ADATSZERKEZETEK ÉS ALGORITMUSOK

OOP – Öröklés

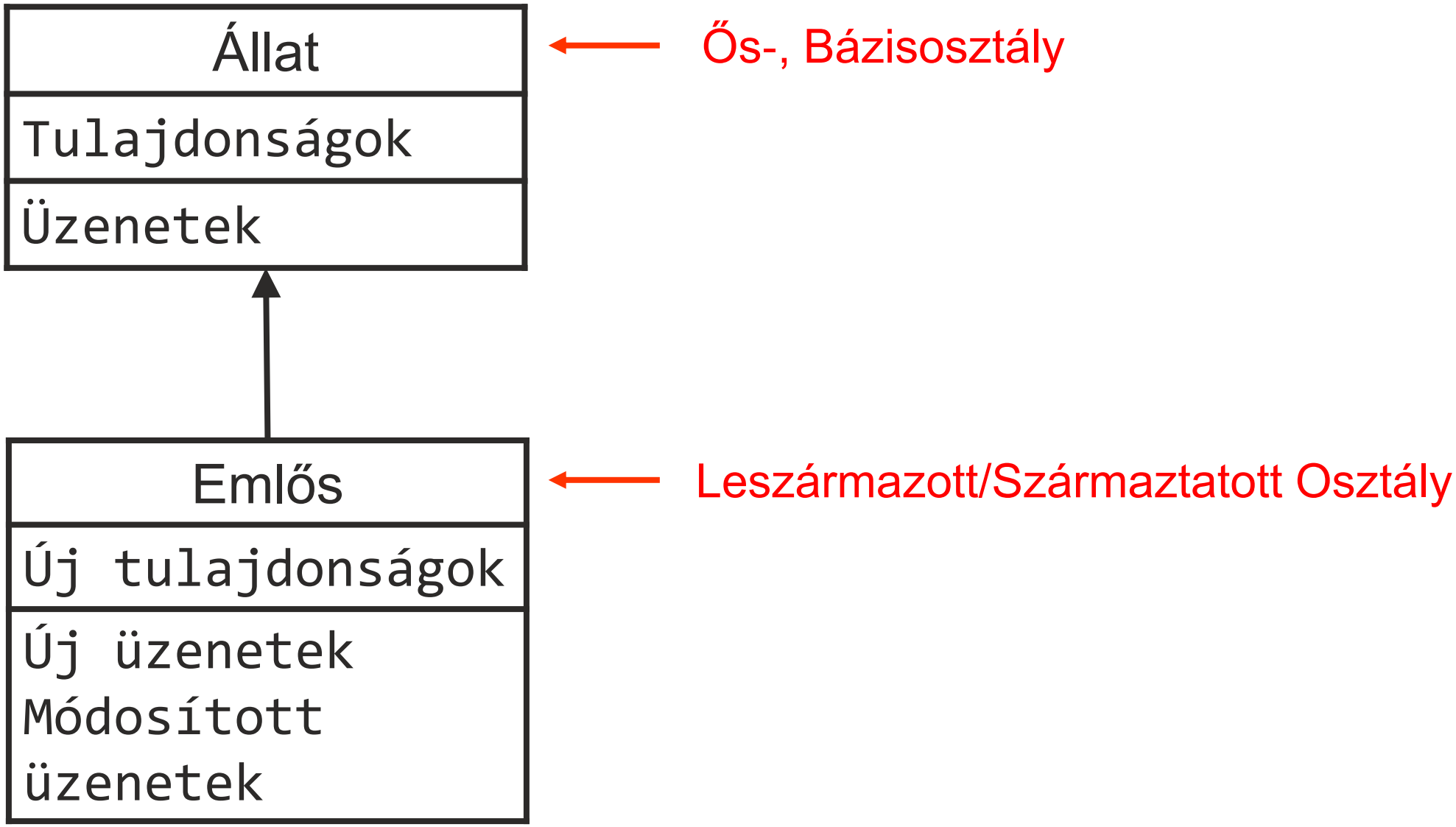
Öröklés

- Az alapgondolat: a gyerekek öröklik őseik metódusait és változóit
- Az örököl azt jelenti, hogy az ősosztály minden metódusa és adattagja a gyerekosztálynak is metódusa és adattagja lesz
  - A gyerek minden új művelete vagy adattagja egyszerűen hozzáadódik az örökölt metódusokhoz és adattagokhoz
  - Minden metódus, amit átdefiniálunk a gyerekben, a hierarchiában felülbírálja az örökölt metódust
- Lehet egyszeres, vagy többszörös
  - Ez a közvetlen ősosztályok számát jelenti

