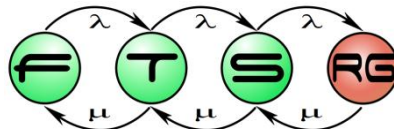


Címtár szolgáltatások

Szatmári Zoltán

Tóth Dániel



Előző és következő részek tartalmából

- Modellezés
- Szkriptelés
- Felhasználókezelés
 - Alapjai, hitelesítés (OPRE)
 - Engedélyezés (OPRE)
 - **Központosított felhasználókezelés, címtárak**

Tartalom

- **A felhasználókezelés nehézségei**
- Címtár szolgáltatások
 - LDAP
 - Active Directory

- Sok rendszer
- Sok felhasználó (minden rendszeren külön-külön)
- Egyszer csak kitör a káosz
 - Elburjánzó felhasználói fiókok
 - Szétszinkronizálódó jelszavak
 - Webes alkalmazásnak, VPN-nek is kéne beléptetés, teljesen más rendszert használnak...

Megoldások a káoszra

- Elburjánzó felhasználói fiókok
→ felhasználói életciklus kezelésére eljárásrend
- Sok rendszer igényel hitelesítést
→ **központosított felhasználói adattár**

Címtár (directory) szolgáltatás

■ Definíció:

- nyilvános adattár
- „intelligens” címjegyzék (phone directory)

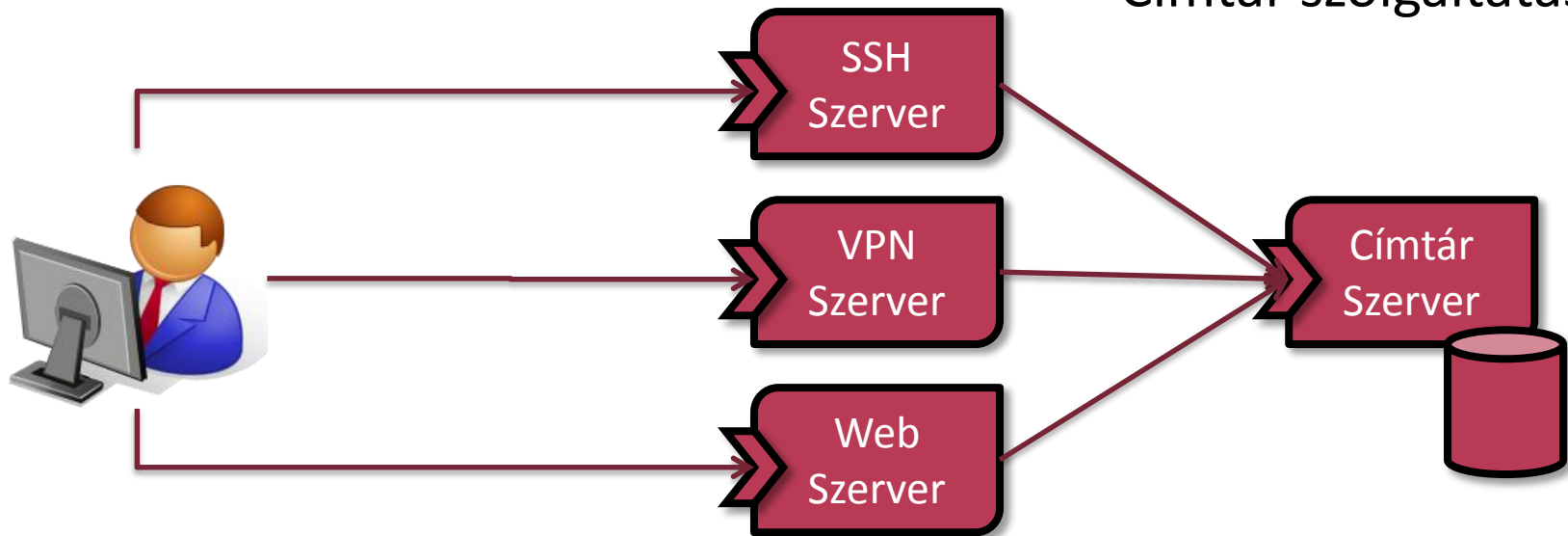
■ Tárolt adatok

- felhasználó adatai (e-mail címek, különböző fajta nevek, azonosítók, ...)
- számítógépek adatai
- biztonsági információk
- bármi egyéb

Címtár szolgáltatás hitelesítésre

Hogy fogja ez megoldani a hitelesítést?

Címtár szolgáltatás



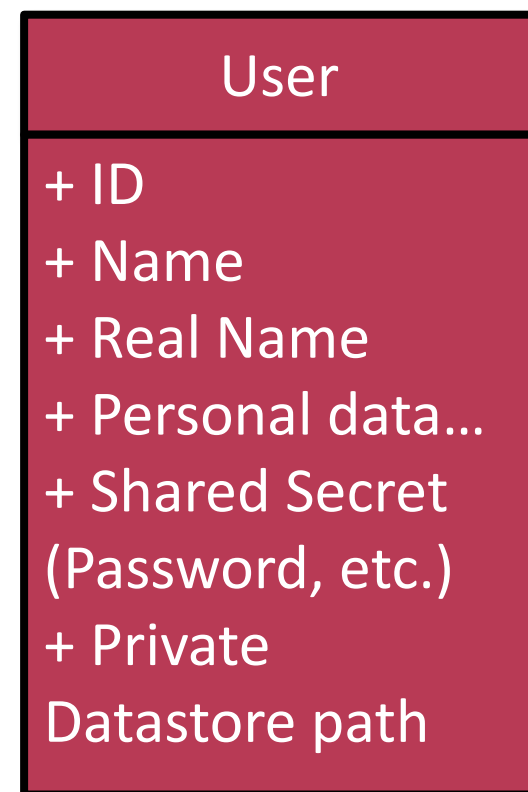
Beléptetés *minden esetben* a címtárban tárolt felhasználói adatok lekérdezésével történik.

Hogy néz ki egy címtár?

- Speciális adatbázis struktúra
 - szigorúan hierarchikus (általában objektum-orientált)

Hogy néz ki egy címtár?

- Speciális adatbázis struktúra
 - szigorúan hierarchikus (általában objektum-orientált)



Hogy néz ki egy címtár?

- Speciális adatbázis struktúra
 - szigorúan hierarchikus (általában objektum-orientált)
- Domináns műveletek:
 - keresés
 - olvasás
 - batch jellegű hozzáadás / módosítás

Címtárak fejlődéstörténete

- DNS (Domain Name Service)
- NIS (Network Information System)
 - volt Sun Yellow Pages (Sun Microsystems, 1988, SunOS 4.0)
- A korszerűbbek
 - X.500 / LDAP
 - Active Directory

Tartalom

- A felhasználókezelés nehézségei
- Címtár szolgáltatások
 - **LDAP**
 - **LDAP bevezetés**
 - LDAP felépítés
 - LDAP a gyakorlatban
 - Összefoglalás
 - Active Directory

Lightweight Directory Access Protocol (LDAP)

Kibocsátó: Internet Engineering Task Force (IETF)

Legutóbbi verzió: LDAPv3 – RFC 4510, 2006

Cél: elosztott címtárszolgáltatások megvalósítása, elérése

X.500

- ISO/OSI X.500 egy szabványcsalád
 - Eredetileg X.400-as levelezés támogatására

X.500

- ISO/OSI X.500 egy szabványcsalád
 - Eredetileg X.400-as levelezés támogatására
- Alapfogalmak: X.500
 - Modellek: X.501
 - Hitelesítés: X.509 (Tovább él az SSL certificate-ekben)
 - Attribútumok: X.520
 - Osztályok: X.521
 - Elérési protokoll: X.519

X.500

- ISO/OSI X.500 egy szabványcsalád
 - Eredetileg X.400-as levelezés támogatására
- Alapfogalmak: X.500
 - Modellek: X.501
 - Hitelesítés: X.509 (Tovább él az SSL certificate-ekben)
 - Attribútumok: X.520
 - Osztályok: X.521
 - Elérési protokoll: X.519
- Ennek része a DAP (Directory Access Protocol)
 - Az ISO/OSI hálózati szolgáltatásokra épül → TCP/IP-re nem jó!
 - Az IETF kézbe vette a dolgot → Ebből lett az LDAP

LDAP

- **LDAP: Lightweight Directory Access Protocol**

LDAP

- **LDAP: Lightweight Directory Access Protocol**
- **L**, mint pehelysúlyú: az X.500 kódnevű protokollcsalád könnyített változata.

LDAP

- **LDAP: Lightweight Directory Access Protocol**
- **L**, mint pehelysúlyú: az X.500 kódnevű protokollcsalád könnyített változata.
- **D**, mint címtárszolgáltatás: elsősorban egy számítógépes hálózat felhasználóit és erőforrásait tartalmazó adatbázis közvetítésére szolgál

LDAP

- **LDAP: Lightweight Directory Access Protocol**
- **L**, mint pehelysúlyú: az X.500 kódnevű protokollcsalád könnyített változata.
- **D**, mint címtárszolgáltatás: elsősorban egy számítógépes hálózat felhasználóit és erőforrásait tartalmazó adatbázis közvetítésére szolgál
- **A**, mint elérés: támogatja az adatok frissítését, törlését, beszúrását és lekérdezését

LDAP

- **LDAP: Lightweight Directory Access Protocol**
- **L**, mint pehelysúlyú: az X.500 kódnevű protokollcsalád könnyített változata.
- **D**, mint címtárszolgáltatás: elsősorban egy számítógépes hálózat felhasználóit és erőforrásait tartalmazó adatbázis közvetítésére szolgál
- **A**, mint elérés: támogatja az adatok frissítését, törlését, beszúrását és lekérdezését
- **P**, mint az elektronikus kommunikáció egyik nyelve: egy TCP/IP felett megvalósított bináris protokoll

Alaptulajdonságok és fogalmak

- Csomópontok, bejegyzések (**entry**)
 - Objektum-orientált szemlélet
- Hierarchikus felépítés (**directory tree**)
- Kitüntetett attribútum (**relative distinguished name - rdn**)
- Megkülönböztető név (**distinguished name - dn**)
- Többértékű attribútumok
 - Kapcsolatok megvalósítása

Alaptulajdonságok és fogalmak

- Csomópontok, bejegyzések (**entry**)
 - Objektum-orientált szemlélet
- Hierarchikus felépítés (**directory tree**)
- Kitüntetett attribútum (**relative distinguished name - rdn**)
- Megkülönböztető név (**distinguished name - dn**)
- Többértékű attribútumok
 - Kapcsolatok megvalósítása

Az LDAP az UML előtt alakult ki, ezért picit más fogalomkészlettel dolgozik, mint amit máshol megszoktunk

- Készítsük el az egyetemünk LDAP adatbázisát!
 - Csomópontok
 - Objektum-orientált szemlélet
 - Hierarchia
 - Kitüntetett attribútum
 - Megkülönböztető név
 - Többértékű attribútum

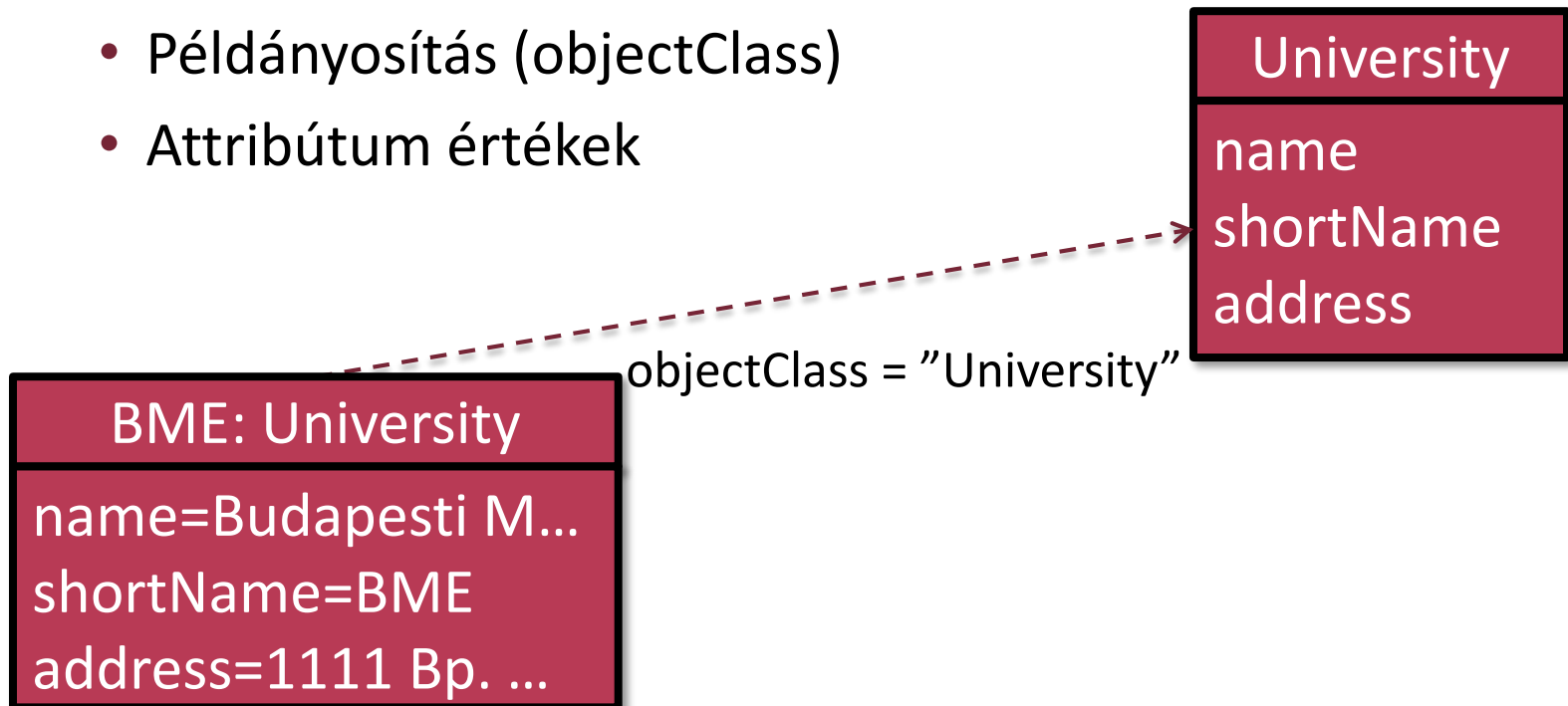
Séma alkalmazása

- Az alapvető modellezési alapfogalmak jelennek meg
 - Séma (metamodell szint)
 - Attribútumok



