



# PROGRAMOZÁS II

## 5. ÓRA: ÖRÖKLÉS VAGY TARTALMAZÁS

# Átfogó példa

- Állatkórház megvalósítása
  - Állatok
  - Gazdák
  - Személyzet
- Milyen osztályok legyenek?
- Hogyan kapcsolódjanak egymáshoz?

# OOP

- Használati esetek



- Domain modell



- Design modell



- Implementáció

# Használati esetek

- Felhasználás folyamata
  - Felhasználók célja
  - Lépések
  - Előfeltételek
  - Következmények
  - Kivételek, speciális esetek

# Felhasználási eset: példa

- Állatkórházban a recepciós ki tudja listázni minden gazda minden állatát.

# Domain modell: koncepcionális osztályok

- Főnevek?
- **Állatkórház**ban a **recepció**s ki tudja listázni minden **gazda** minden **állat**át.
- Osztályok:
  - Állatkórház
  - Recepció **?**
  - Gazda
  - Állat

# Domain modell: koncepcionális osztályok

- Főnevek?
- **Állatkórház**ban a **recepció**s ki tudja listázni minden **gazda** minden **állat**át.

- Osztályok:

- Állatkórház
- ~~Recepció~~s
- Gazda
- Állat

A recepció ebben a felhasználási esetben magát a felhasználót jelöli

Az általa végzett műveletek megvalósításáért a felhasználói interfész és a vezérlő osztályok felelősek

Ettől még lehet osztály a kódban, ha máshol szükség van rá

# Domain modell: koncepcionális osztályok

**Gazda**

**Allat**

**AllatKorhaz**



# Domain modell: attribútumok

- Gazda
  - Név
  - Lakcím
  - Telefonszám
  - Születési idő
- Állat
  - Név
  - Születési idő
  - Fajta
- Állatkórház
  - Név
  - Cím
  - Telefonszám

# Domain modell: attribútumok

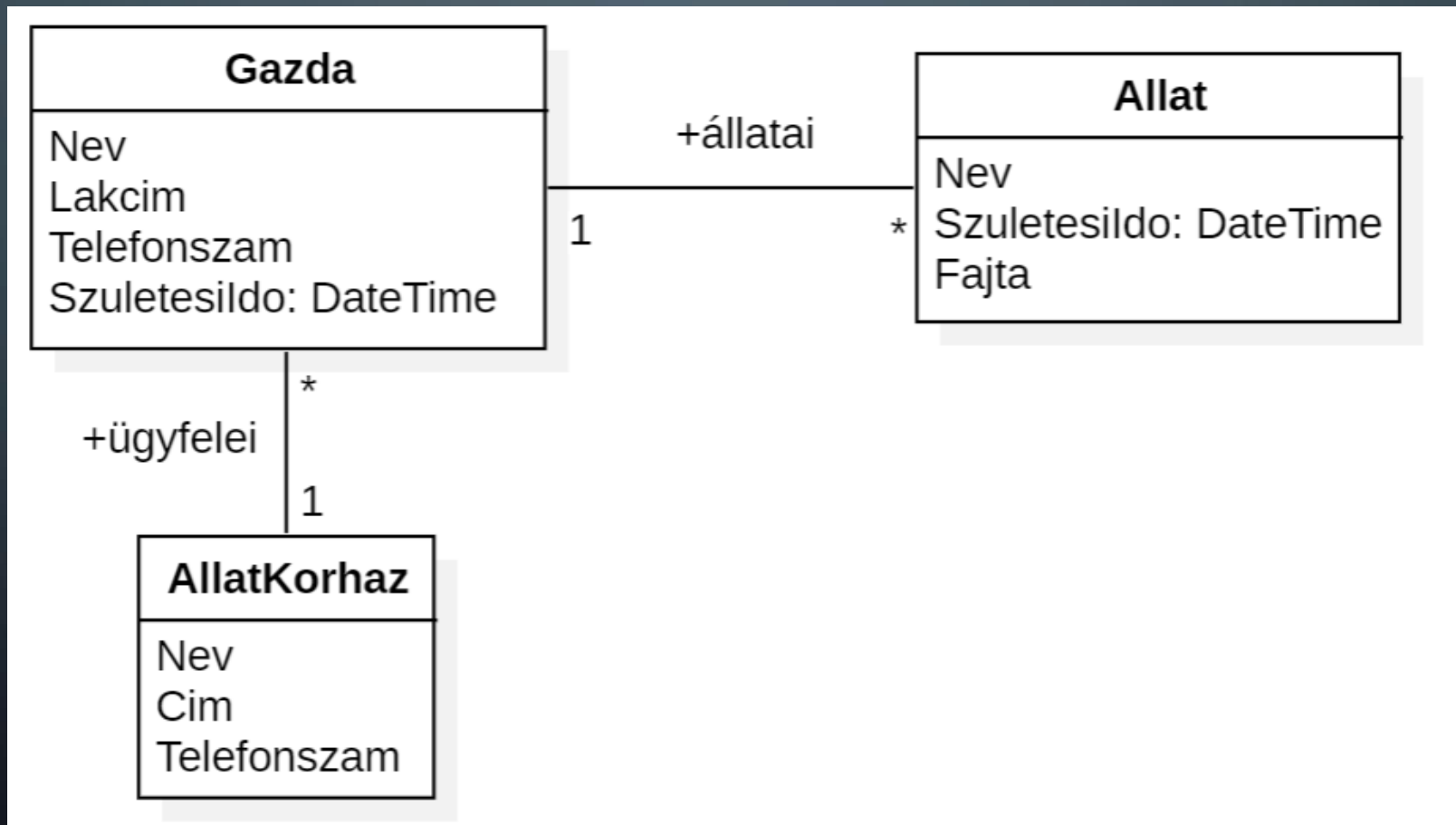
Gazda
Nev
Lakcim
Telefonszam
Szuletesido: DateTime