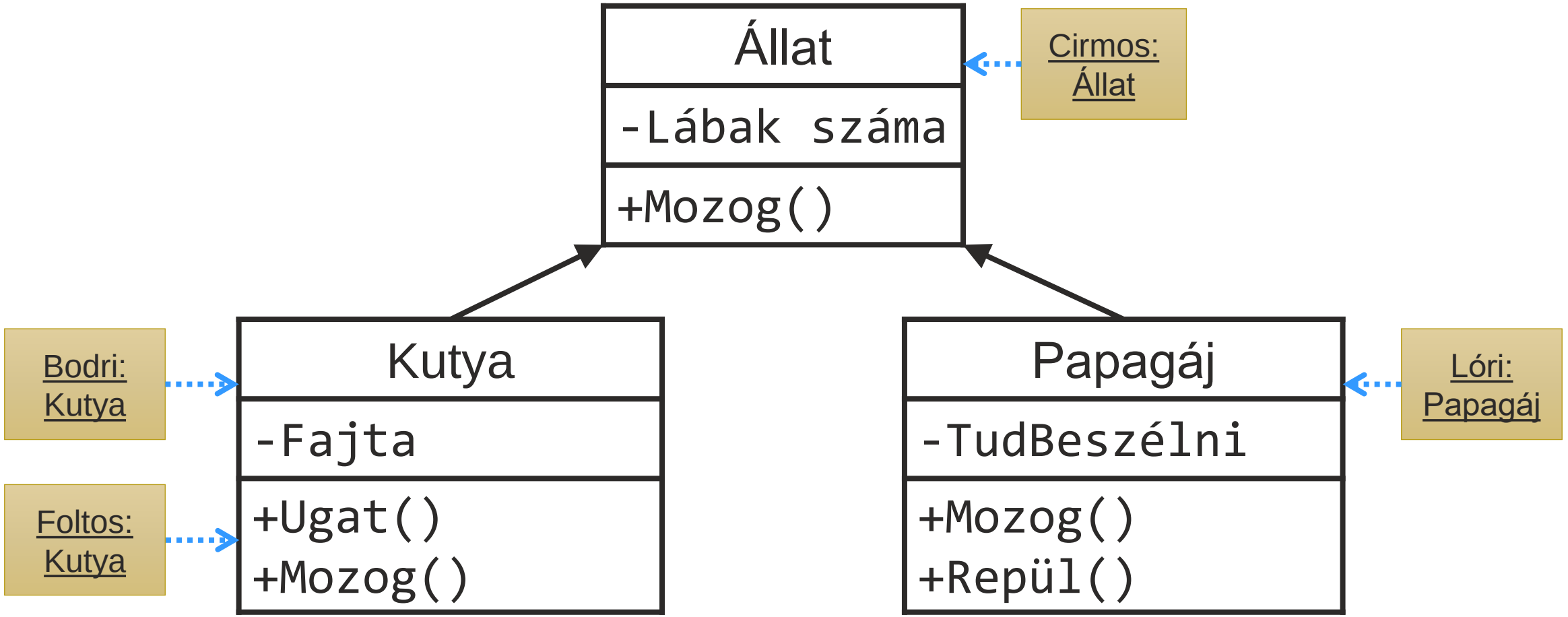
Öröklés

- Mit örököl a leszármazott?
  - Tulajdonságokat (Adattagokat)
  - Metódusokat (Üzeneteket)
- Mit nem örököl a leszármazott?
  - Ősosztály konstruktorait, destruktorát
    - Ám tudja használni / meghívni / delegálni
  - Ősosztály értékadás operátorát
- Mit tehet a leszármazott osztály?
  - Új tulajdonságokat vezethet be
  - Új metódusokat vezethet be
  - Felüldefiniálhat, vagy elfedhet már meglévőket
  - Új konstruktorokat és destruktort