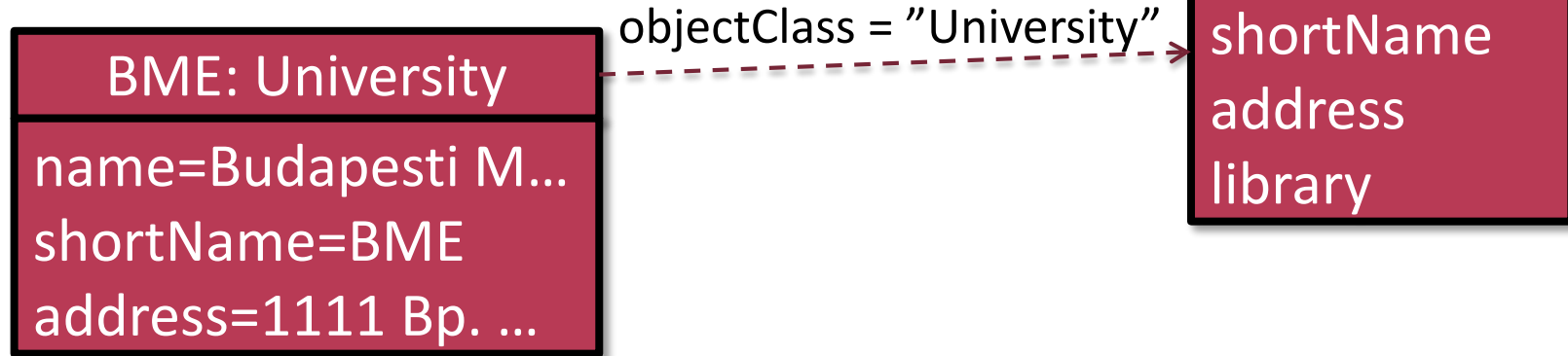
Séma alkalmazása

- Az alapvető modellezési alapfogalmak jelennek meg
 - Séma (metamodell szint)
 - Attribútumok
 - Egyed (példánymodell szint)
 - Példányosítás (objectClass)
 - Attribútum értékek