Allat
Nev
Szuletesido: DateTime
Fajta

AllatKorhaz
Nev
Cim
Telefonszam

# Domain modell: asszociációk

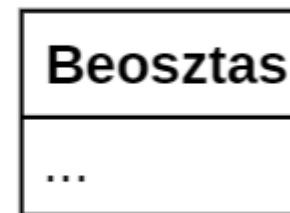
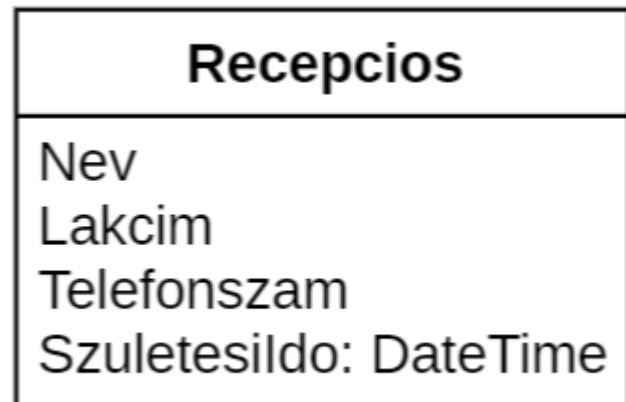
- Gazda állatai
- Állatkórház ügyfelei



# Konceptcionális osztály: recepció

- Erre is szükség lesz, amikor a kórház alkalmazottjait szeretnénk kezelni
- Attribútumok:
  - Név
  - Lakcím
  - Telefonszám
  - Születési idő
  - **Beosztás**

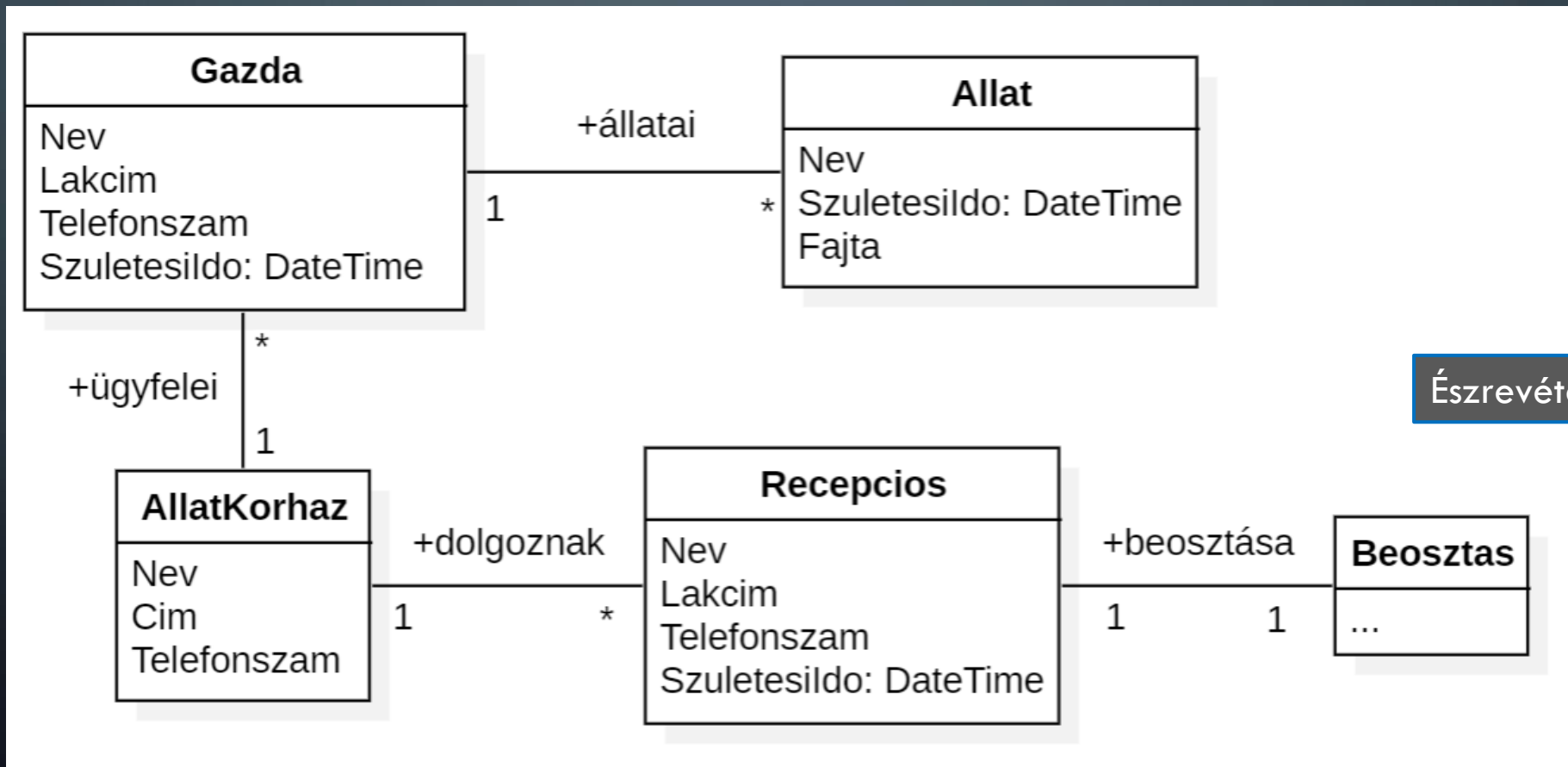
Újabb összetett koncepció, újabb osztály



A beosztás részleteibe most nem megyünk bele

# Recepció: asszociációk

- A kórházban dolgoznak recepciósok
- Minden recepciósnak van egy beosztása



Észrevételek?

# OSZTÁLYDIAGRAM

# Mi az osztálydiagram?

- Modell, ami tartalmaz:
  - Osztályokat
  - Azok elemeit
  - Osztályok közötti kapcsolatokat
- Célszerű rajzoló szoftvert használni
  - Umbrello
  - StarUML
  - Lucidchart
  - draw.io
  - ...

