

Ügyfél-kiszolgáló felépítmény megvalósítása az objektum-orientált technológia segítségével. Többszörösségek. Állapottal rendelkező és állapot nélküli kiszolgálók.

Ügyfél-kiszolgáló logikai rétegek

- 1 réteg: Front-end, előtér : GUI (grafikus kezelői felület)
- 2 réteg: Középréteg: Üzleti logika (BOM, Business Object Model)
- 3 réteg: Back-end, háttér : Adatbázis

Kétrétegű ügyfél-kiszolgáló változatok

1. Üzleti logika és adatbázis ugyanazon a gépen
2. Központi kezelői felület és távoli terminálok
3. Üzleti logika részben az ügyfélnél
4. Üzleti logika teljes egészében az ügyfélnél
5. ...ugyanaz helyi adattárolással

Háromrétegű ügyfél-kiszolgáló változatok

1. Klasszikus háromrétegű szerkezet
2. Üzleti logika részben a háttérben
3. Üzleti logika részben az ügyfélnél

ÖTRÉTEGŰ FELÉPÍTMÉNY

- Megjelenítési réteg: a szigorúan vett megjelenítési feladatokért felelős
- Közvetítő réteg: a megjelenítés és az üzleti logika közötti illesztő réteg
- Üzleti Logika
- Adatbázis Hozzáférés:
- Adatbázis

Példa: TIR: Országos hálózati térképészeti alkalmazás regionális (megyei) alközpontokkal

Ügyfél-Kiszolgáló felépítmény megvalósítása:

1. Ügyfél-kiszolgáló összekapcsolás, mint OO kapcsolat (association)
2. Többszörösségek: hány ügyfél használhat egy kiszolgálót
3. Tartalmazási kapcsolat/élettartam: per-client, kliensenkénti kiszolgálópéldány
4. Többszörösségek
 - ügyfél/0..* - a kiszolgáló folyamatosan üzemel
 - ügyfél/1..* - a kiszolgálót az első ügyfél indítja, az utolsó ügyfél lekapcsolja
5. Multi-threading/többszálúság: az ügyfél szolgáltatásai párhuzamosan is hívhatók/egyidejűleg több szolgáltatás is futhat
6. Proxy objektum
7. Korai és kései típuslekötés

Ügyfél-Kiszolgáló (Kliens-Szerver) felépítmény megvalósítása:

1. A kiszolgáló egypéldányú osztály, a szolgáltatások egymástól függetlenek? → állapot nélküli (stateless) kiszolgáló
 - ELŐNY: 1. a szerver összeomlása esetén bármikor, minden további nélkül újra indítható 2. a szolgáltatások nagyobb szabású (durvább szemcsézettségű/granularity) műveletek/tranzakciók
 - HÁTRÁNY: állapotinformációk minden hívást többszörösen megterhelnek
2. A kiszolgáló több osztályt is rendelkezésre bocsát, a szolgáltatások nem függetlenek egymástól?
→ többállapotú (stateful) kiszolgáló

- ELŐNY: egy sok-osztályos OO-terv könnyebben szétosztható több összetevőre
- HÁTRÁNY: 1. csábít a finom szemcsézettségű interakciókra ➔ megterheli a hálózatot. 2. Összeomlás utáni visszaállíthatóság problémás