

Objektum orientált fejlesztés	Vizsga 2004. dec 21.	Jegy:
Név:	Neptun:	Pont:

1. Legyen az alábbi ObjectStore perzisztens osztályunk, amelyet az adatbázisban a "PersRoot"-ból indulva tárolunk.

```
public class PersObj {
    public int a;
    public int geta() { return a;} }

```

Legyen egy tranzien Application programunk, amelyben a fentebb említett adatbázist UPDATE-re nyitjuk meg. Az alábbi programrészlet ebből az Application programból való.

```
Transaction tr = Transaction.begin(ObjectStore.UPDATE);
myobj = (PersObj)db.getRoot("PersRoot");
myobj.a += 23;
tr.commit(ObjectStore.RETAIN_HOLLOW);

```

Mi lesz a hatása a fenti programrészletnek ? Adjon magyarázatot ! Ha elégedetlen az eredménnyel, mit kell tenni ? (6 pont)

**„a” az adatbázisban nem változik meg.
Azért nem, mert a perzisztens PersObj nem tud róla, hogy az „a”-t a tranzien Application megváltoztatta.
A tranzien Application-t persistence aware-ré kell tenni a posztprocesszálskor (-pa switch)**

Mindegyik válasz 2 pont

2. Ha ObjectStore-ban ugyanazon DB adatbázist több szálból akarjuk konkurrens tranzakciókkal kezelni, akkor annak van-e és ha igen mi a feltétele ? (2 pont)

Szálanként önálló session-t kell létrehozni, mert egy session-höz egyszerre csak egy tranzakció tartozhat. Közös session-höz tartozó szálak osztoznak valamennyi adatbázis objektumon (így a tranzakción is).

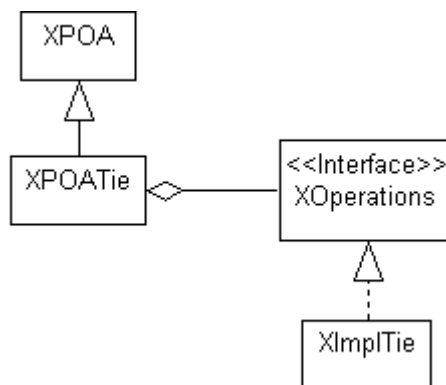
3. Az ObjectStore-ban milyen állapotai vannak egy perzisztens objektumnak ? (3 pont)

hollow..... active

stale

minden helyes válasz 1 pont

4. Adja meg a szervant delegálással történő megvalósítását leíró UML osztálydiagramot ! (5 pont)



Pontozás: osztályok – 1-1 pont
kapcsolatok: 0.5 pont darabja
együtt 5.5 pont - Bonus

5. Ha Servant Locator-t akar használni, akkor hogyan állítja be az alábbi POA policy-ket ? (5 pont)

ServantRetention: **NON_RETAIN**.....

RequestProcessing: **USE_SERVANT_MANAGER**.....
minden helyes válasz 2,5 pont

6. Az Aspektus Orientált programozásban (4 pont)

mik a komponensek? **főprogram, aspektus**.....

hogyan valósul meg a genericitás ? **aspektusok szövése**.....

mik az interfészek ? (1 pont) **szövési pontok**.....

hogyan adaptálunk ? (1 pont) **ragasztó kód**.....

minden helyes válasz 1 pont

7. Mit kell tennünk, ha egy alkalmazás több, független eseménycsatornát is használni szeretne (3 pont)?

minden csatornához külön Event Service-t kell indítani (2 pont).

Ezekhez egyenként csatlakozni kell (1 pont).

8. A Notification Service támogatja Sequence Event-ek használatát. Mire valók ezek, és miért van rájuk szükség? (6 pont)

A sequence eventekkel lehetőség van arra, hogy egy metódushívással egyszerre több eseményt is megkapjunk, illetve átadjunk. (3 pont)

A metódushívásoknál nagy az alap overhead. Minél több paramétert tudunk átadni egy hívással, az egy paraméterre eső hálózati terheltség annál kisebb lesz (2 pont). A szekvenciális eseményekkel így a hálózati terheltséget tudjuk csökkenteni (1 pont).

9. A Trader Service megengedi, hogy szolgáltatás típusok jellemzőit *mandatory* módosítóval lássuk el. Mit jelent ez, és milyen kapcsolata van a dinamikus jellemzőkkel? (4 pont)

A *mandatory* azt jelenti, hogy ajánlat regisztrálásakor a jellemzőnek kötelező értéket megadni. (2 pont). A dinamikus jellemzőkhöz semmi köze (2 pont).

10. Mire képesek és mire nem a *Request Interceptorok*? (4 pont)

Képesek:

- hívás/válasz elkapására, ebben meg tudják nézni a hívási információkat
- rendszer kivétel dobására
- corba objektum hívás kiadására
- a kliens és a szervant tudomása nélkül működhetnek

Nem képesek:

- a hívások paramétereinek és visszatérési értékeinek módosítására

(minden jo 1, minden rossz -1, 5 van benne)

11. A Notification Service az eventekhez definiálja az EventReliability QoS jellemzőt, amely felveheti a PERSISTENT értéket.

- Mit jelent ebben az esetben a perzisztencia (3 pont):

A még át nem vett események a szolgáltatás újraindítása után sem vesznek el, hanem továbbra is elérhetők maradnak.

- Mely feltételeknek kell fennállniuk, hogy a fenti QoS jellemző érvényesüljön (5 pont):

a ConnectionReliability értéknek (1 pont) (csatorna, admin és proxy szinten egyaránt (3 pont)) szintén PERSISTENT-nek (1 pont) kell lennie.

Eredmények értékelése:

Pontszám	<i>Osztályzat</i>
21 -	2
28 -	3
35 -	4
42 -	5