# Miben más a domain modelltől?

- Konkrétumokat is mutat
  - Attribútumok típusa, láthatósága
  - Metódusok formája, láthatósága
  - Kapcsolatok típusa
    - Tartalmazás (Composition)
    - Hivatkozás (Aggregation)
    - Általánosítás (Generalization)

Ez az öröklés



# Osztálydiagram: példa

A sor eleje jelöli  
a láthatóságot

## ExampleClass

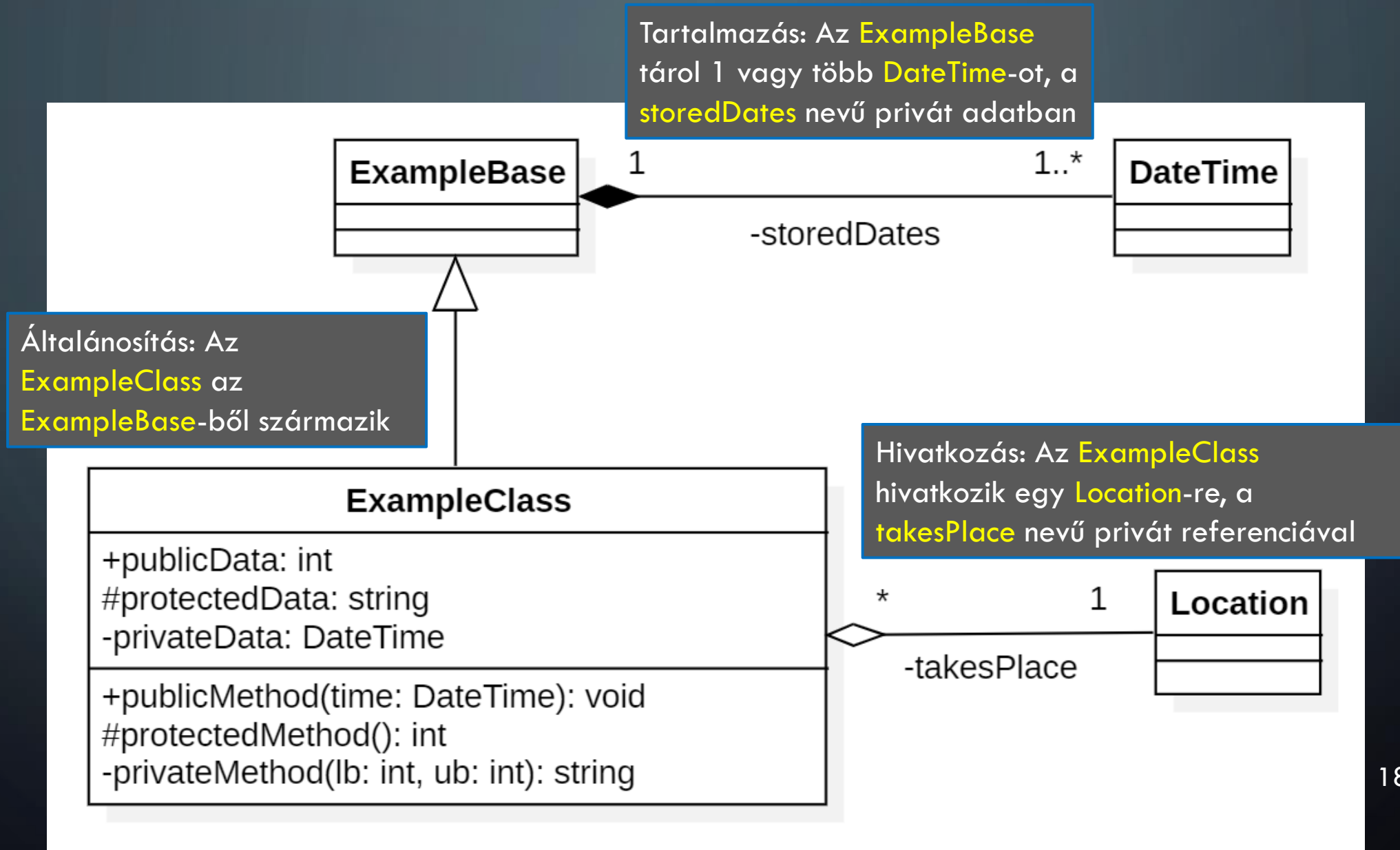
+publicData: int  
#protectedData: string  
-privateData: DateTime