Öröklés – IS-A reláció



Öröklés – IS-A reláció



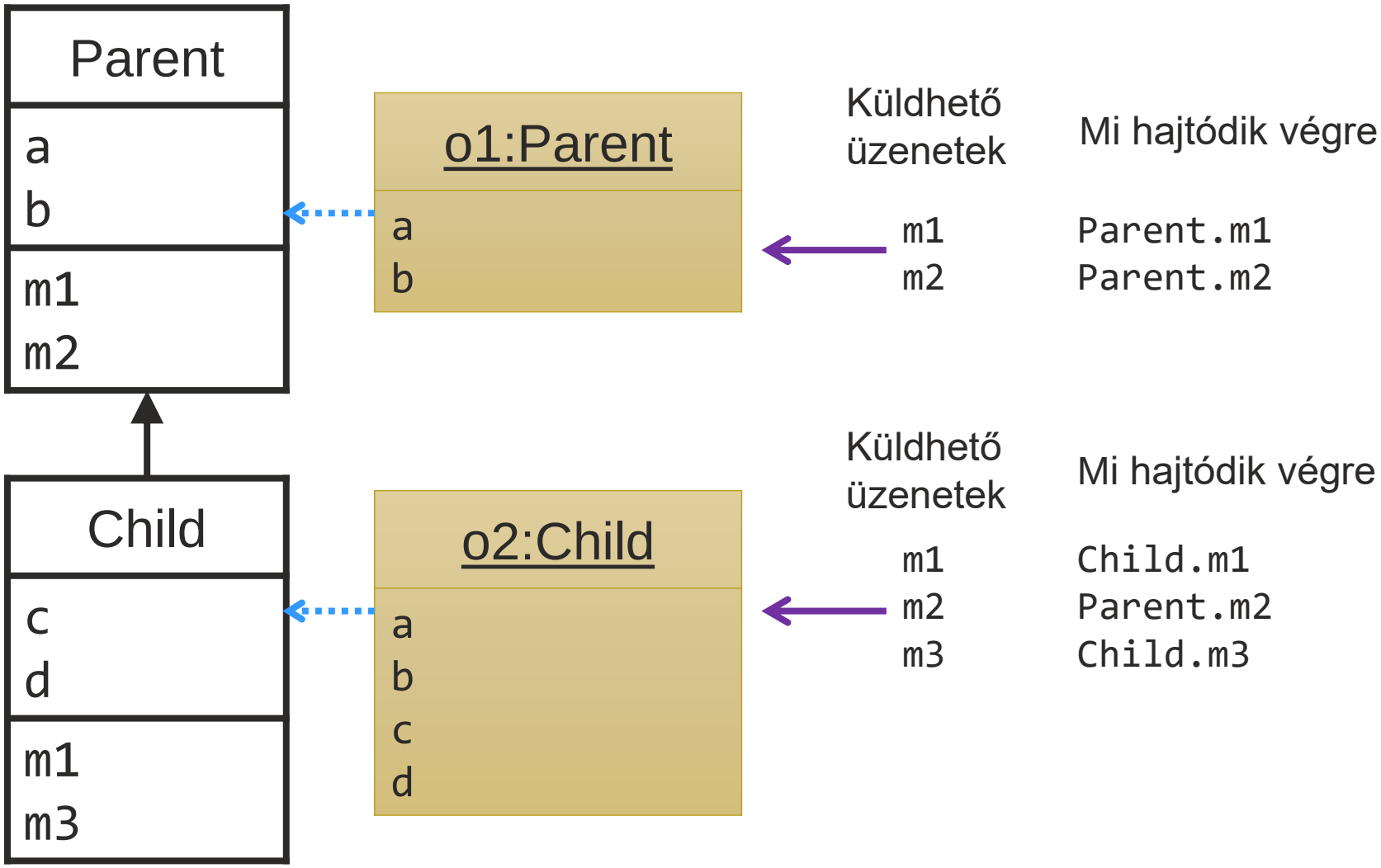
Implementáció újrahasznosítás

- Alosztályképzés
  - Használd egy típus implementációját egy másik típus implementálására!
  - Gyakran használjuk az őstípus implementációját az altípus implementálására
  - A gyakran használt OO programozási nyelvek keverik az altípus és alosztály fogalmakat.

