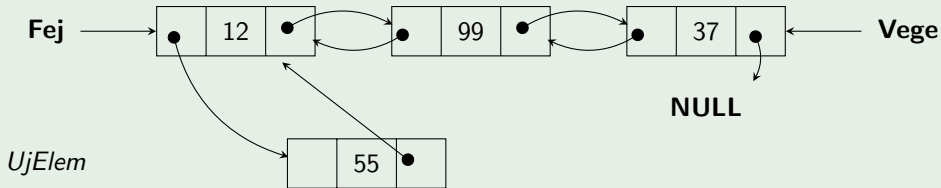
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás lista elejére

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $UjElem.köv \leftarrow Fej$
- 3  **$Fej.elozo \leftarrow Ujelem$**
- 4  $Fej \leftarrow UjElem$



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

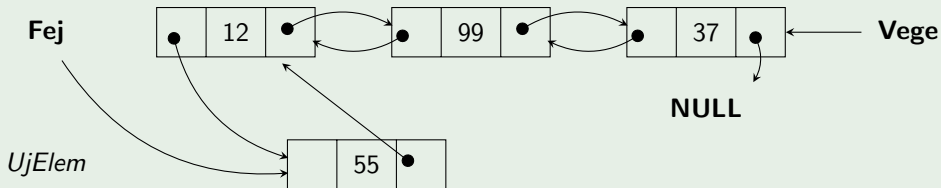
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás lista elejére

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $UjElem.köv \leftarrow Fej$
- 3  $Fej.elozo \leftarrow Ujelem$
- 4  $Fej \leftarrow UjElem$



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

Beszúrás lista végére - hasonlóan, mint a lista elejére

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $Vege.köv \leftarrow UjElem$
- 3  $UjElem.elozo \leftarrow Vege$
- 4  $Vege \leftarrow UjElem$

# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás $p$ elem elé

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $p.elozo.köv \leftarrow UjElem$
- 3  $UjElem.elozo \leftarrow p.elozo$
- 4  $UjElem.köv \leftarrow p$
- 5  $p.elozo \leftarrow UjElem$

# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Beszúrás $p$ elem után

Műveletek:

- 1  $UjElem \leftarrow nodeCreate(adat)$
- 2  $p.köv.elozo \leftarrow UjElem$
- 3  $UjElem.köv \leftarrow p.köv$
- 4  $UjElem.elozo \leftarrow p$
- 5  $p.köv \leftarrow UjElem$



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Törlés: `delete(p)`

Törlési lehetőségek:

- első elem
  - utolsó elem
  - nem első és nem utolsó (belső)
- 
- **előfeltétel:** Az adott  $p$  cím egy létező listaelem címe.
  - **utófeltétel:** Az adott  $p$  című mező törlődik a listából.

# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

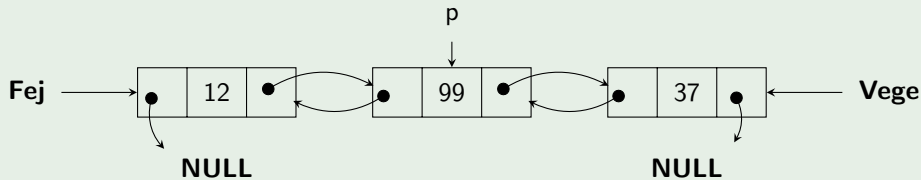
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Belső elem törlése

Műveletek:

- 1  $p.\text{elozo.kov} \leftarrow p.\text{kov}$
- 2  $p.\text{kov.elozo} \leftarrow p.\text{elozo}$
- 3  $\text{destroyNode}(p)$



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

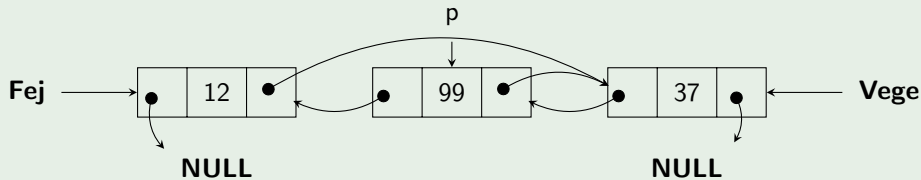
## Belső elem törlése

Műveletek:

1 **p.elozo.kov**  $\leftarrow$  **p.kov**

2 **p.kov.elozo**  $\leftarrow$  **p.elozo**

3 **destroyNode(p)**



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

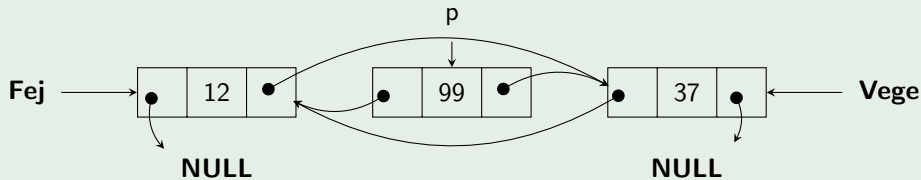
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Belső elem törlése

Műveletek:

- 1  $p.\text{elozo.kov} \leftarrow p.\text{kov}$
- 2  $p.\text{kov.elozo} \leftarrow p.\text{elozo}$
- 3  $\text{destroyNode}(p)$



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

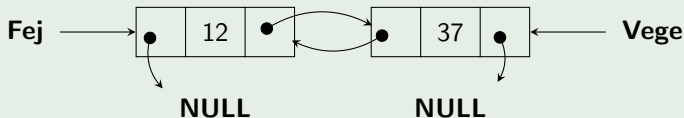
Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Belső elem törlése

Műveletek:

- 1  $p.\text{elozo.kov} \leftarrow p.\text{kov}$
- 2  $p.\text{kov.elozo} \leftarrow p.\text{elozo}$
- 3 **destroyNode(p)**



# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Első elem törlése

Műveletek:

- 1 Fej  $\leftarrow$  p.kov
- 2 p.kov.elozo  $\leftarrow$  *NULL*
- 3 destroyNode(p)

## Utolsó elem törlése

Műveletek:

- 1 Vege  $\leftarrow$  p.elozo
- 2 p.elozo.kov  $\leftarrow$  *NULL*
- 3 destroyNode(p)

# Duplán láncolt lista AAT: műveletek

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Bejárás előlről

- **előfeltétel:** -
- **utófeltétel:** Valamennyi elemet érinti a lista első elemével kezdődően. Figyelem, az utolsó érvényes elem az utolsó elem, nem pedig a NULL.

## Bejárás hátulról

- **előfeltétel:** -
- **utófeltétel:** Valamennyi elemet érinti a lista utolsó elemével kezdődően. Figyelem, az utolsó érvényes elem az első elem, nem pedig a NULL.

# Duplán láncolt lista AAT: másképp

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Egyszerűsítés?

- Lehet-e két strázsával implementálni a duplán láncolt lista AAT-t?
- Egyszerűbb lesz?
- Egy strázsával is lehetséges?



Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Körkörösén láncolt lista AAT

# Körkörösén láncolt lista AAT

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Szerkezet, tulajdonságok

- Gyakorlatilag egy egyszeresen vagy duplán láncolt lista.
- Az elző elem előzője az utolsó elem.
- Az utolsó elem következője az első elem.
- Csak a *Fej* mutatóra van szükség, viszont arra szükség van, hogy legyen belépési pont a listába

# Körkörösén láncolt lista AAT

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

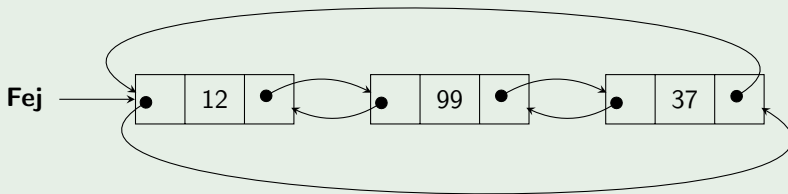
Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Példa: háromelemű körkörösén láncolt lista

- a 12-es adatot tartalmazó elem a **listafej**, előzője az *utolsó*
- a listához a listafejen keresztül férünk hozzá
- minden listaelem tárol egy mutatót, mely a **következő listaelem címét** tartalmazza (ha duplán láncolt, akkor *előzo* mutató is van)
- a 37-es adatot tartalmazó elem következője *null*, ez a lista **vége**



# Duplán láncolt lista AAT - egy strázsával

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

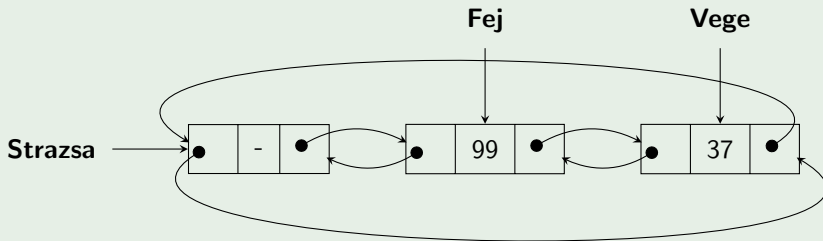
Duplán láncolt  
lista AAT

Körkörös  
láncolt lista AAT

Rendezett lista  
AAT

## Példa: duplán láncolt lista egy strázsával

- Valódi Fej:  $Fej := strazsa.kov$
- Valódi Vége tulajdonsága:  $Vege.kov := strazsa$



Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

Rendezett lista AAT

# Körkörösén láncolt lista AAT

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Szerkezet, tulajdonságok

- Gyakorlatilag egy egyszeresen vagy duplán láncolt lista.
- A lista elemei minden pillanatban rendezettek (ha vannak).

## Új művelet: rendezett beszúrás

Megkeressük az elem leendő helyét a listában, majd a már ismert algoritmusok egyikével beszúrjuk.

- **előfeltétel:** -
- **utófeltétel:** A lista tartalmazza az új elemet, és elemei rendezettek.

Adatszerkezetek

Vekov Géza

Láncolt lista AAT

Egyszeresen  
láncolt lista AAT:  
listaelem

Duplán láncolt  
lista AAT

Körkörösén  
láncolt lista AAT

Rendezett lista  
AAT

## Kérdések

Kérdések? 😊