A végén ott a típus

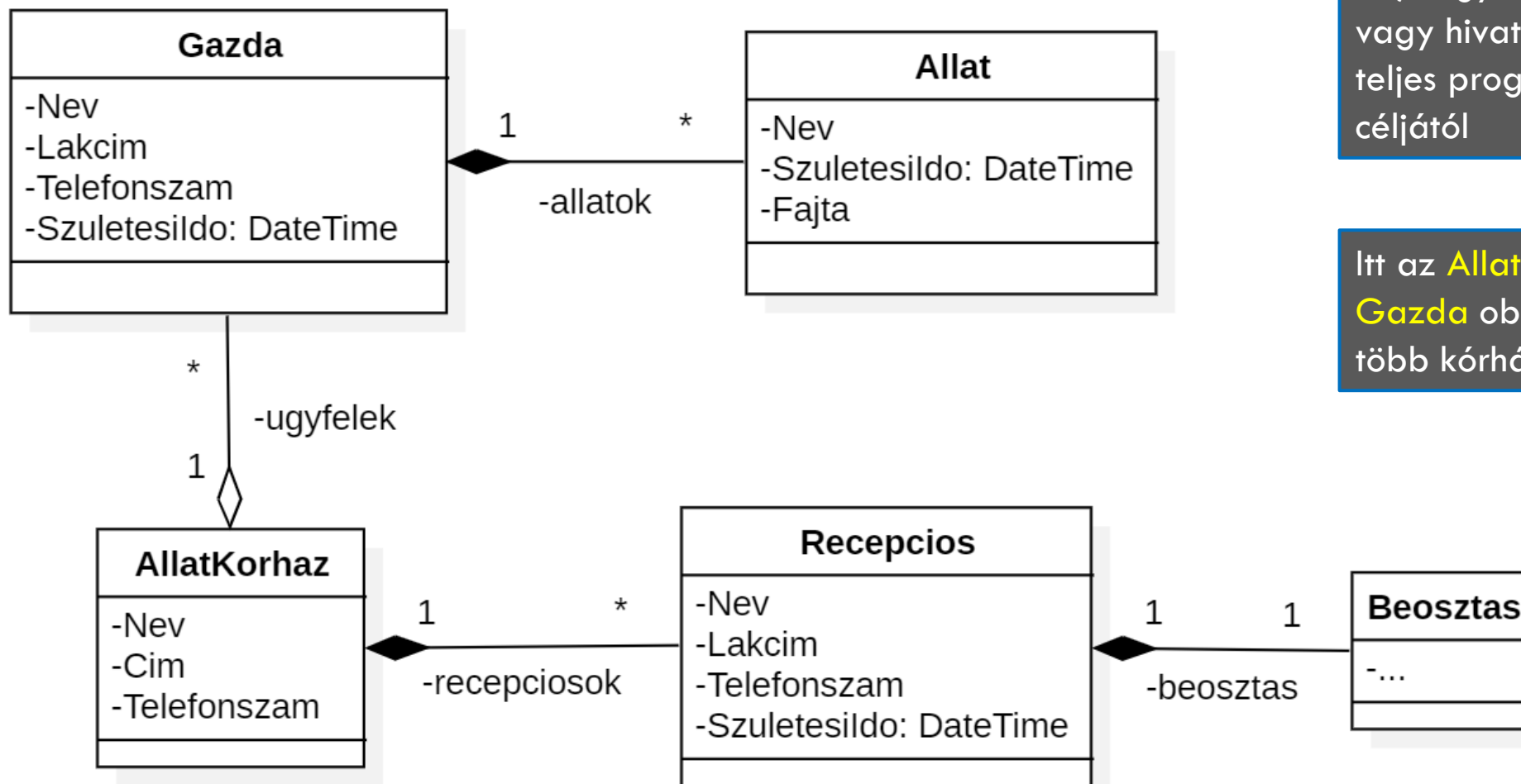
+publicMethod(time: DateTime): void  
#protectedMethod(): int  
-privateMethod(lb: int, ub: int): string

Metódusok  
paraméterekkel és  
visszatérési típusokkal

# Osztálydiagram: példa asszociációk



# Példa feladat: osztálydiagram



Az, hogy egy kapcsolat tartalmazás vagy hivatkozás nagyban függ a teljes program szerkezetétől és céljától

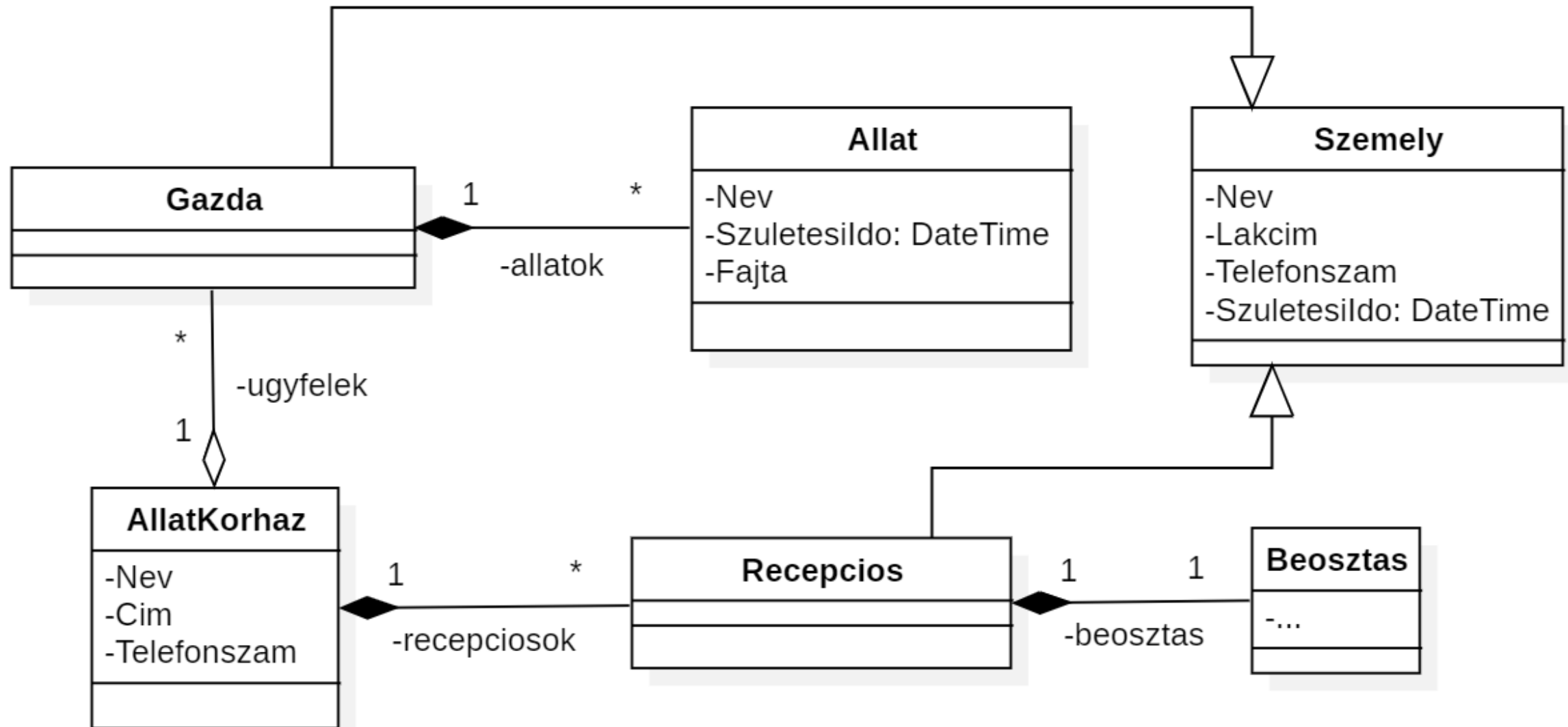
Itt az **AllatKorhaz** csak hivatkozik a **Gazda** objektumokra, ami jó, ha több kórház is van a programban