Felüldefiniálás/Elfedés

- Két különböző jelenség
- Ami közös a felüldefiniálás (átdefiniálás) illetve elfedés során:
  - Az ősosztályban már meglevő, bevezetett függvénynek/üzenetnek ad új definíciót
    - A szignatúra azonos
- Ami különböző
  - A metóduselérés hogyan történik polimorfizmus esetén
  - A két működést kulcsszavakkal lehet szabályozni általában

Küldhető üzenetek, elérhető adatok



Polimorfizmus

- Polimorfizmus (többalakúság)
  - Az a jelenség, hogy egy azonosító (név: változó, függvény) nem csak egyfajta „objektumra” hivatkozhat.
- A polimorfizmusnak több formája is van
  - Altípusos
    - A (változó)név egy szupertípusra és leszármazottjaira hivatkozhat
  - Nyers – Duck typing
  - Parametrikus
    - Lényegében a sablonok
  - Ad hoc
    - Lényegében a függvények túlterhelése

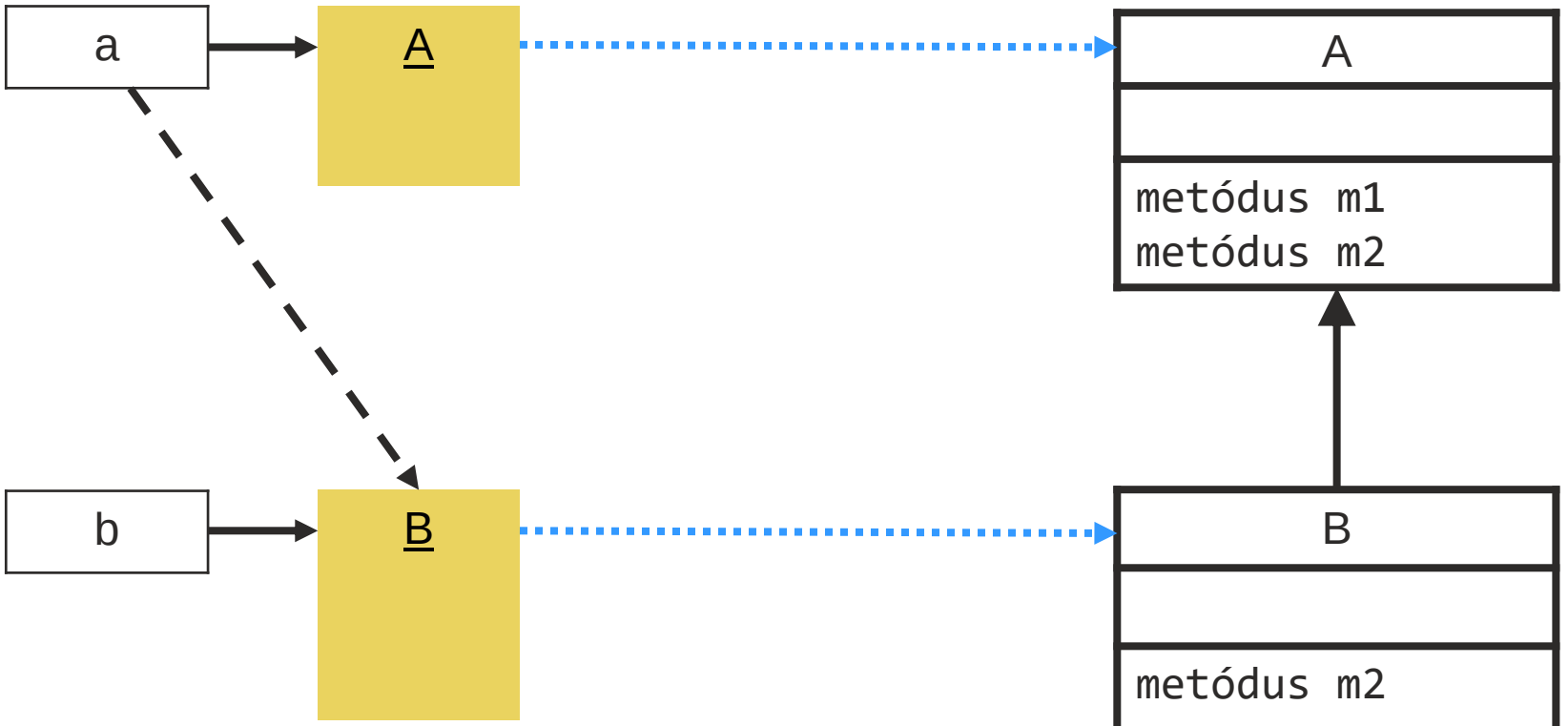
Altípusos polimorfizmus

- Változó nem csak egyfajta típusú objektumra hivatkozhat, hanem annak altípusaira is
  - Statikus típus: a deklaráció során kapja.
  - Dinamikus típus: futásidőben, éppen milyen típusú objektumra hivatkozik
    - a statikus típus, vagy annak leszármazottja

`A` típus altípusa `B`, ha a `B` minden olyan helyzetben használható, ahol az `A` is.