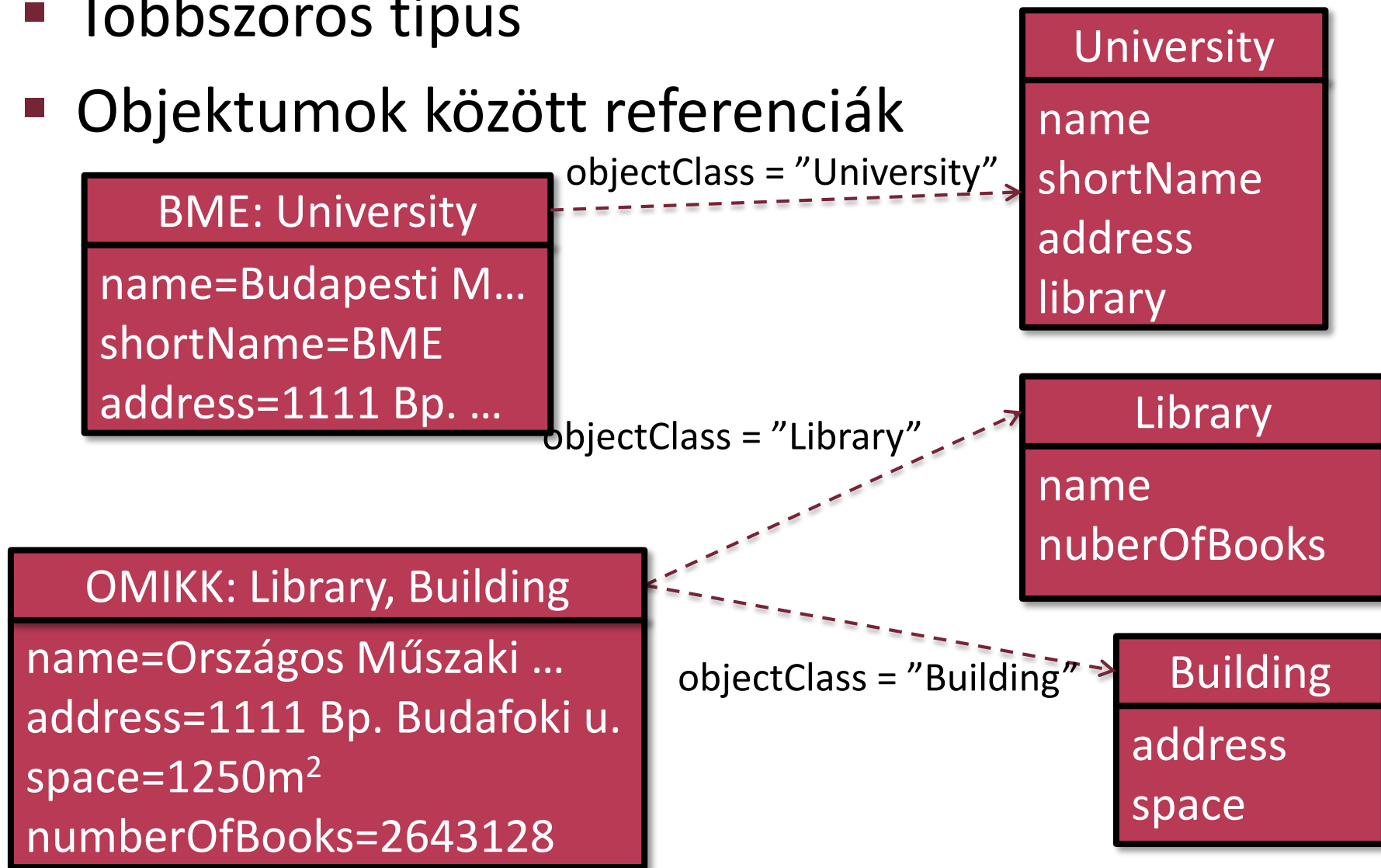
Objektum-orientált szemlélet

- Többszörös típus
- Objektumok között referenciák



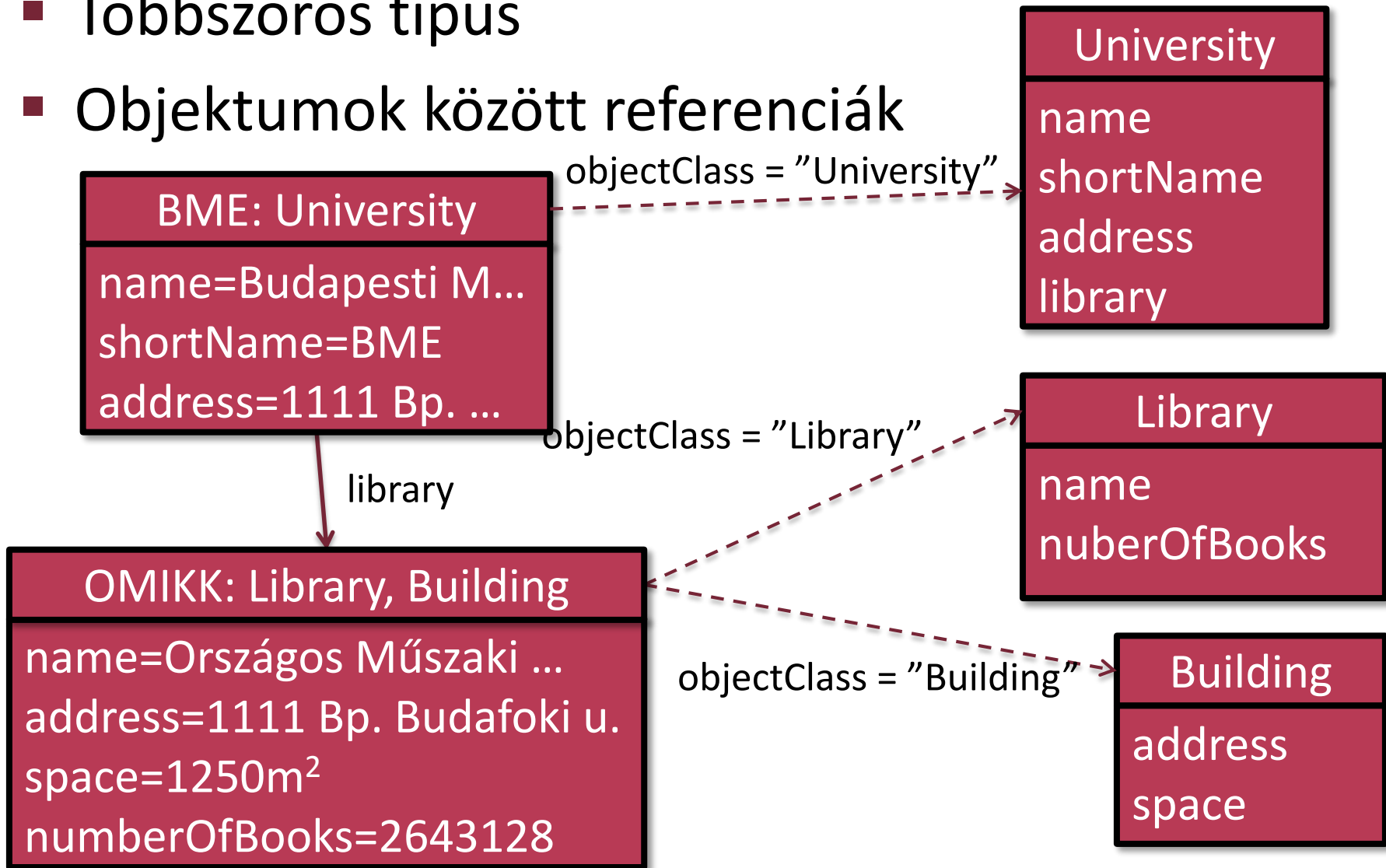
Objektum-orientált szemlélet

- Többszörös típus
- Objektumok között referenciák



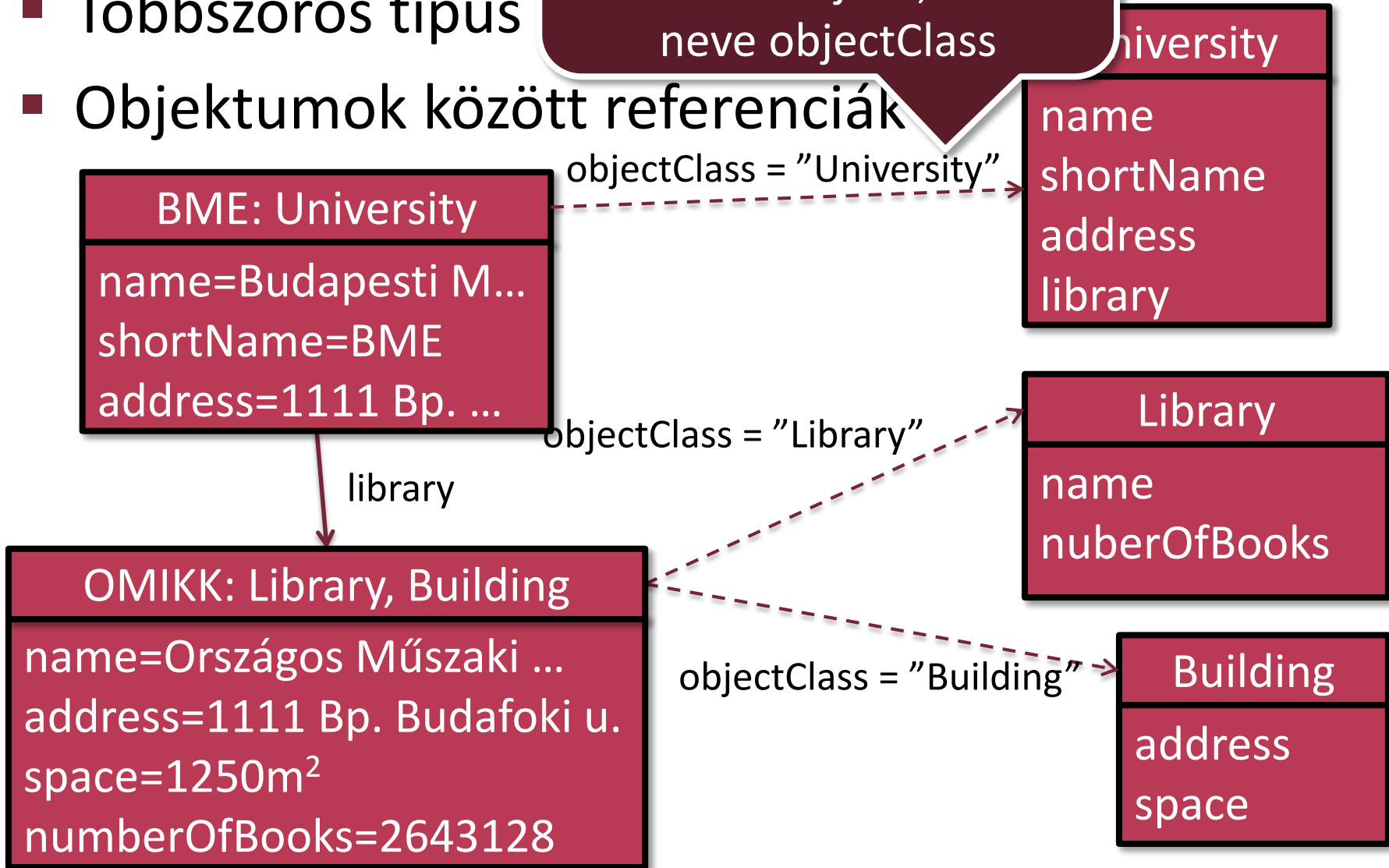
Objektum-orientált szemlélet

- Többszörös típus
- Objektumok között referenciák



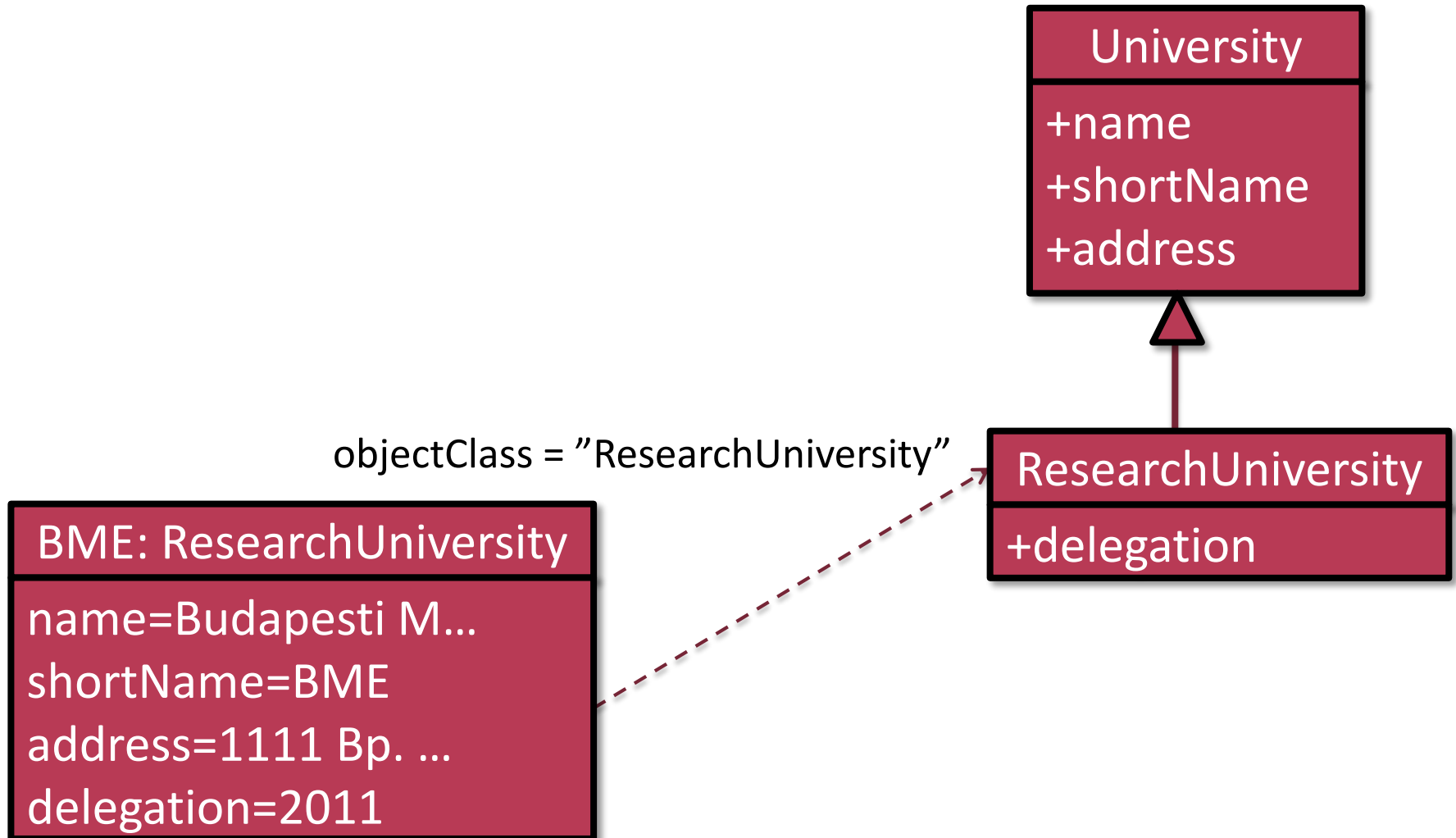
A típus-példány kapcsolatát is egy referencia írja le, ennek neve objectClass

- Többszörös típus
- Objektumok között referenciák



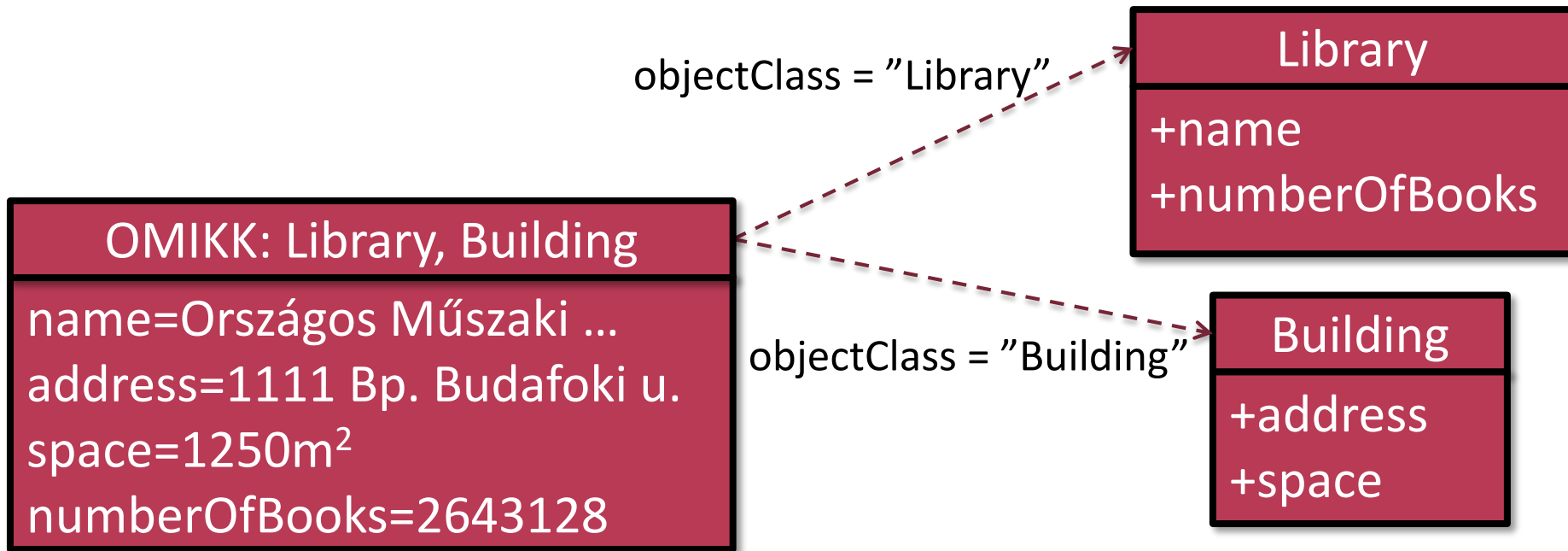
Objektum-orientált szemlélet

- Öröklődnek az attribútumok, referenciák



Objektum-orientált szemlélet

- Egy objektumnak több típusa is lehet, ilyenkor az osztályokban definiált attribútumok uniója szerepel az objektumban.



Hierarchikus felépítés

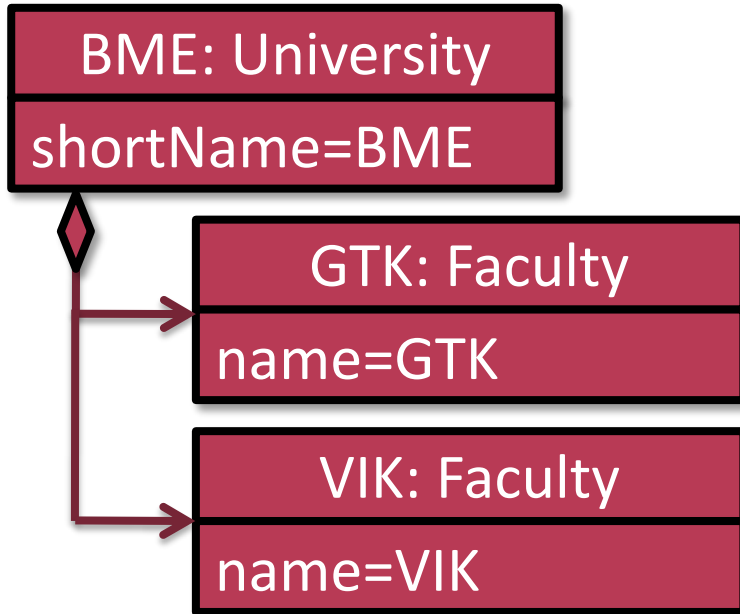
- A csomópontok tartalmazási hierarchiát alkotnak

