

7. Óra

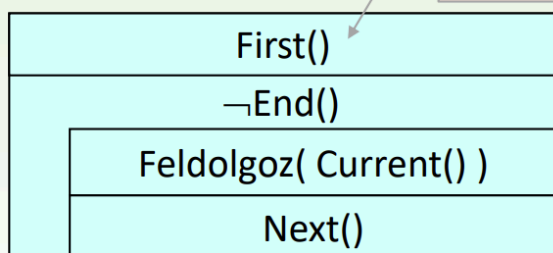
Miről lesz szó

Felsorolható objektumok (**IEnumerable**) és felsorolók (**IEnumerator**)

Felsorolás algoritmus

- Célszerű a gyűjtemények felsorolását olyan algoritmusba ágyazni, amely garantálja, hogy a felsorolás műveletei mindig csak a számukra „értelmes” állapotban kerüljenek végrehajtásra.

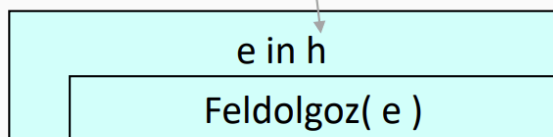
gyűjteményt felsoroló műveletek



```
for( First(); !End(); Next() )
{
    Process(Current());
}
```

Itt h egy felsorolható (**IEnumerable**) objektum, amelynek tartalmát a foreach ciklus nem változtathatja meg. Ha azonban h-ban objektumok, pontosabban azok hivatkozásai vannak, akkor azok adattagjai módosíthatók.

felsorolandó gyűjtemény



```
foreach( var e in h )
{
    Process(e);
}
```

a felsorolt elemek típusa automatikusan kideríthető

```
interface IEnumerable { // Felsorolható objektum
    public IEnumerator GetEnumerator();
}

interface IEnumerator { // Felsoroló
    public object Current; // ⇔ Current()
    public bool MoveNext(); // ⇔ Next(), True ha sikerült mozgatni, False ha nem
    public void Reset(); // ⇔ First()
}
```

Ezt az interface-t megvalósítja a List és tömb is!

```
IEnumerable enor1 = new List<int>() { 1, 2, 3, 4 };
IEnumerable enor2 = new int[] { 1, 2, 3, 4 };

IEnumerator etor = enor1.GetEnumerator();
```

```

etor.Reset();
while (etor.MoveNext()) {
    Process(etor.Current); // itt csinálhatunk valamit vele
}

// vagy

foreach (var e in enor1) {
    Process(e); // itt csinálhatunk valamit vele
}

```

1-n kapcsolat

🔗 Az 1-n kapcsolatot az „1” oldali objektumban tárolt pointer-gyűjtemény tárolja. A kapcsolat felpopulálásakor az objektumok memóriacímeit kell ebbe elhelyezni.

Magyarul:

Egy 1-n kapcsolatban az "1" oldalán egy gyűjtemény (általában lista) tartja magában a másik oldalról hozzá tartozó objektumokat (ezek pointerek, mert az objektumok a heapen vannak, C#-ban egy sima objektum listát jelent ez).

A gyűjtemény nevét a kapcsolat másik oldalán jelzik (nem az "1" oldalán)

A nyíl típusa a kapcsolat típusát írja le (Asszociáció: pöttyös nyíl, Aggregáció: üres rombusz, Kompozíció: teli rombusz, ...) [Előadás diái](#)

📦 A kapcsolat fajtája lehet:

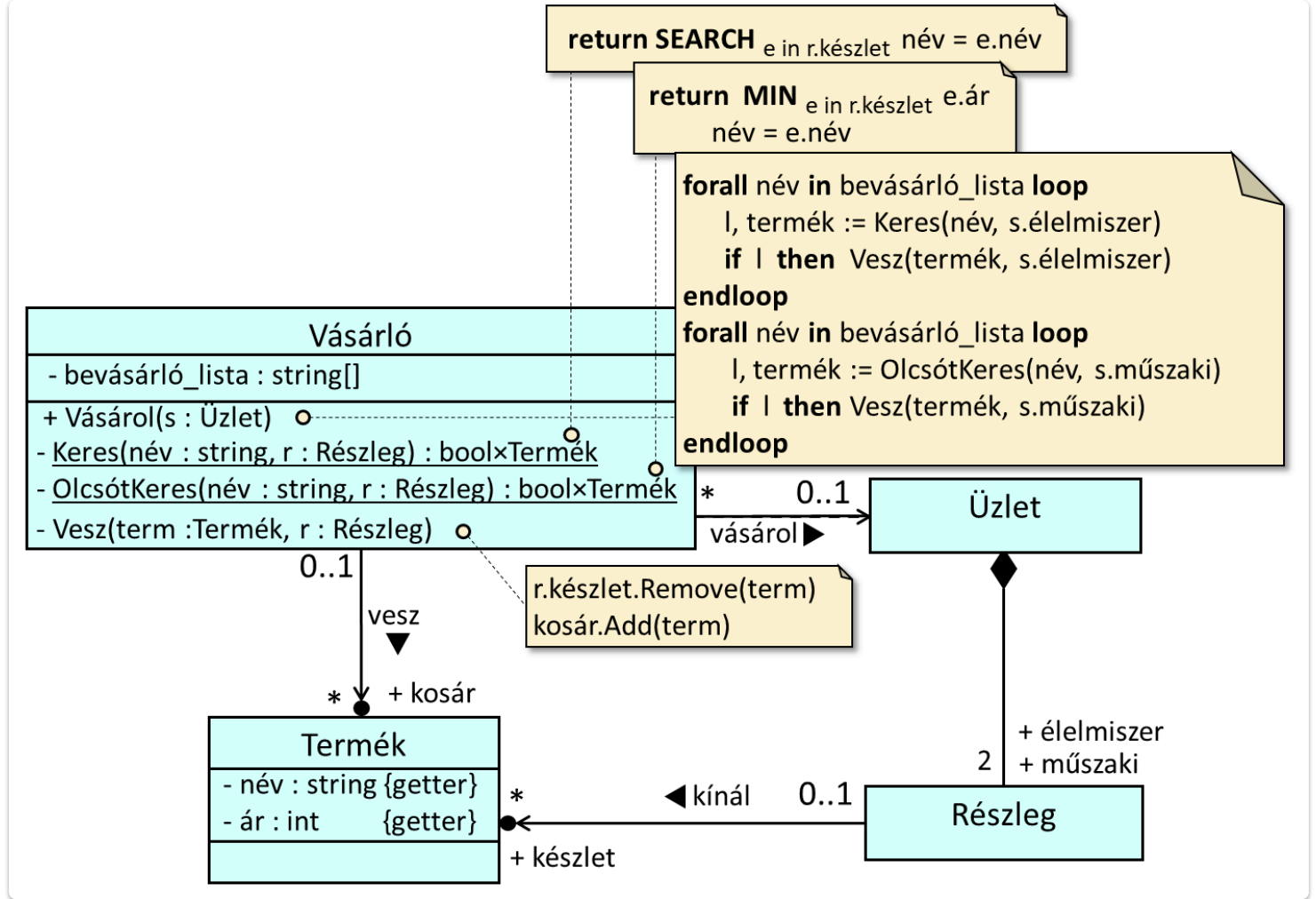
- **Függőség** (*dependency*)
- **Asszociáció** (*association*) vagy társítás
- **Aggregáció** (*shared aggregation*) vagy tartalmazás
- **Kompozíció** (*composite aggregation*) vagy szigorú tartalmazás
- **Származtatás** vagy öröklődés (*inheritence*)

Pl: (a feladatból, lásd alább)

- **Vásárló 0..1** → *** Termék**
 - **1 Vásárlóhoz bárhány Termék** tartozhat (akár 0)
 - **1 Termékhez 0 vagy 1 Vásárló** tartozhat
 - Kapcsolatot tárolja: **Vásárló kosár** adattagja
- **Részleg 2** --- **1 Üzlet** (itt az 1 az default, ha nincs odaírva)
 - **1 Részleghez pontosan 1 Üzlet** tartozik
 - **1 Üzlethez pontosan 2 Részleg** tartozik
 - Kapcsolatot tárolja: **Üzlet élelmiszer** és **műszaki** adattagja

Feladatok

Vásárlás



Kisbeadandó (7/10) (4. batch)