# Fejlesztési lehetőség?

- A gazda és a recepciós nagyon hasonló
- Lehet-e a kód ismétlést csökkenteni?
- Igen: öröklés
  - Legyen egy személy koncepció, ami általánosítja a gazdát és a recepcióst is

Lehetne arra is gondolni, hogy a **Személy** és az **Állat** általánosítható **Élőlénnyé**, de a program céljától függően most nincs erre szükség

# Példa feladat örökléssel

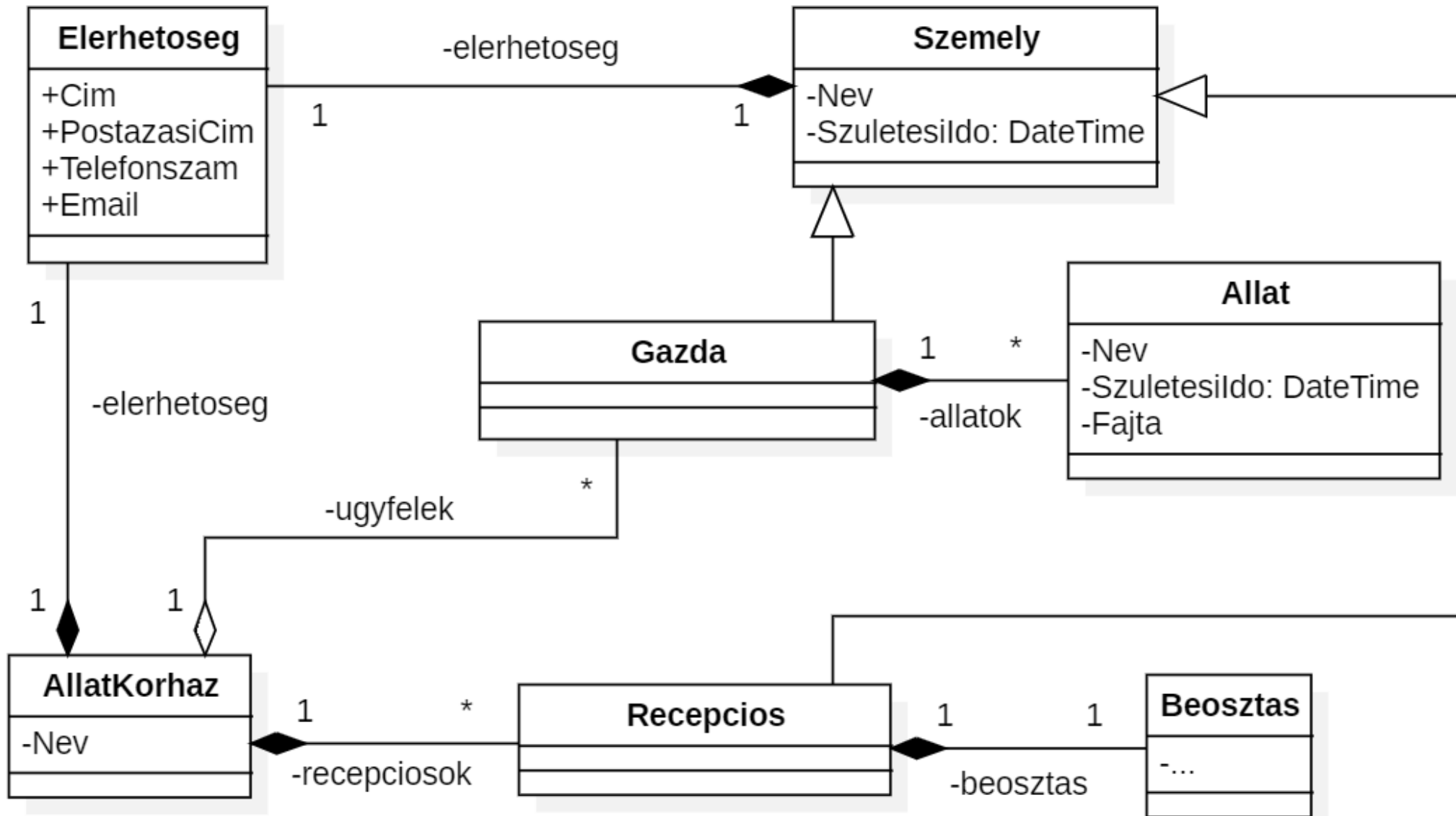


# Fejlesztési lehetőség?

- A személyeknek és a kórháznak egyaránt van címe és telefonszáma
  - Kiegészíthető lenne akár email-címmel, postázási címmel, stb.
- Lehet-e a kód ismétlést csökkenteni?
- Igen: delegáljuk az elérhetőségeket egy külön koncepcióba
  - A személy és a kórház is tartalmazni fog egy ilyen

Nem öröklés, mert sem a **Személy** sem a **Kórház** nem egy **Elérhetőség**, csak használják azt