BME: University

shortName=BME

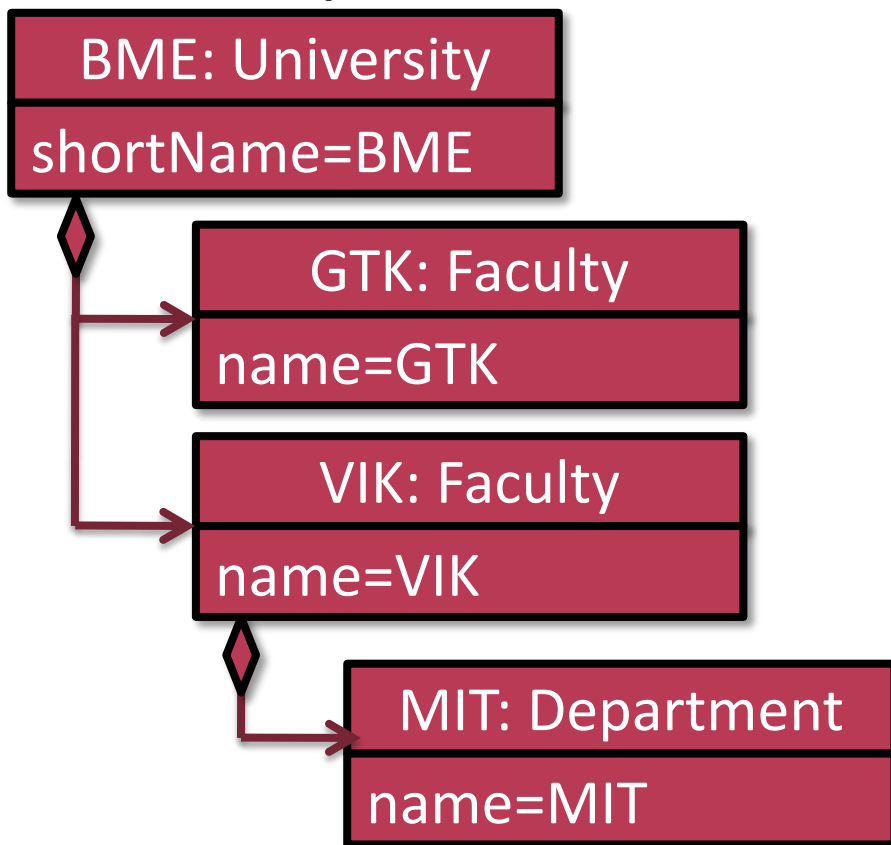
Hierarchikus felépítés

- A csomópontok tartalmazási hierarchiát alkotnak



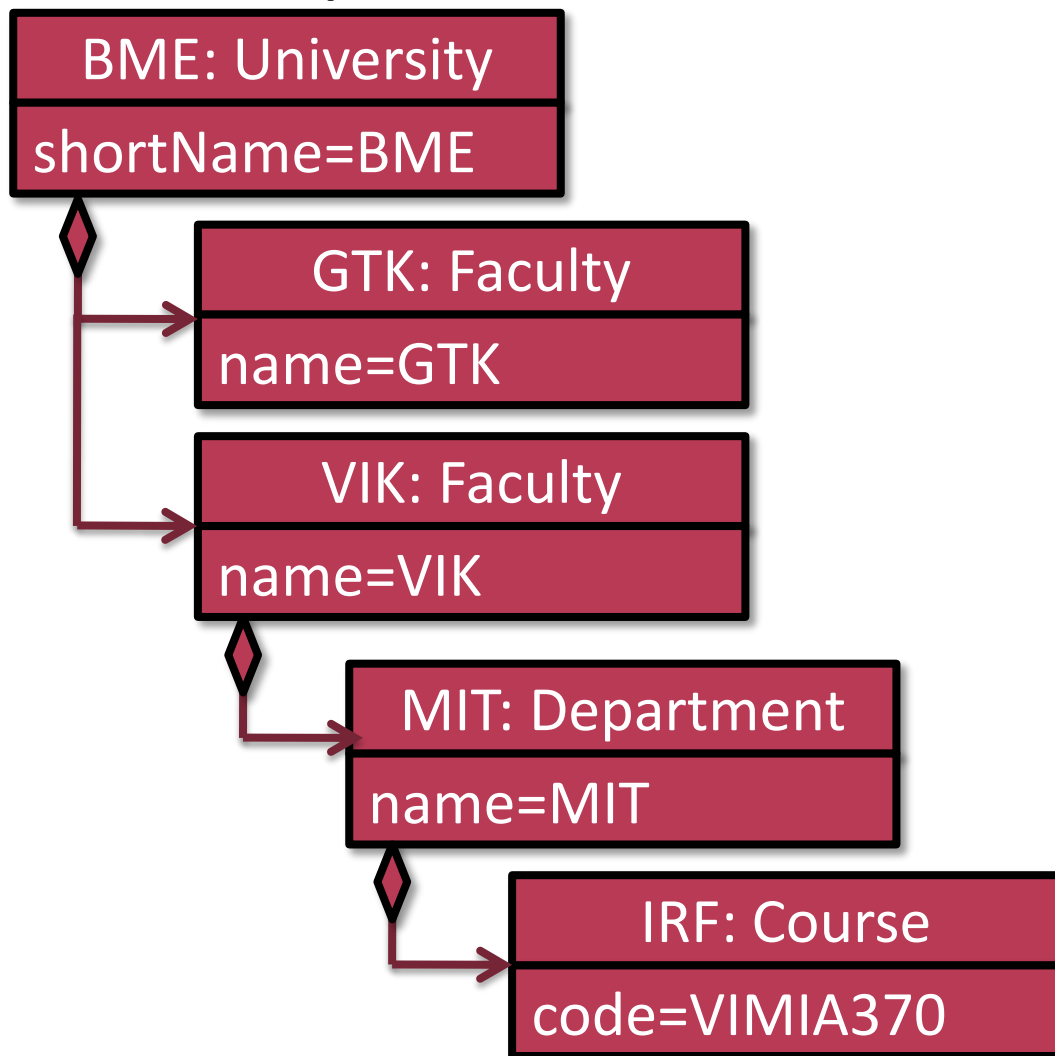
Hierarchikus felépítés

- A csomópontok tartalmazási hierarchiát alkotnak



Hierarchikus felépítés

- A csomópontok tartalmazási hierarchiát alkotnak

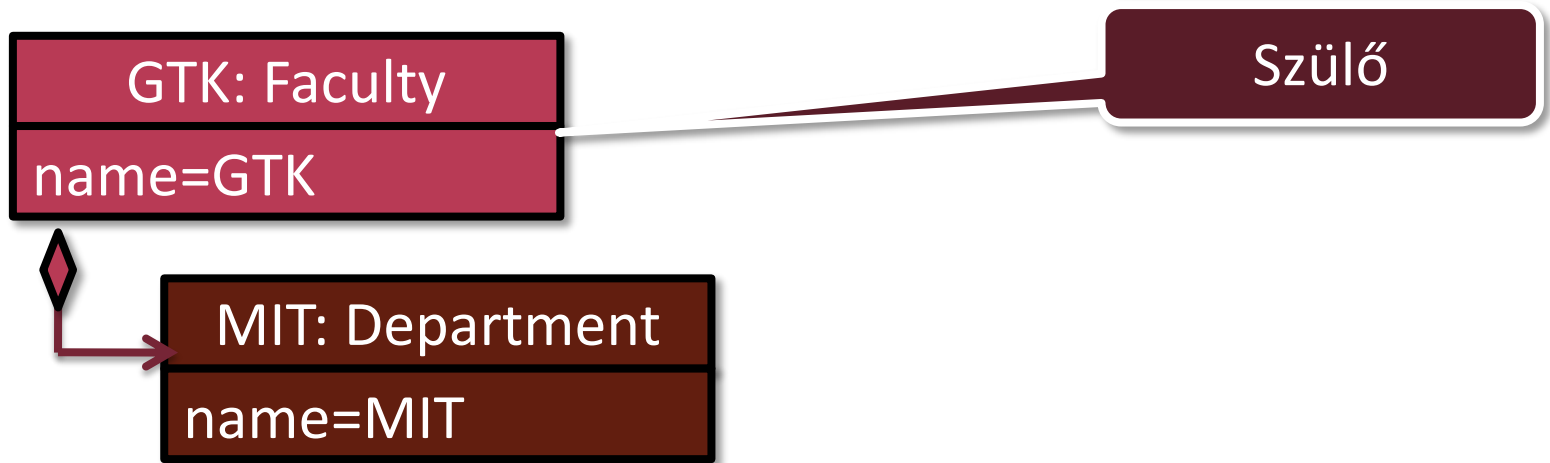


Csomópontok közötti viszonyok

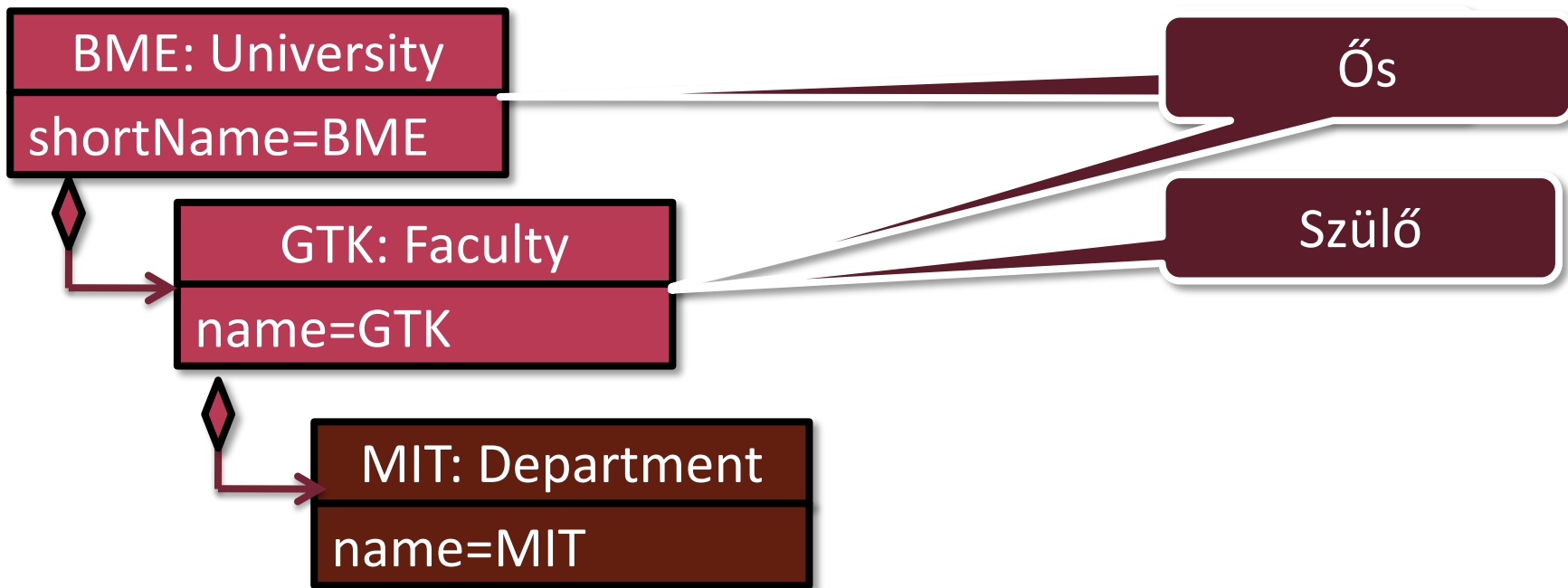
MIT: Department

name=MIT

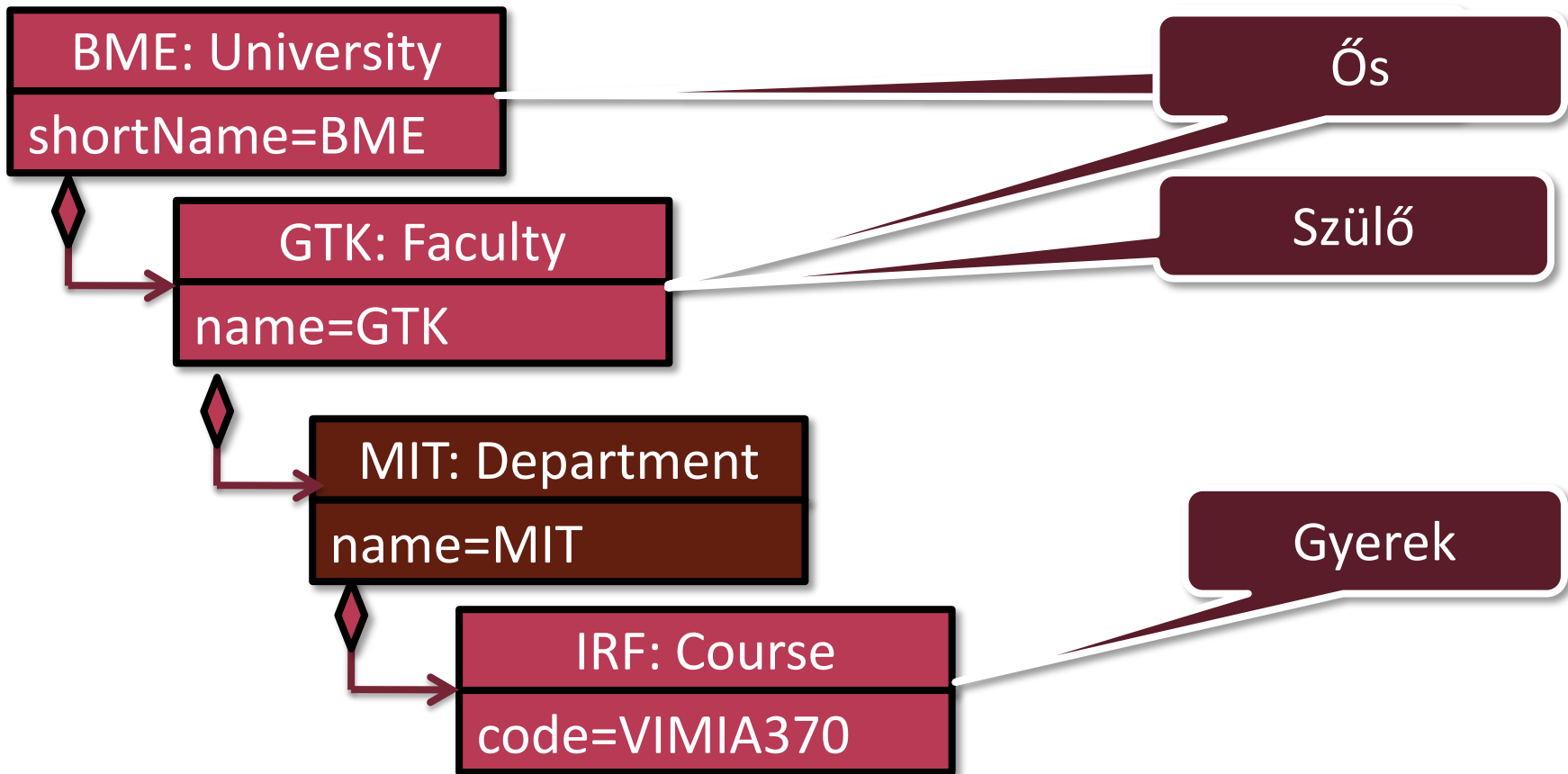
Csomópontok közötti viszonyok



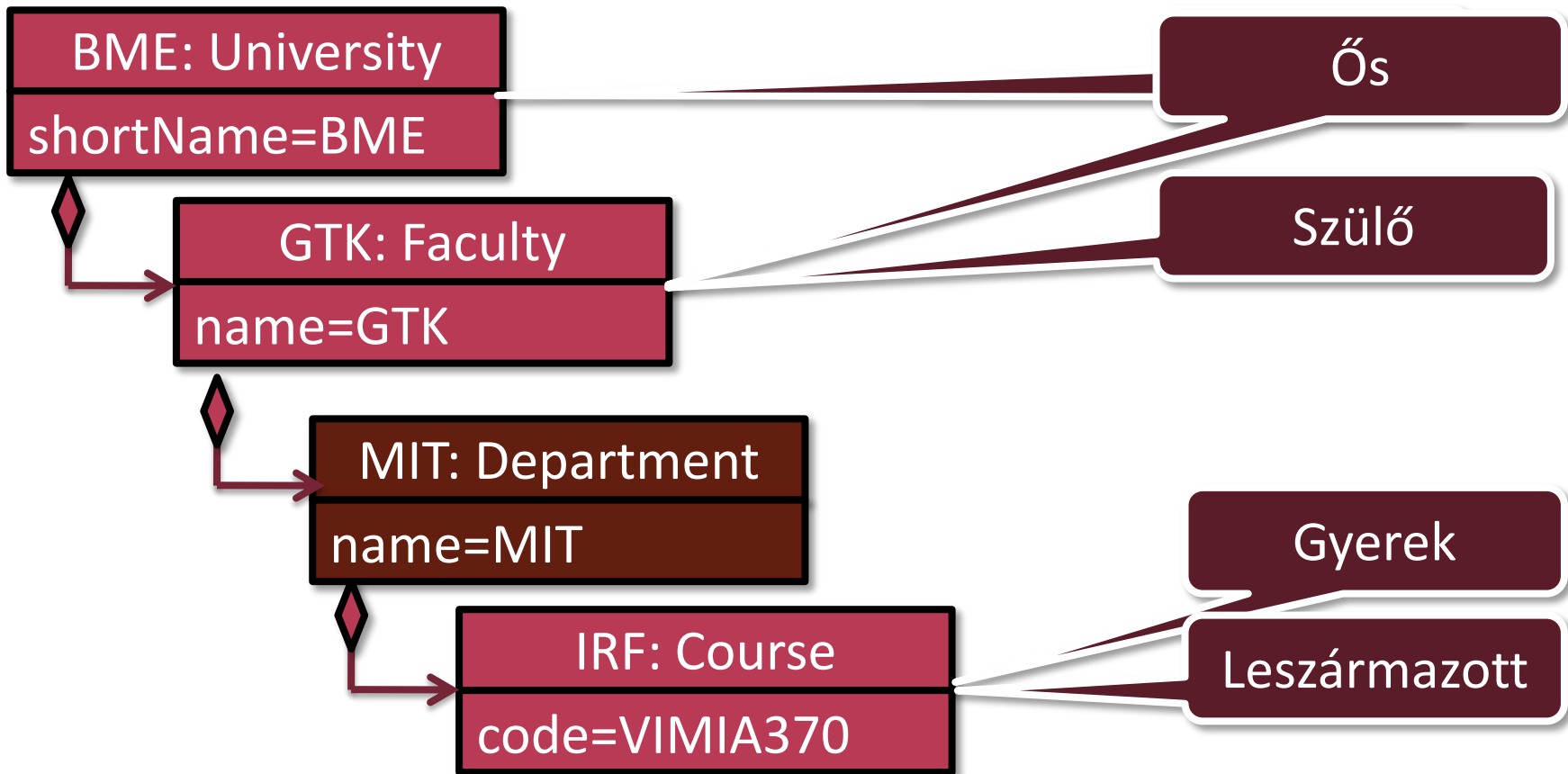
Csomópontok közötti viszonyok



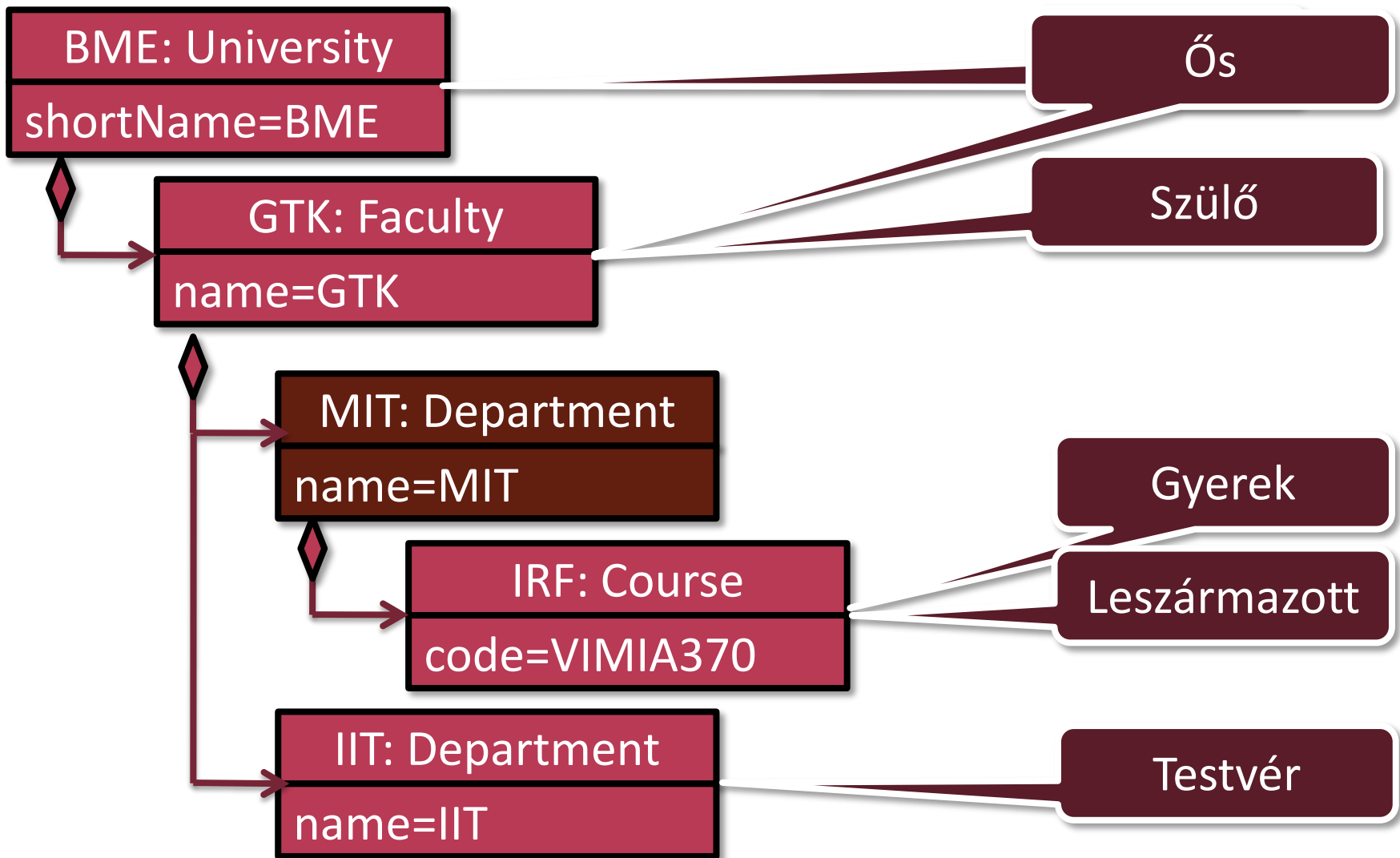
Csomópontok közötti viszonyok



Csomópontok közötti viszonyok