- A `Kutya` egyben `Állat` is (IS-A).
- A `Háromszög` az egy `Alakzat`.
- ***C++ esetén a public öröklődéssel létrejött leszármazottak altípusai az ősnek.***

Altípusos polimorfizmus



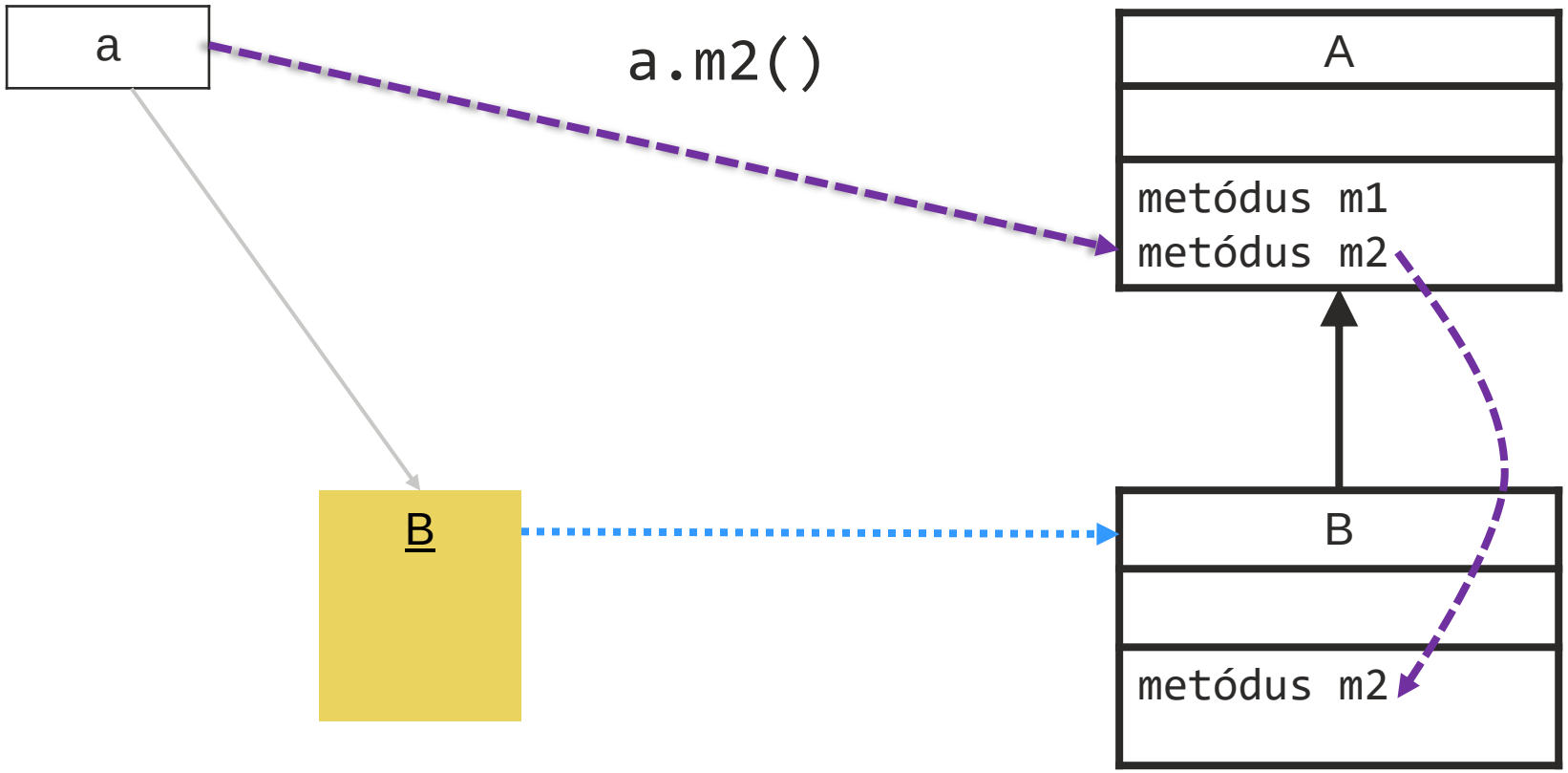
a:=b

Altípusos polimorfizmus

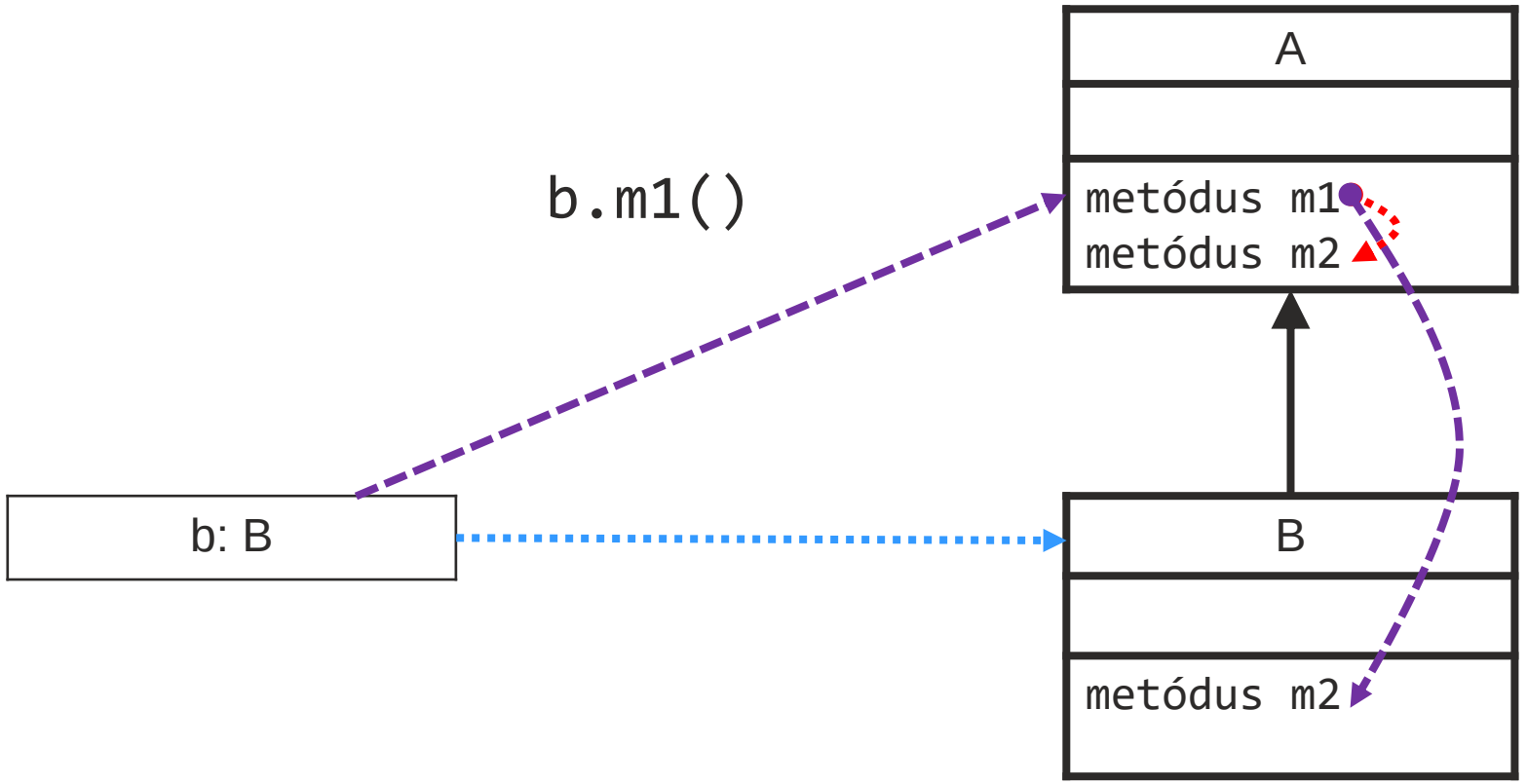
- Az előző példában a `a` változó
  - Statikus típusa: `A`
  - Értékadást követően a dinamikus típusa `B`

- Mi történjen, ha a `B` osztály által felüldefiniált `m2()` függvényt hívjuk az `A` statikus típusú változót használva?
  - Egy `B` objektum viselkedése az objektum típusának feleljen meg!
  - Ezt a dinamikus összekapcsolás valósítja meg