# Példa feladat delegált elérhetőségekkel

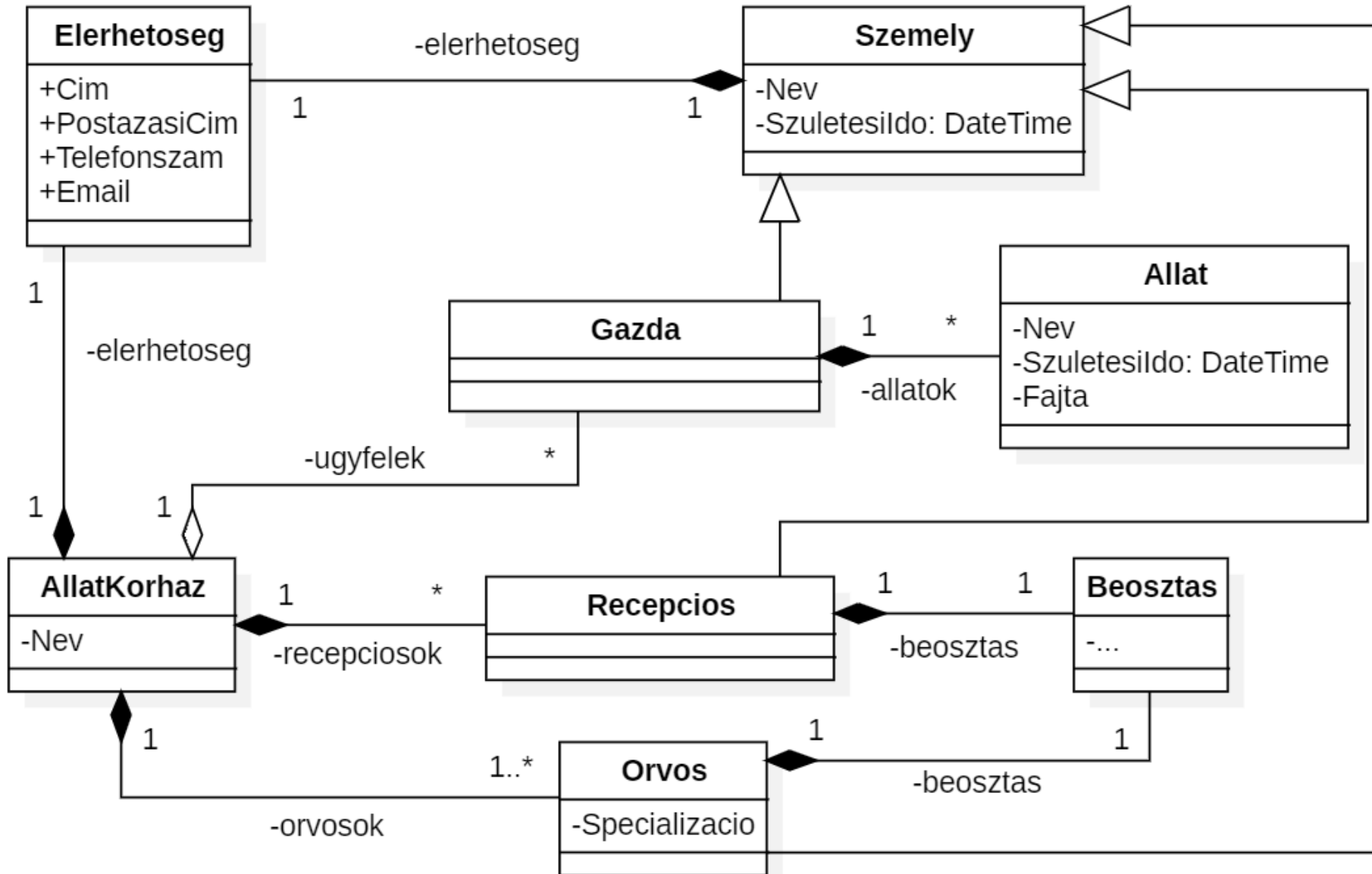


# Következő lépés: orvos

- Orvos kezelése a rendszerben
  - Új koncepció
- Az orvos is egy személy
- Attribútumok:
  - Specializáció
  - Beosztás