Csomópontok közötti viszonyok

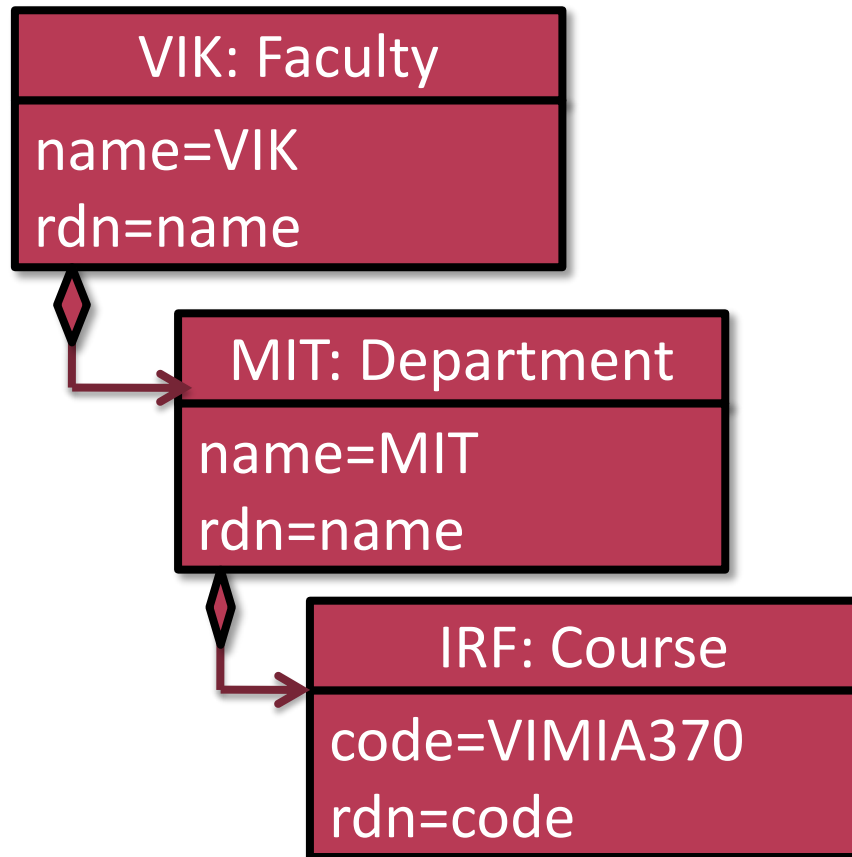


Kitüntetett attribútum

- RDN (relative distinguished name)
 - Megmutatja, hogy melyik attribútumot akarjuk egyedi névként használni

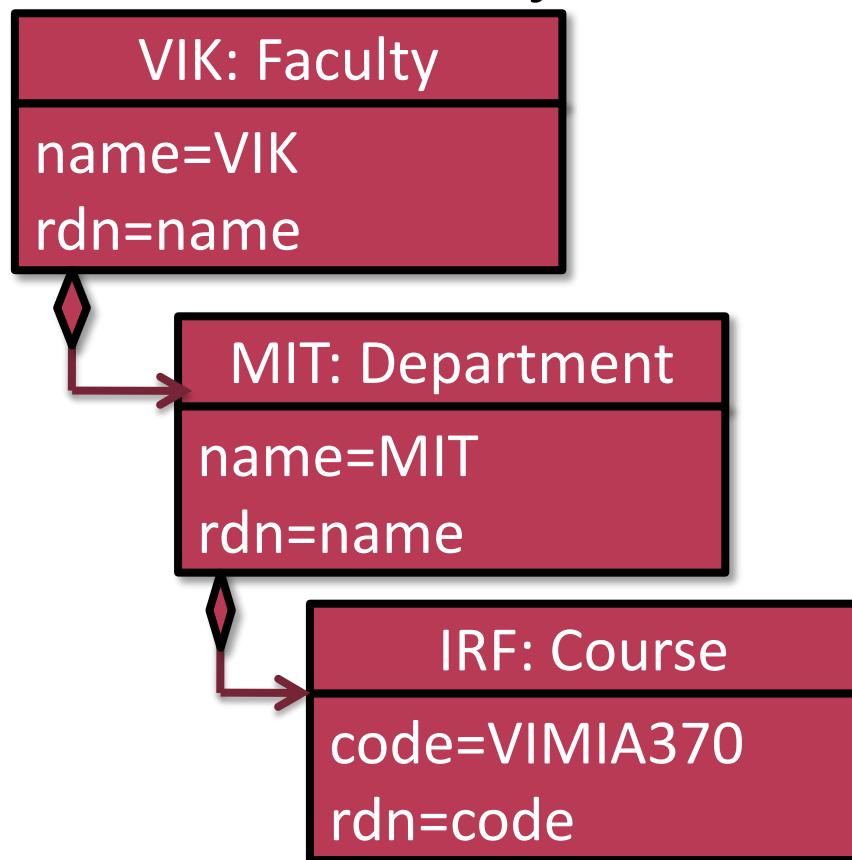
Kitüntetett attribútum

- RDN (relative distinguished name)
 - Megmutatja, hogy melyik attribútumot akarjuk egyedi névként használni



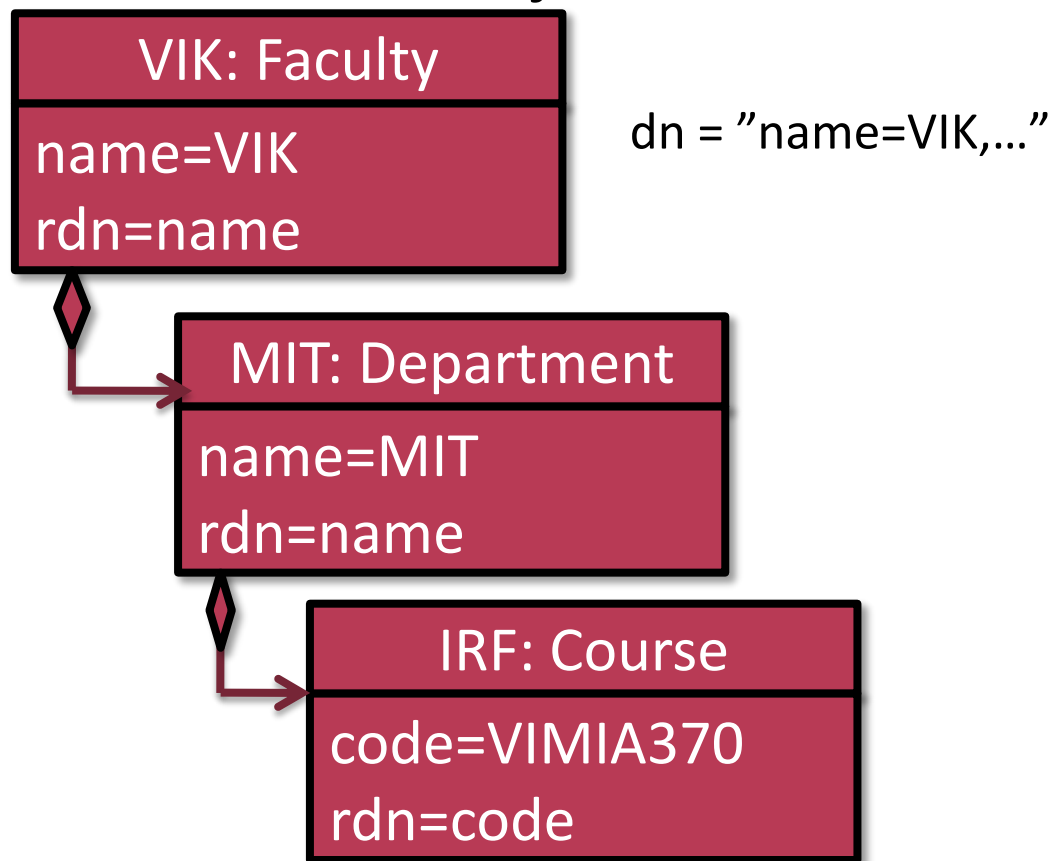
Megkülönböztető név

- DN (distinguished name)
 - A tartalmazások mentén egyedileg azonosítható minden objektum a szülők RDN listájával.



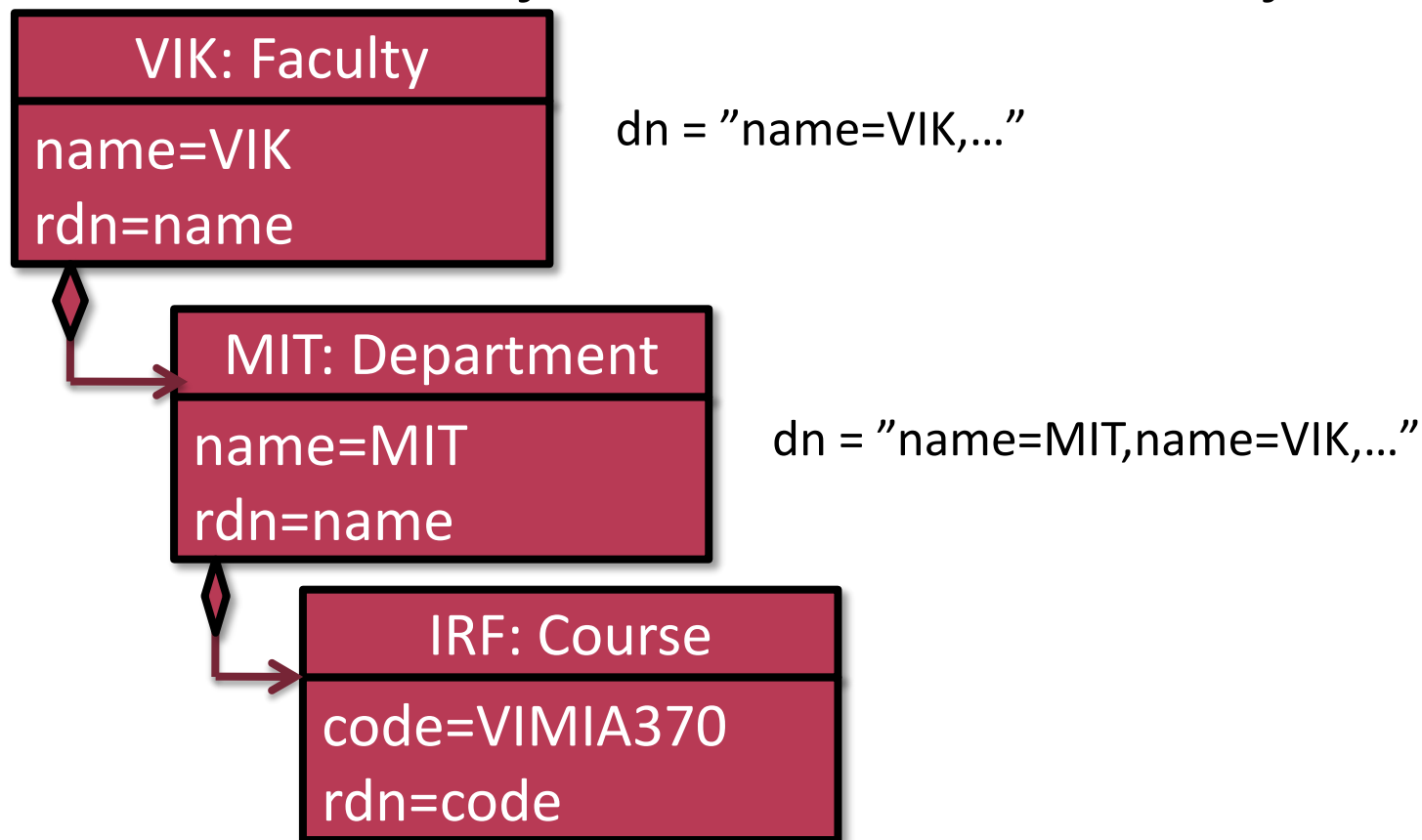
Megkülönböztető név

- DN (distinguished name)
 - A tartalmazások mentén egyedileg azonosítható minden objektum a szülők RDN listájával.