Dinamikus összekapcsolás



Dinamikus összekapcsolás

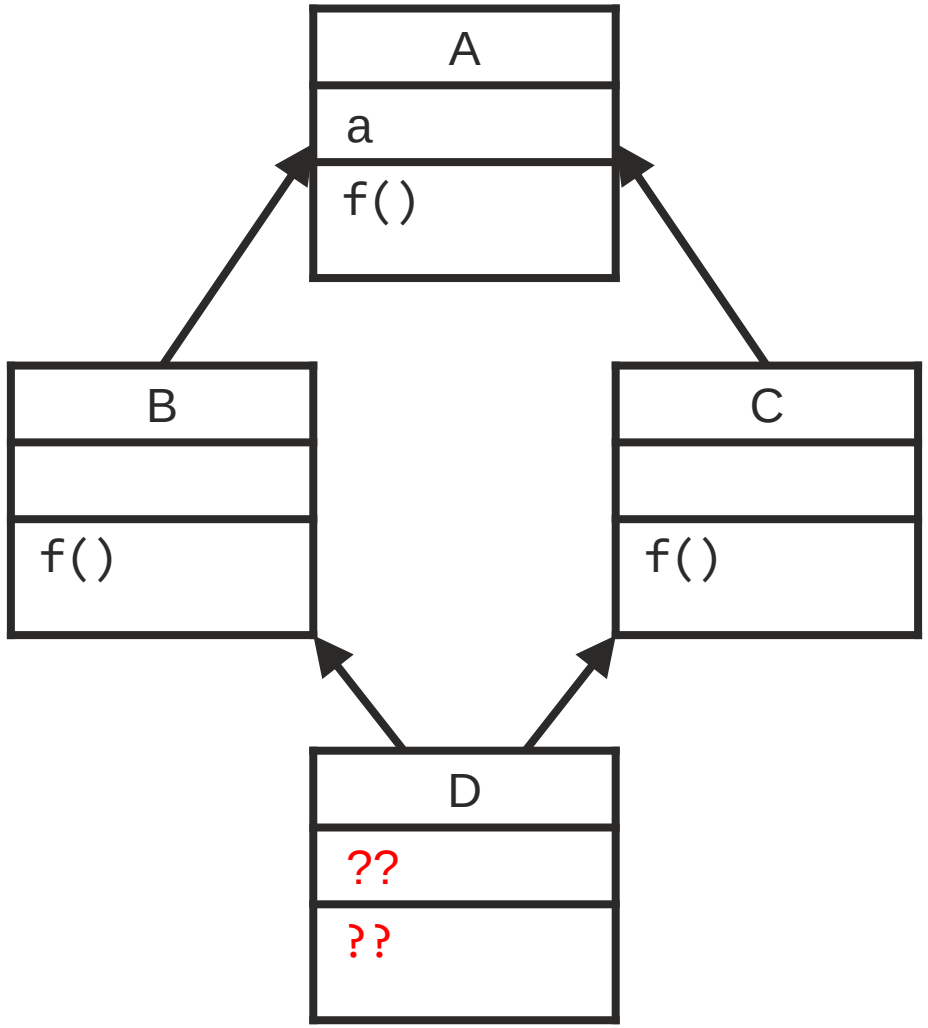


Dinamikus összekapcsolás

- Run-time fogalom
  - Az a jelenség, hogy a változó éppen aktuális dinamikus típusának megfelelő metódus implementáció hajtódik végre
  - Felüldefiniált függvények/metódusok esetén történik meg
    - Elfedés esetén nem!

A többszörös öröklődés problémái

- Diamond problem
  A bázisosztály definiál f függvényt és a attribútumot.
  - Ezt B és C osztályok függetlenül származtatják – felüldefiniálják
  - D osztály B és C osztályokból közvetlenül származik többszörös örökléssel
  - Ekkor
  1. A D-beli f melyiket jelentse?
  2. Az „a” attribútum hány példányban jelenjen meg D-ben?



A többszörös öröklődés problémái

- A két kérdés lényegében ugyanazt a problémát veti fel:
  ha kétértelműség van, hogyan válasszunk?
  - A legtöbb esetben az ilyen kódot nem lehet lefordítani
    - A fordító, vagy a futtató környezet kétértelműségre (*ambiguous*) hivatkozva hibajelzéssel leáll.
  - Megoldás
    - Az ősosztály mondja meg, hogy mit szeretne tenni ilyen esetben.
    - A származtatott osztály mondja meg, hogy melyiket szeretné használni.