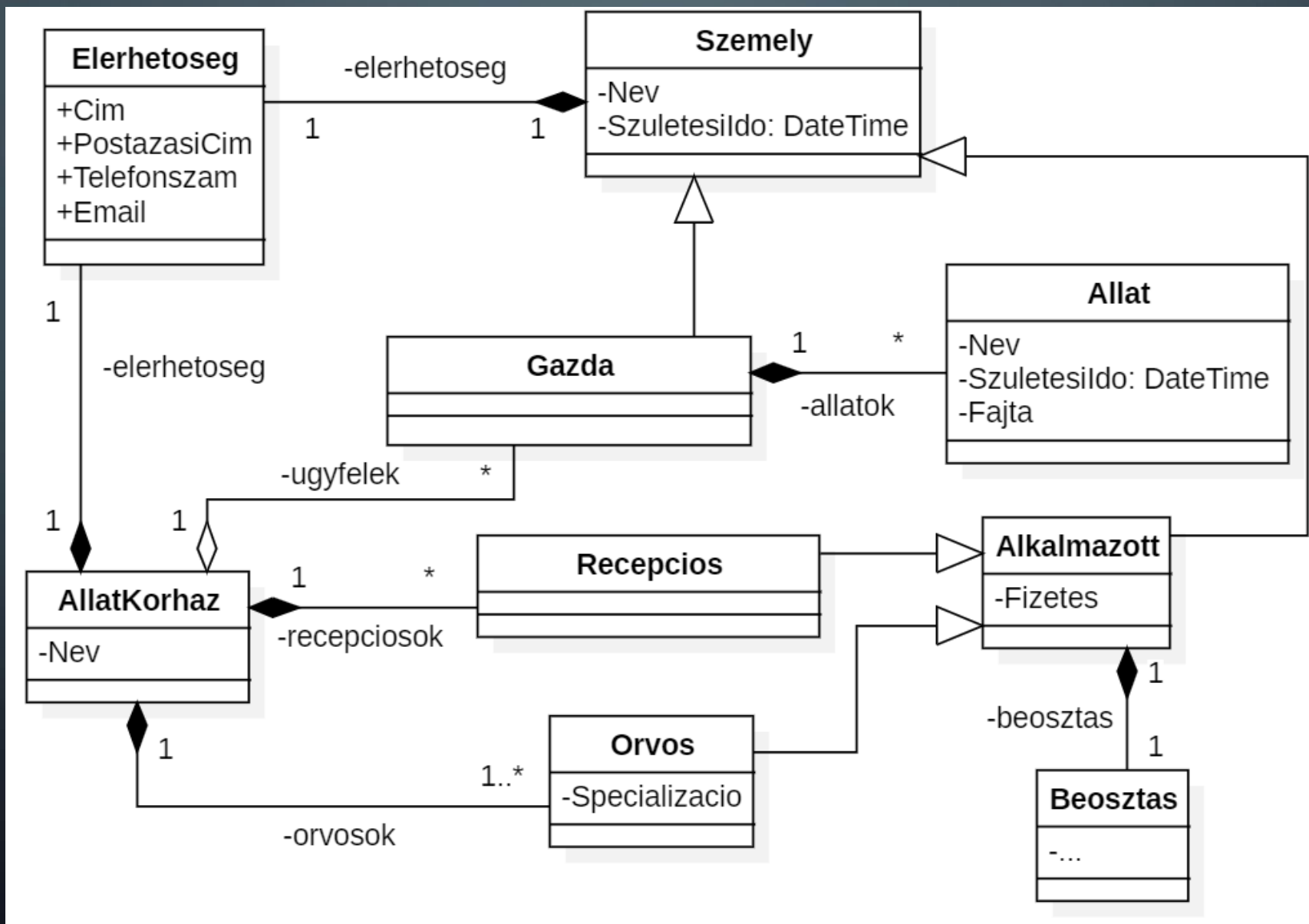
# Következő lépés: orvos



# Fejlesztési lehetőség?

- A recepciós és az orvos is személy, akiknek van beosztása
- Lehet-e a kód ismétlést csökkenteni?
  - Igen: öröklés
    - Legyen egy alkalmazott koncepció, ami általánosítja a recepcióst és az orvost is
    - Hozzá lehet tenni mindent, ami egy alkalmazott esetén fontos adat (pl. fizetés)

# Fejlesztés: Alkalmazott



# Felhasználási eset: folytatás

- Állatkórházban a recepciós ki tudja listázni minden gazda minden állatát.
- A teljes folyamat modellezéséhez kellenek a függvények
- Mit kezdünk a recepcióssal?
  - Ő itt a felhasználó
  - Kapcsolódni kell a felhasználói interfészhez

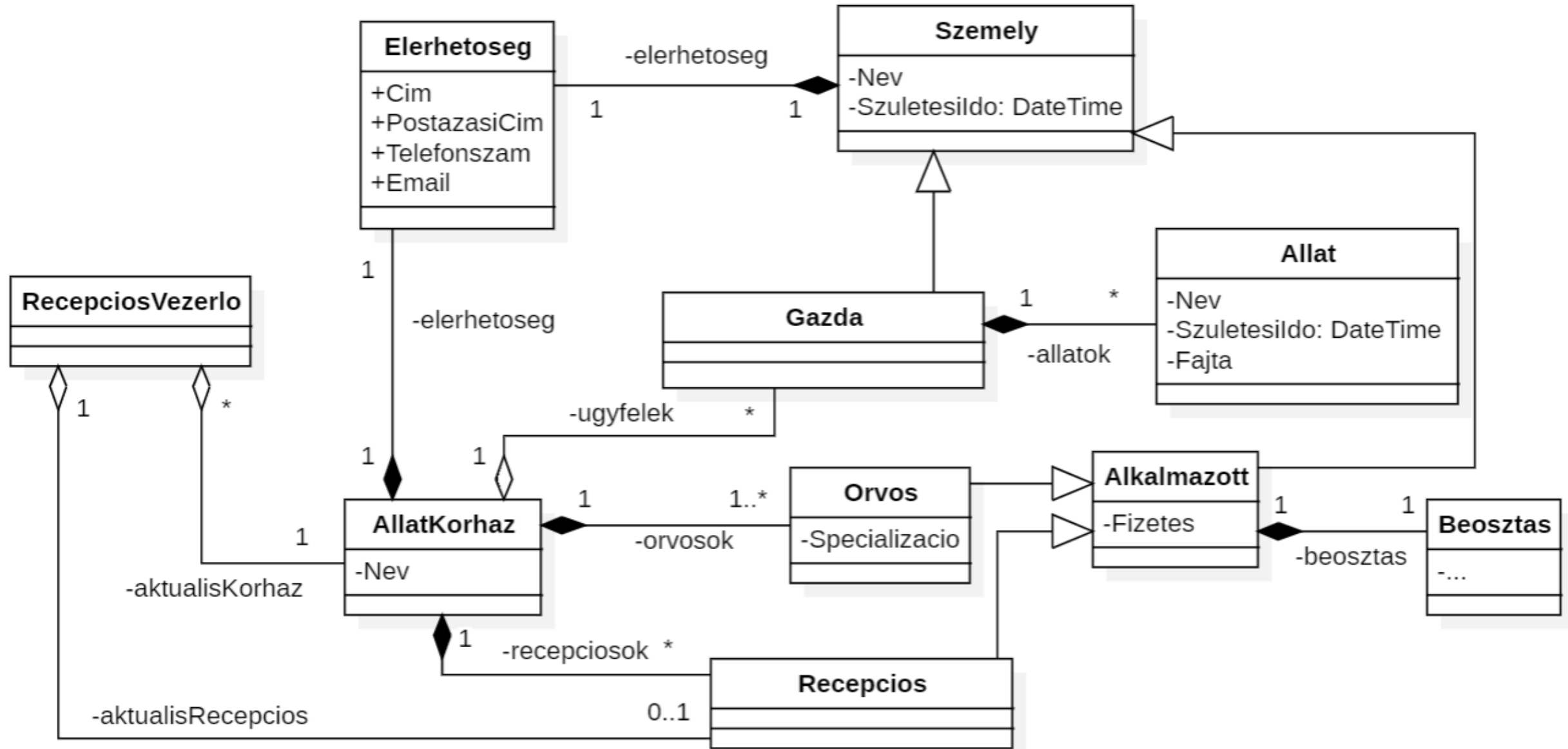
# Felhasználási eset: kapcsolódás a UI-hoz