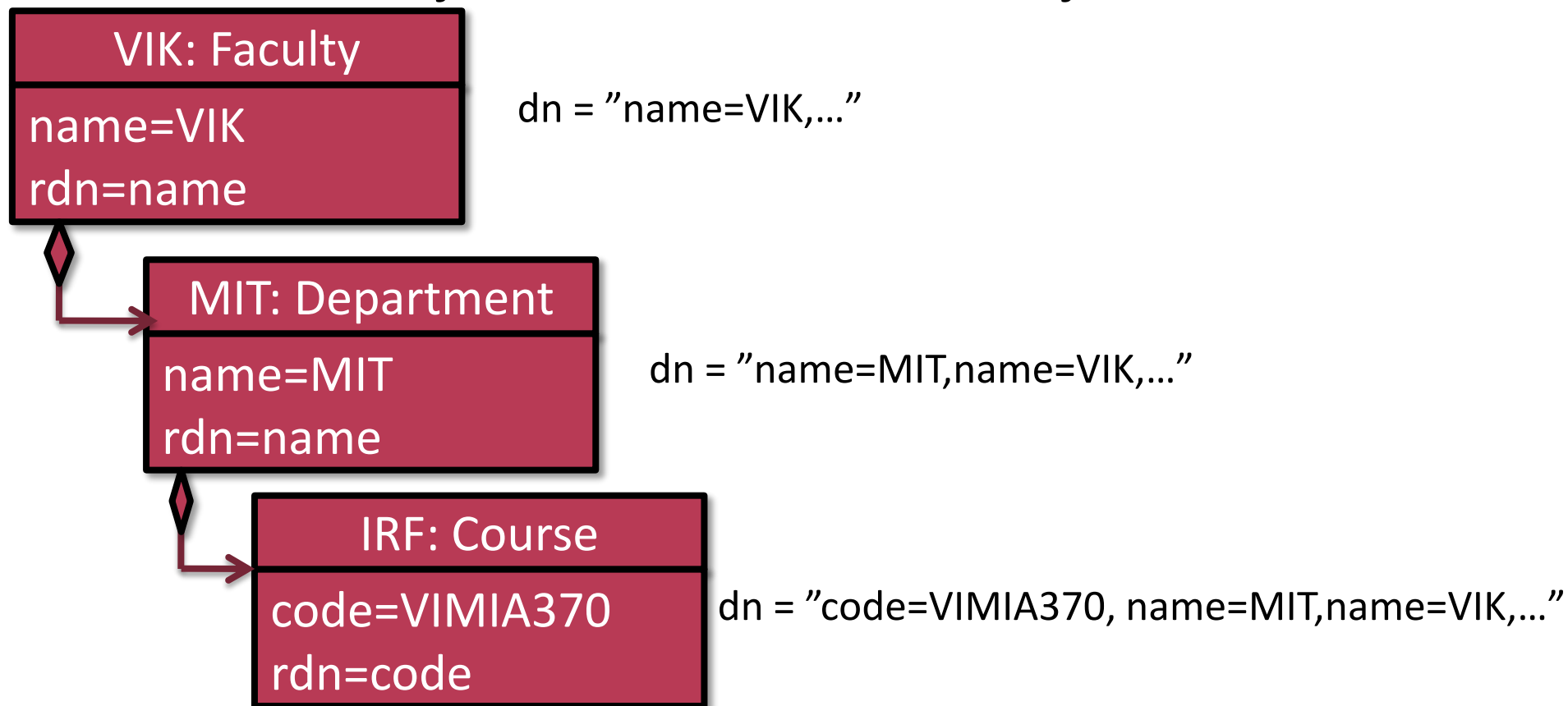
Megkülönböztető név

- DN (distinguished name)
 - A tartalmazások mentén egyedileg azonosítható minden objektum a szülők RDN listájával.



Megkülönböztető név

- DN (distinguished name)
 - A tartalmazások mentén egyedileg azonosítható minden objektum a szülők RDN listájával.

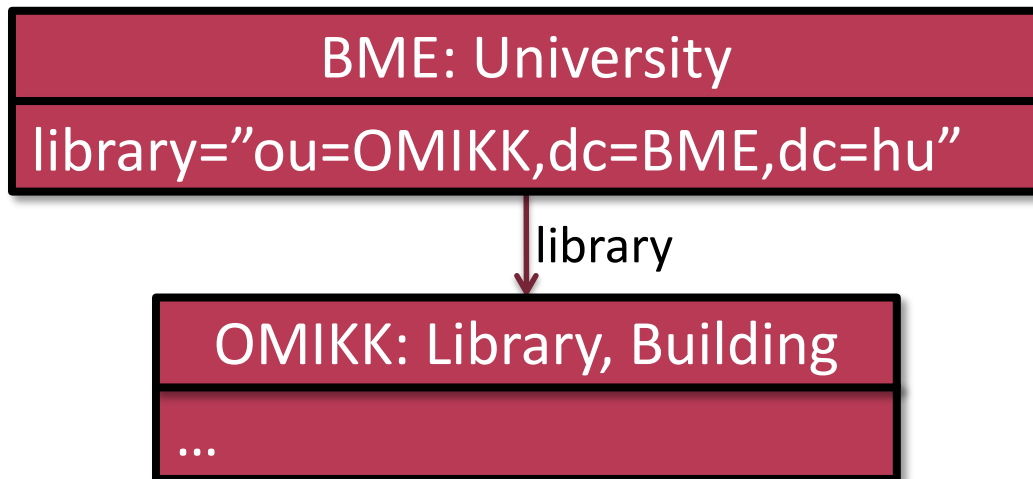


Megkülönböztető név

- Kitüntetett gyökér elem
 - Jellemzően valamilyen domain-ből származik
 - Pl.: "dc=bme,dc=hu"
- A DN felépítéséből adódóan egyedi azonosítást tesz lehetővé

Megkülönböztető név

- Kitüntetett gyökér elem
 - Jellemzően valamilyen domain-ből származik
 - Pl.: "dc=bme,dc=hu"
- A DN felépítéséből adódóan egyedi azonosítást tesz lehetővé
 - Referenciák ez alapján hivatkoznak a célpontra



Megkülönböztető név

- Kitüntetett gyökér elem
 - Jellemzően valamilyen domain-ből származik
 - Pl.: "dc=bme,dc=hu"
- A DN felépítéséből adódóan egyedi azonosítást tesz lehetővé
 - Referenciák ez alapján hivatkoznak a célpontra



dn = "dc=BME,dc=hu"

Megkülönböztető név

- Kitüntetett gyökér elem
 - Jellemzően valamilyen domain-ből származik
 - Pl.: "dc=bme,dc=hu"
- A DN felépítéséből adódóan egyedi azonosítást tesz lehetővé
 - Referenciák ez alapján hivatkoznak a célpontra



dn = "dc=BME,dc=hu"

OMIKK: Library, Building

dn = "ou=OMIKK,dc=BME,dc=hu"

Többértékű attribútumok

■ Attribútumok felvehetnek

○ Egy értéket

- Pl.: kód

○ Több értéket (lista)

- Pl.: hallgató

dn = "code=VIMIA370,ou=MIT,ou=VIK,dc=BME,dc=hu"

IRF: Course

code=VIMIA370

rdn=code

student="nk=ABCDEF,year=2010,dc=bme,dc=hu"

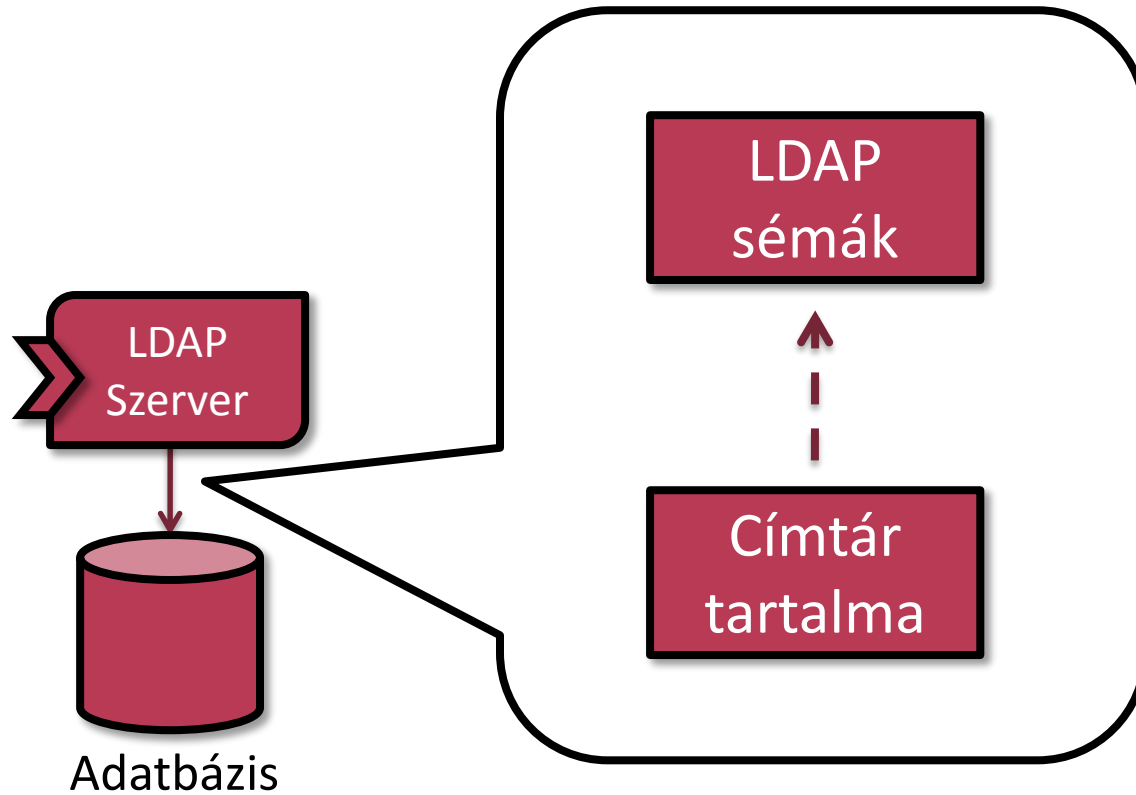
student="nk=GHIJKL,year=2011,dc=bme,dc=hu"

...

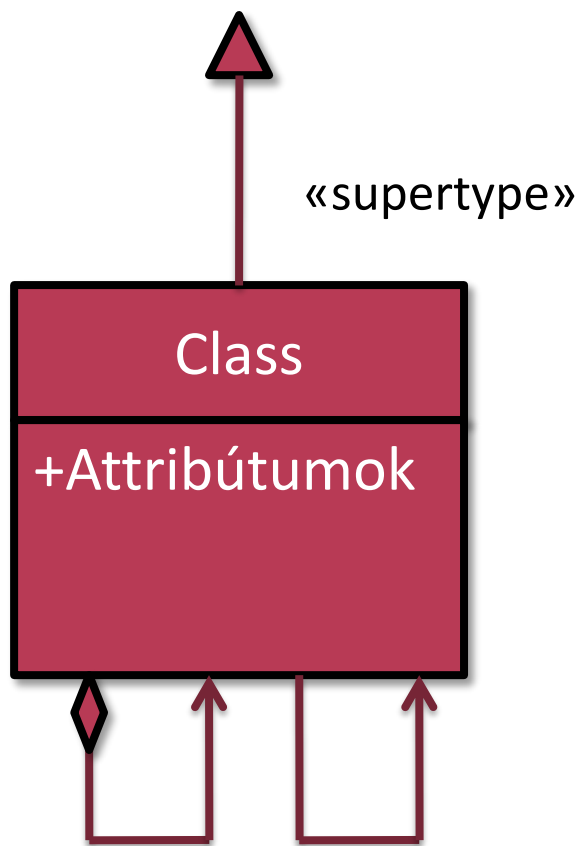
Tartalom

- LDAP
 - LDAP bevezetés
 - **LDAP felépítés**
 - LDAP a gyakorlatban
 - Összefoglalás

LDAP felépítése



LDAP séma

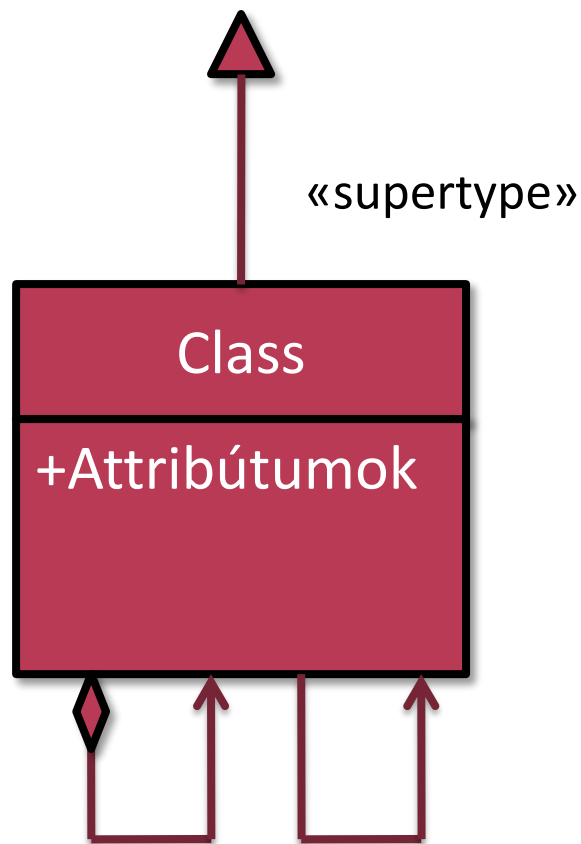


Tartalmazott Referenciák
elemek

LDAP séma

- Statikus

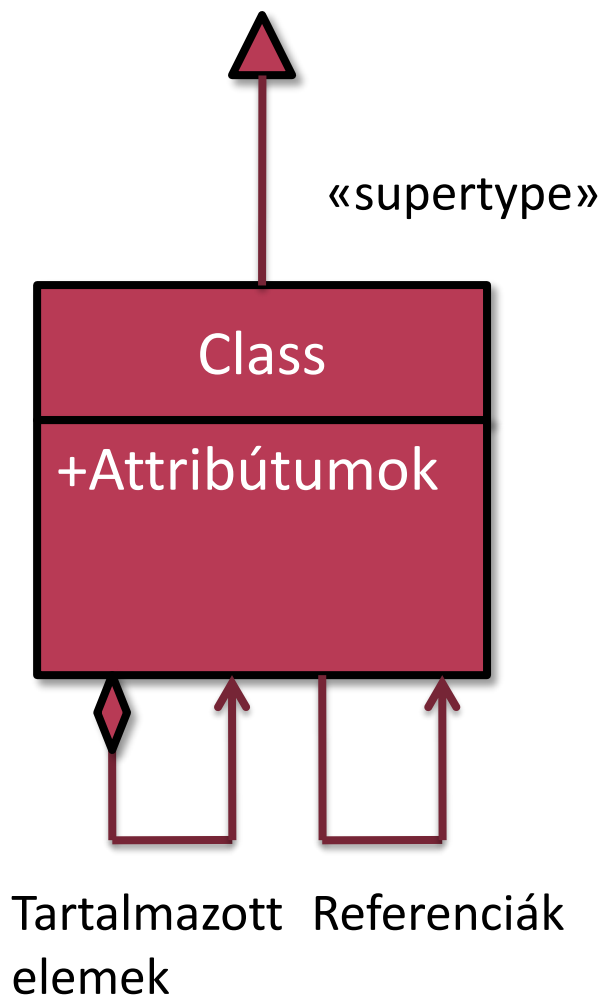
- Működés közben nem változik
- Konfigurációs fájlokban adják meg (ASN.1 formátumban)