Absztrakt osztály

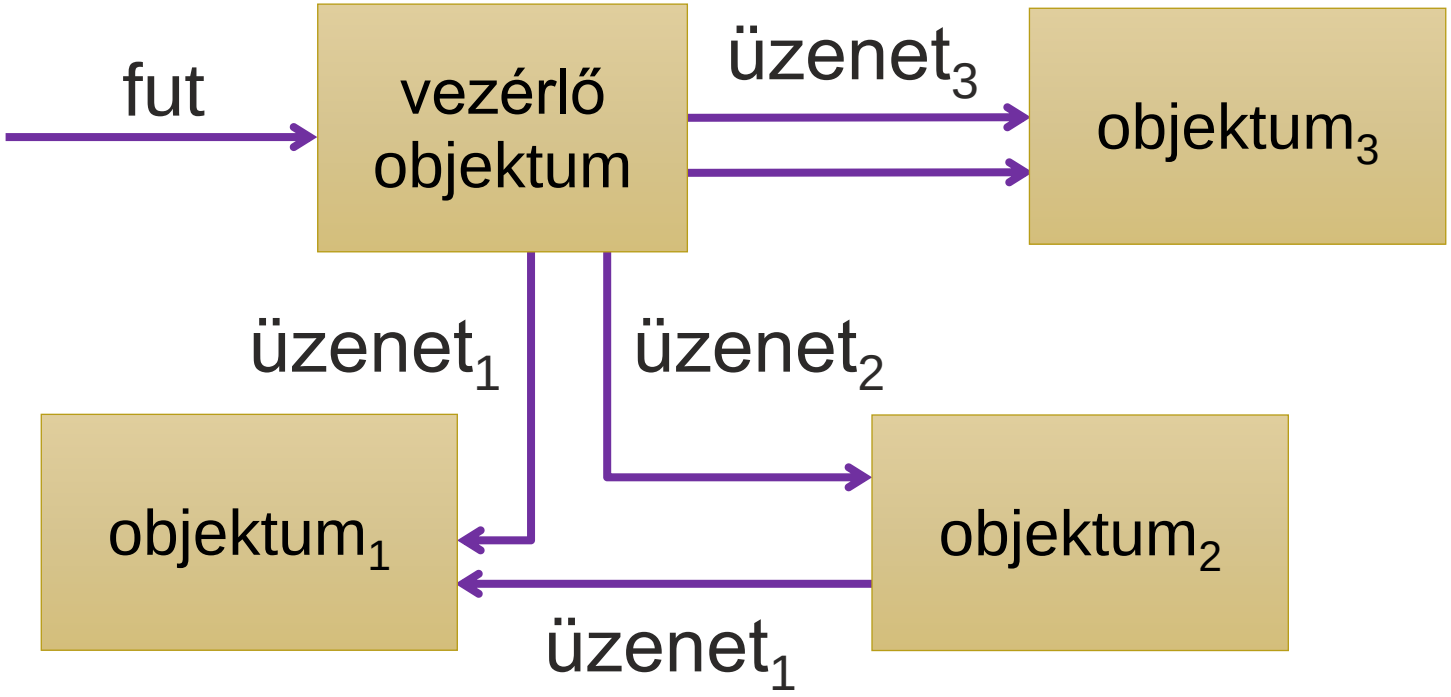
- Tervezés eszköze
- Egy felső szinten összefogja a közös tulajdonságokat
- A metódusok között van olyan, aminek csak specifikációja van, törzse nincs
- Nem hozható létre példánya.
- A leszármazott teszi konkréttá.

Mi az objektumorientált programozás?

- A programozó definiálhat altípus kapcsolatokat
- A típusszabályok megengedik, hogy az altípus használható legyen a szupertípus helyén
  - altípusos polimorfizmus
- Típus-vezérelt metódus elérés
  - dinamikus kötés
- Implementáció megosztása
  - öröklődés

OO program

- Egy objektumorientált program egymással kommunikáló objektumok összessége, melyben minden objektumnak megvan a feladatköre



Osztály, példány

- Minden objektum?
- Akkor az osztályok is …
  - Lehet belső állapotuk,
  - Küldhetünk üzeneteket neki …
  - Minek az objektuma?
    - Metaosztály
      - Singleton objektum
      - És a metaosztály is objektum?

Verem

Következő téma