- Nem szükséges most eldönteni, hogy a felület hogyan néz ki
- Elég a kapcsolatot megadni
  - Vezérlők

# RecepciosVezerlo

- Új osztály: RecepciosVezerlo
- Innen indulnak a recepciós által indított felhasználási esetek
- Attribútumok
  - Egyelőre nincs szükség konkrét attribútumot megnevezni
- Asszociációk
  - AllatKorhaz: ismernie kell, hogy melyik kórházzal dolgozik
    - Hivatkozás
  - Recepcios: Esetleg tudhatja, hogy melyik recepciós dolgozik éppen
    - Hivatkozás
    - Opcionális

# RecepciosVezerlo



# Metódusok

- Állatkórházban a recepciós ki tudja listázni minden gazda minden állatát.
- Milyen metódusok kellenek ennek a megvalósításához?



# Tudások

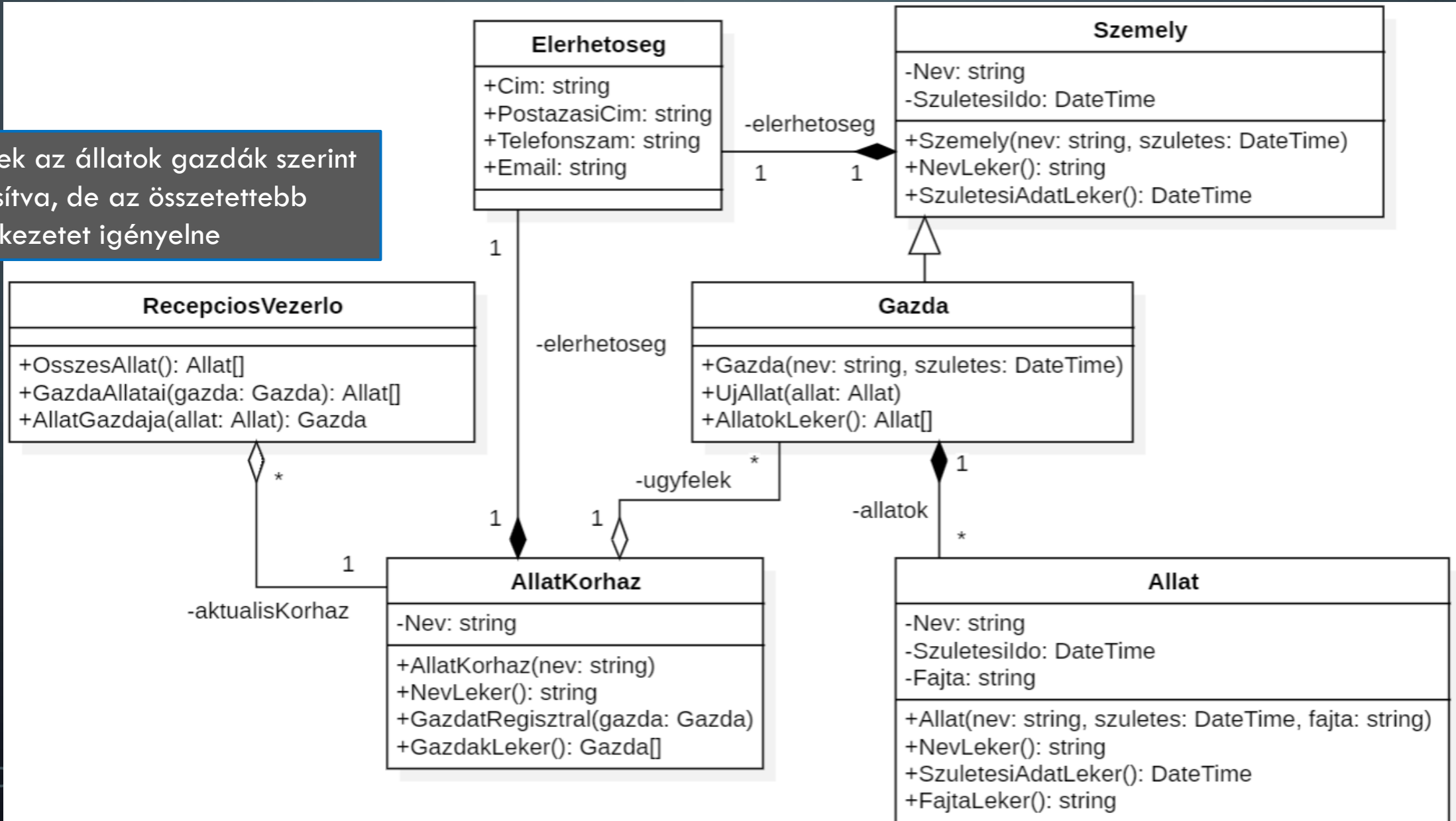
- Ki tárolja az állatok adatait?
  - Aki ismeri őket: Gazda (tartalmazás)
- Ki tárolja a gazdák adatait?
  - Aki mindet ismeri: Ezzel egyelőre nem foglalkoztunk
  - Ki kezeli még őket: Állatkórház (hivatkozás)
- Ki tárolja az Állatkórházat?
  - Senki: Ő reprezentálja a teljes tudásunk
  - Hogy érjük el: Vezérlőn keresztül (hivatkozás)

# Felelősségek

- Ki adja meg egy állat adatait?
  - Aki tudja: maga az Állat.
- Ki adja meg egy gazda állatait?
  - Aki tudja: Gazda
- Ki adja meg a Gazda adatait?
  - Aki tudja: maga a Gazda
- Ki adja meg a kórházban regisztrált összes gazdát?
  - Aki tudja: Állatkórház

# Metódusok

Lehetnének az állatok gazdák szerint csoportosítva, de az összetettebb adatszerkezetet igényelne



# Folyamat