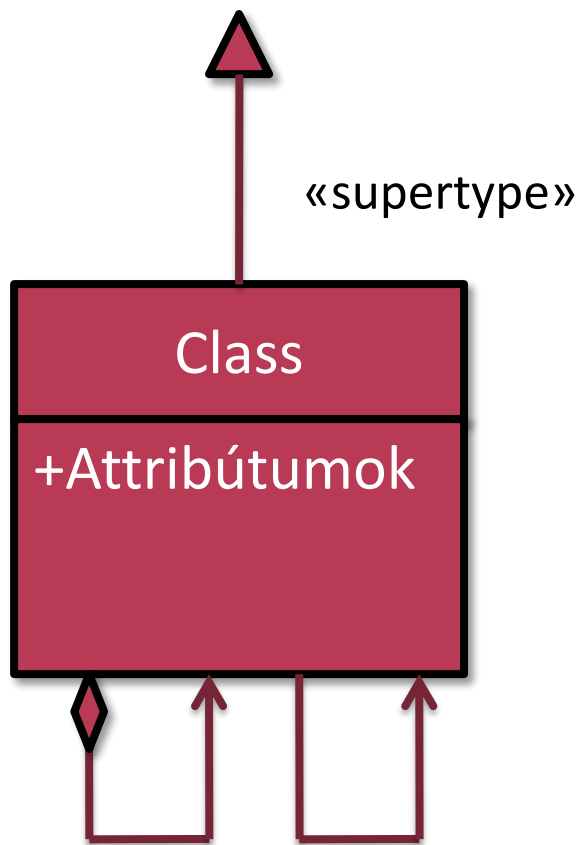
Tartalmazott Referenciák
elemek

LDAP séma

- Statikus
 - Működés közben nem változik
 - Konfigurációs fájlokban adják meg (ASN.1 formátumban)
- Szabványos
 - Van számos többé-kevésbe de facto szabvány séma
 - Pl. core, cosine (X.500), java, nis, inetorgperson



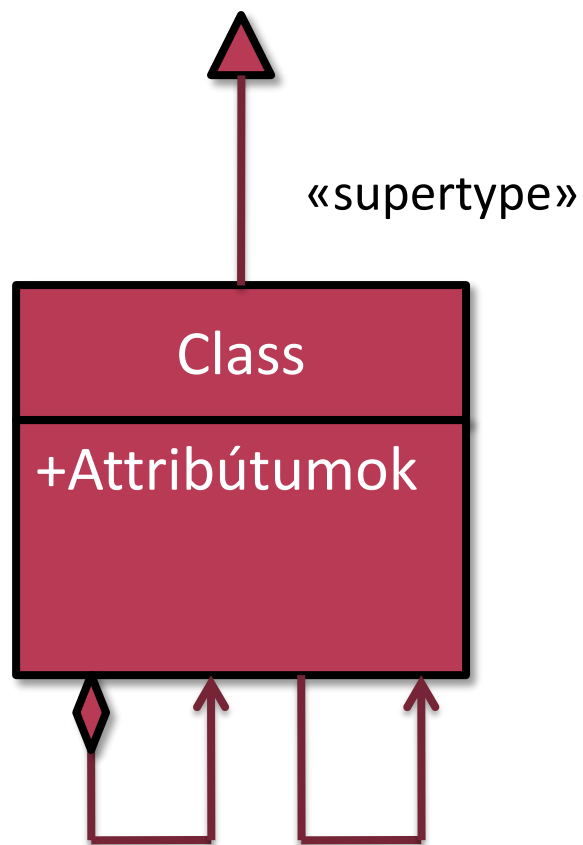
LDAP séma



Tartalmazott
elemek

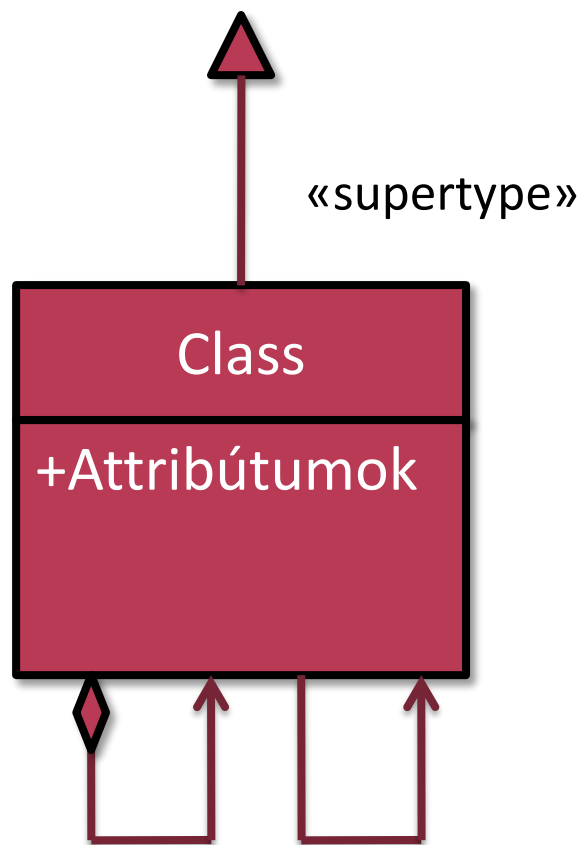
Referenciák

LDAP séma



- Minden elemnek van egy azonosítója (OID)
 - osztálynak és attribútumnak is
 - Pl.: inetOrgPerson
2.16.840.1.113730.3.2.2
 - álnevek használata
 - Pl.: uid és userid

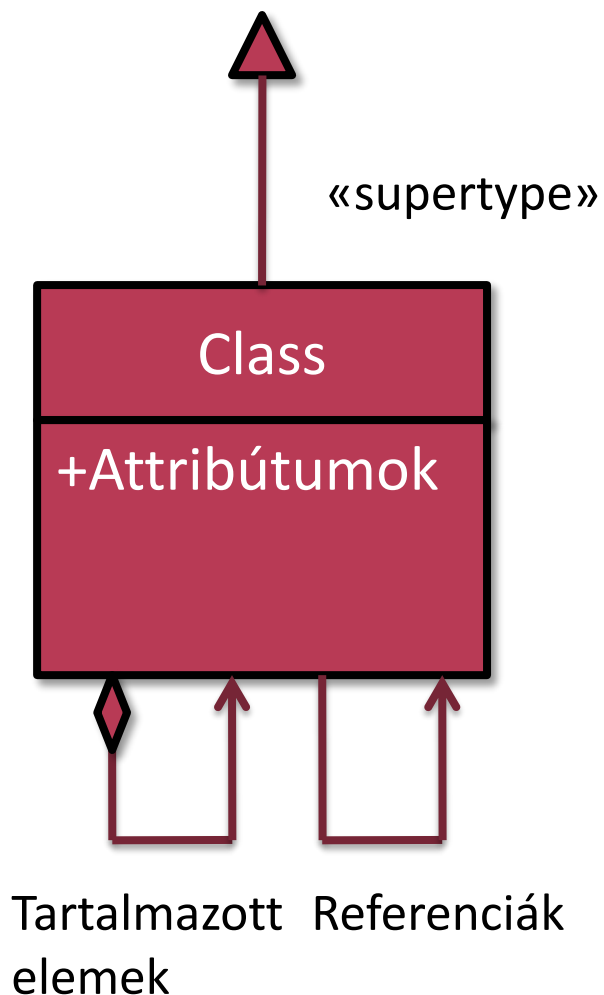
LDAP séma



- Minden elemnek van egy azonosítója (OID)
 - osztálynak és attribútumnak is
 - Pl.: inetOrgPerson
2.16.840.1.113730.3.2.2
 - álnevek használata
 - Pl.: uid és userid
- Van öröklés az osztályok között

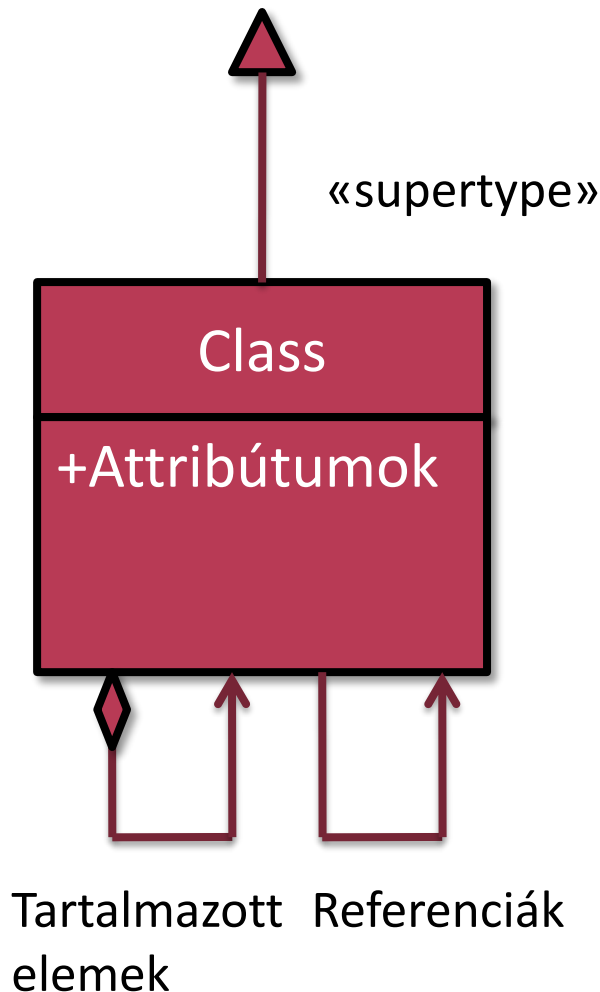
Tartalmazott Referenciák
elemek

LDAP séma



- Minden elemnek van egy azonosítója (OID)
 - osztálynak és attribútumnak is
 - Pl.: inetOrgPerson
2.16.840.1.113730.3.2.2
 - álnevek használata
 - Pl.: uid és userId
- Van öröklés az osztályok között
- Attribútumok
 - lehetnek kötelezőek, opcionálisak,
 - van multiplicitásuk is (lista)

LDAP séma



- Minden elemnek van egy azonosítója (OID)
 - osztálynak és attribútumnak is
 - Pl.: inetOrgPerson
2.16.840.1.113730.3.2.2
 - álnevek használata
 - Pl.: uid és userid
- Van öröklés az osztályok között
- Attribútumok
 - lehetnek kötelezőek, opcionálisak,
 - van multiplicitásuk is (lista)
- A referenciák valójában string attribútumok

LDAP séma

■ Osztályok típusai

○ Absztrakt

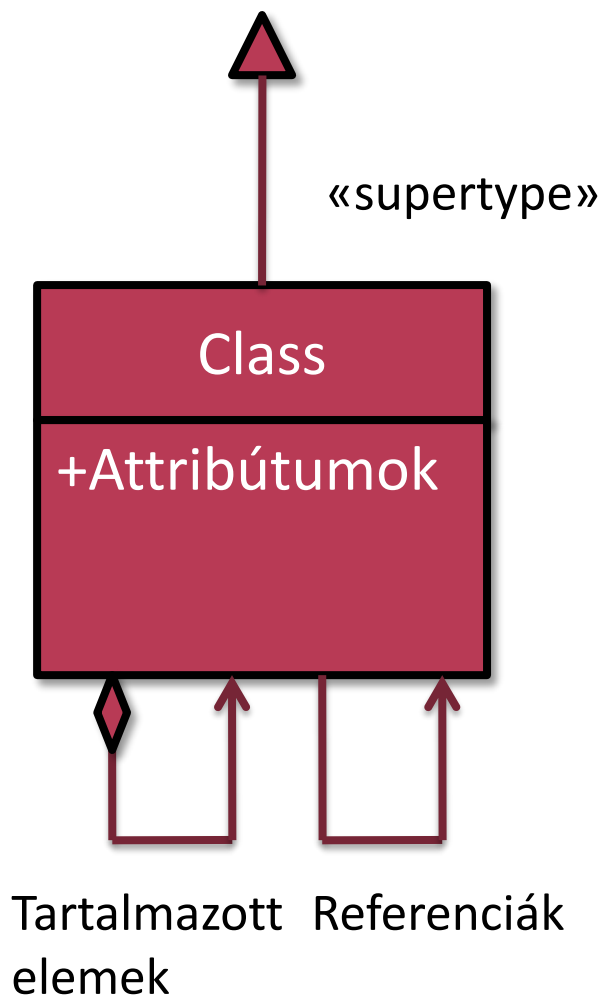
- Alapvető struktúra kialakítása
- A felhasználó számára nincs releváns információja.
- Pl.: top

○ Strukturális

- Alapvető tulajdonságokat ad meg
- Egymást kizáró osztályok
- Pl.: person és group

○ Kiegészítő

- Egyes sémák kiegészítésére
- Pl.: inetOrgPerson, PosixAccount



Példa osztály: Person

```
objectclass ( 2.5.6.6 NAME 'person'  
    DESC 'RFC2256: a person'  
    SUP top STRUCTURAL  
    MUST ( sn $ cn )  
    MAY ( userPassword $  
        telephoneNumber $  
        seeAlso $  
        description )  
)
```

Megvalósítások



- IBM Tivoli Directory Server, IBM DB2 backend adatbázissal
- OpenLDAP (open source) Pl. BerkleyDB 4.2 backend adatbázissal (lehet más is)
- Oracle Internet Directory
- Sun Java System Directory Server JDBC alapú adatbázisokkal
- Linux, UNIX (Pl. AIX), VMware ESX server, stb. PAM (Pluggable Authentication Modules) használatával
- Hálózati beléptetés (Pl. VPN, WLAN esetén)
- Webalkalmazások: Apache, PHP, Tomcat stb.
- Adatbáziskezelők: MySQL, PostgreSQL stb.

Tartalom

- LDAP
 - LDAP bevezetés
 - LDAP felépítés
 - **LDAP a gyakorlatban**
 - Összefoglalás

- OpenLDAP szerver
- Apache Directory Studio kliens
- Szervezeti egységekbe csoportosítás
- Felhasználók csoportokba rendelése
- Attribútumok

Szöveges LDAP transzfer formátum

LDIF (LDAP data interchange format):

```
dn: uid=don,dc=thefamily,dc=local
cn: Don Corleone
givenName: Don
sn: Corleone
uid: don
telephoneNumber: +1 888 555 6789
mail: don@thefamily.local
sons: cn=michael,dc=thefamily,dc=local
sons: cn=santino,dc=thefamily,dc=local
sons: cn=fredo,dc=thefamily,dc=local
objectClass: inetOrgPerson
objectClass: mafiaPerson
objectClass: person
objectClass: top
```

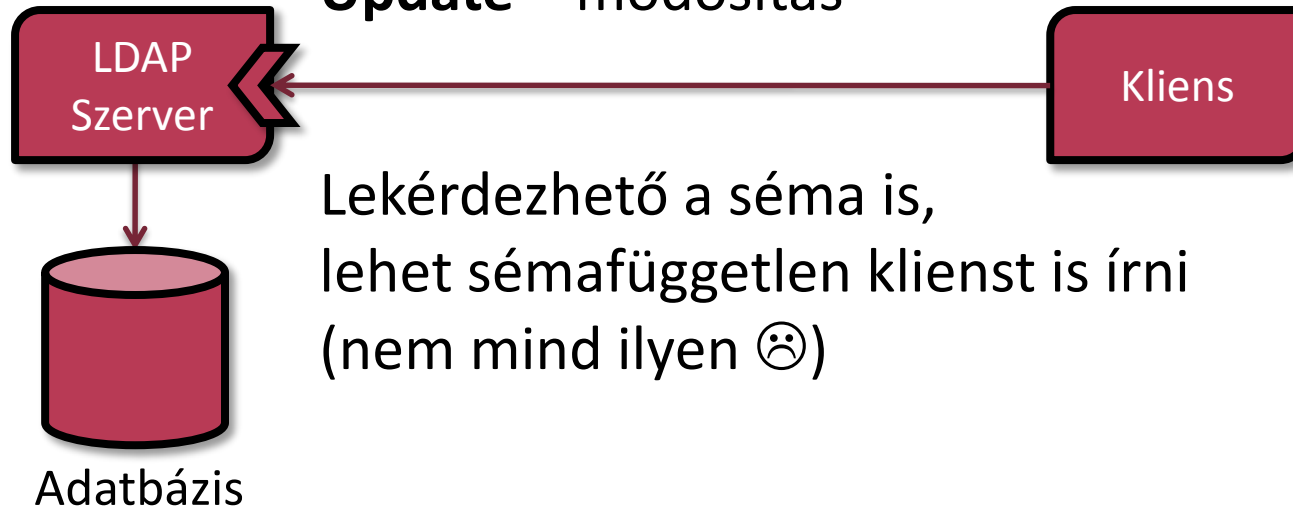
LDAP műveletek

Alapműveletek:

Bind – autentikáció

Search – lekérdezés, keresés

Update – módosítás



Gyakori LDAP osztályok

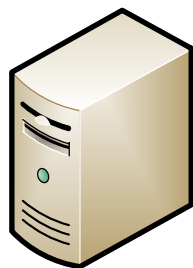
Osztályok és RDN-nek használt attribútumaik

- dcObject
 - Domain component (dc)
- organizationalUnit
 - Organizational unit (ou)
- person
 - Common name (cn)
 - Surname (sn)
- groupOfNames
 - Common name (cn)

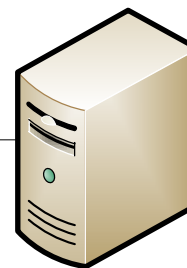
LDAP URL

- Csomópontok egy halmazának kiválasztására
- `proto://host:port/DN?attributes?scope?filter`
 - Proto - ldap/ldaps
 - Host:port – a címtár szerver elérhetősége
 - DN – keresés kiindulóponja
 - Attributes - keresett attribútumok listája
 - Scope – keresés mélysége
 - base: pontosan azt az egy csomópontot keressük
 - one: csak egy szinten keresünk
 - sub: teljes részében keresünk
 - Filter – keresőkifejezés
 - Pl.: `(&(objectClass=maffiaPerson)(uid=don))`
 - kvázi szabványos „prefix” leíró nyelv

Példarendszer



IRF Server
suse.irf



Ubuntu server
ubuntu.irf

OpenLDAP szerver

SSH szerver
Webkiszolgáló
DNS szerver

- PosixUser, PosixGroup és groupOfNames LDAP sémák
 - Linux shell bejelentkezés (PAM modul, pl. SSH)
 - VPN csatlakozás (OpenVPN ldap_auth_plugin)
 - Webes hozzáférés-szabályozás
 - Apache Basic hitelesítés (mod_auth_ldap)
 - Keretrendszer által támogatott hitelesítés (pl. Drupal, WP)

- Hierarchikusan strukturált adatok tárolása
 - DNS (PowerDNS LDAP modul + DNSDomain séma)
 - Növény- és állatrendszertani adatok tárolása

- LDAP menedzsment eszközök
 - Apache Directory Studio
 - Webes menedzsment felület (phpLDAPAdmin)
- Programozási nyelvek
 - Java, C#, PHP, ...
 - Gyakorlatilag bármelyik nyelv rendelkezik megfelelő függvénykönyvtárral

■ PyLDAP

- <http://pyldap.readthedocs.org/en/latest/index.html>

```
#!/usr/bin/env python3
import pprint
from pyldap import LDAPClient
client = LDAPClient()
client.set_credentials("SIMPLE", ("cn=root,dc=irf,dc=local", "**"))
conn=client.connect()

result=conn.search("dc=irf,dc=local", 2, "(cn=cotter)")

pp = pprint.PrettyPrinter(indent=4)
pp.pprint(result)
```

- LDAP adatbázis **parancssorból** történő használata
 - ldapsearch
 - ldapadd
 - ldapmodify
- Jellemző parancssori kapcsolók
 - -x : Egyszerű azonosítás használata
 - -b: Keresés gyökér eleme
 - -D: Felhasználó DN-je
 - -W: jelszó bekérése
 - -H: LDAP szerver URI-je
 - '(ObjectClass=posixAccount)': keresési kritérium

Tartalom

- LDAP
 - LDAP bevezetés
 - LDAP felépítés
 - LDAP a gyakorlatban
 - **Összefoglalás**

Hogyan építsünk LDAP-ot?

Hogyan építsünk LDAP-ot?

- Objektum struktúra ránk van bízva, de ne toljunk ki magunkkal!

Hogyan építsünk LDAP-ot?

- Objektum struktúra ránk van bízva, de ne toljunk ki magunkkal!
 - Mindenkinek lehet gyereke, de célszerű csak DomainComponent vagy OrganizationalUnitokat használni tartalmazóelemként

Hogyan építsünk LDAP-ot?

- Objektum struktúra ránk van bízva, de ne toljunk ki magunkkal!
 - Mindenkinek lehet gyereke, de célszerű csak DomainComponent vagy OrganizationalUnitokat használni tartalmazóelemként
 - A DomainComponentek célszerű, ha követik a DNS névhierarchiát, de ez nem kötelező

Hogyan építsünk LDAP-ot?

- Objektum struktúra ránk van bízva, de ne toljunk ki magunkkal!
 - Mindenkinek lehet gyereke, de célszerű csak DomainComponent vagy OrganizationalUnitokat használni tartalmazóelemként
 - A DomainComponentek célszerű, ha követik a DNS névhierarchiát, de ez nem kötelező
 - Csoportosítsunk típusok szerint (pl. Group-ok és Personok külön részfába), illetve szervezeti egységek szerint is

Hogyan építsünk LDAP-ot?

- Objektum struktúra ránk van bízva, de ne toljunk ki magunkkal!
 - Mindenkinek lehet gyereke, de célszerű csak DomainComponent vagy OrganizationalUnitokat használni tartalmazóelemként
 - A DomainComponentek célszerű, ha követik a DNS névhierarchiát, de ez nem kötelező
 - Csoportosítsunk típusok szerint (pl. Group-ok és Personok külön részfába), illetve szervezeti egységek szerint is
 - A tartalmazás rendtartási célt szolgál, ne hordozzon funkcionális jelentést

Hogyan építsünk LDAP-ot?

- Objektum struktúra ránk van bízva, de ne toljunk ki magunkkal!
 - Mindenkinek lehet gyereke, de célszerű csak DomainComponent vagy OrganizationalUnitokat használni tartalmazóelemként
 - A DomainComponentek célszerű, ha követik a DNS névhierarchiát, de ez nem kötelező
 - Csoportosítsunk típusok szerint (pl. Group-ok és Personok külön részfába), illetve szervezeti egységek szerint is
 - A tartalmazás rendtartási célt szolgál, ne hordozzon funkcionális jelentést
 - Funkcionális csoportosításra Role vagy GroupOfNames

Hogyan építsünk LDAP-ot?

- Objektum struktúra ránk van bízva, de ne toljunk ki magunkkal!
 - Mindenkinek lehet gyereke, de célszerű csak DomainComponent vagy OrganizationalUnitokat használni tartalmazóelemként
 - A DomainComponentek célszerű, ha követik a DNS névhierarchiát, de ez nem kötelező
 - Csoportosítsunk típusok szerint (pl. Group-ok és Personok külön részfába), illetve szervezeti egységek szerint is
 - A tartalmazás rendtartási célt szolgál, ne hordozzon funkcionális jelentést
 - Funkcionális csoportosításra Role vagy GroupOfNames
 - Néha sajnos a kliensek megkötik, hogy milyen osztályt használhatunk, ilyenkor jó a többszörös típusozás

LDAP vs. RDBMS

- Miért LDAP, miért nem relációs adatbázis?

LDAP vs. RDBMS

- Miért LDAP, miért nem relációs adatbázis?
 - Mindegyiknek van előnye és hátránya

LDAP vs. RDBMS

- Miért LDAP, miért nem relációs adatbázis?
 - Mindegyiknek van előnye és hátránya
 - LDAP
 - + Hatékony keresés (hierarchikus is)
 - + Széles támogatottság
 - + Többszörös tipizálás
 - Lassú módosítás

LDAP vs. RDBMS

- Miért LDAP, miért nem relációs adatbázis?
 - Mindegyiknek van előnye és hátránya
 - LDAP
 - + Hatékony keresés (hierarchikus is)
 - + Széles támogatottság
 - + Többszörös tipizálás
 - Lassú módosítás
 - RDBMS
 - + Hatékony keresés
 - + Hatékony módosítás
 - Merev adatmodell

Mire figyeljünk

- Akkor hatékony, ha

Mire figyeljünk

- Akkor hatékony, ha
 - sok a keresés jellegű művelet
 - atomi műveleteket használunk

Mire figyeljünk

- Akkor hatékony, ha
 - sok a keresés jellegű művelet
 - atomi műveleteket használunk
- Veszélyes, ha
 - felhasználókat csak ebben tároljuk

Mire figyeljünk

- Akkor hatékony, ha
 - sok a keresés jellegű művelet
 - atomi műveleteket használunk
- Veszélyes, ha
 - felhasználókat csak ebben tároljuk
 - Ki indítja el az LDAP-ot? („róka fogta csuka” esete)

Mire figyeljünk

- Akkor hatékony, ha
 - sok a keresés jellegű művelet
 - atomi műveleteket használunk
- Veszélyes, ha
 - felhasználókat csak ebben tároljuk
 - Ki indítja el az LDAP-ot? („róka fogta csuka” esete)
 - rendszerfelhasználókat belepakoljuk

Mire figyeljünk

- Akkor hatékony, ha
 - sok a keresés jellegű művelet
 - atomi műveleteket használunk
- Veszélyes, ha
 - felhasználókat csak ebben tároljuk
 - Ki indítja el az LDAP-ot? („róka fogta csuka” esete)
 - rendszerfelhasználókat belepakoljuk
 - Csomagkezelő törli a felhasználót, holott másik hoszton még kellhet
 - Létrejöhet olyan felhasználó ami adott hoszton nem kell

Hozzáférés vezérlés

- Nem jó, ha akárki módosíthatja

Hozzáférés vezérlés

- Nem jó, ha akárki módosíthatja
- Az LDAP-ban tárolunk jelszavakat is →
nem jó, ha bárki bármit olvashat
 - Jelszó lehet cleartext, vagy MD5, SHA1 hash is
 - Nem lehetetlen visszafejteni a hash-et sem...

Hozzáférés vezérlés

- Nem jó, ha akárki módosíthatja
- Az LDAP-ban tárolunk jelszavakat is →
nem jó, ha bárki bármit olvashat
 - Jelszó lehet cleartext, vagy MD5, SHA1 hash is
 - Nem lehetetlen visszafejteni a hash-et sem...
- Hozzáférés szabályozható:
 - Objektum vagy részfa szinten
 - Séma szinten (osztály típus, vagy attribútumra szűrés)

Hozzáférés vezérlés

- Nem jó, ha akárki módosíthatja
- Az LDAP-ban tárolunk jelszavakat is →
nem jó, ha bárki bármit olvashat
 - Jelszó lehet cleartext, vagy MD5, SHA1 hash is
 - Nem lehetetlen visszafejteni a hash-et sem...
- Hozzáférés szabályozható:
 - Objektum vagy részfa szinten
 - Séma szinten (osztály típus, vagy attribútumra szűrés)
- Az LDAP felhasználói is az LDAP-ban tárolódnak