

MI	Név, felvételi azonosító, Neptun-kód: MEGOLDÁS	pont(45) :
-----------	---	------------

Csak felvételi vizsga: <input type="checkbox"/>	csak záróvizsga: <input type="checkbox"/>	közös vizsga: <input type="checkbox"/>
---	---	--

Közös alapképzéses záróvizsga – mesterképzés felvételi vizsga Mérnökinformatikus szak BME Villamosmérnöki és Informatikai Kar

2016. május 31.
MEGOLDÁSOK

A dolgozat minden lapjára, a kerettel jelölt részre írja fel nevét, valamint felvételi azonosítóját, záróvizsga esetén Neptun-kódját!

A fenti táblázat megfelelő kockájában jelölje X-szel, hogy csak felvételi vizsgát, csak záróvizsgát, vagy közös felvételi és záróvizsgát kíván tenni!

A feladatok megoldásához csak papír, írószer, zsebszámológép használata megengedett, egyéb segédeszköz és a kommunikáció tiltott. A megoldásra fordítható idő: 120 perc. A feladatok után azok pontszámát is feltüntettük.

A megoldásokat a feladatlapra írja rá, illetve ott jelölje. Teszt jellegű kérdések esetén elegendő a kiválasztott válasz betűjelének bekarikázása. Kiegészítendő kérdések esetén, kérjük, adjon világos, egyértelmű választ. Ha egy válaszon javítani kíván, teszt jellegű kérdések esetén írja le az új betűjelet, egyébként javítása legyen egyértelmű.

A feladatlapra írt információk közül csak az eredményeket vesszük figyelembe. Az áttekinthetetlen válaszokat nem értékeljük.

A vizsga végeztével mindenképpen be kell adnia dolgozatát. Kérjük, hogy a dolgozathoz más lapokat ne mellékeljen.

Felhívjuk figyelmét, hogy illegális segédeszköz felhasználása esetén a felügyelő kollegák a vizsgából kizárják, ennek következtében felvételi vizsgája, illetve záróvizsgája sikertelen lesz, amelynek letételét csak a következő felvételi, illetve záróvizsga-időszakban kísérelheti meg újból.

Specializációválasztás (Csak felvételi vizsga esetén kell kitölteni)

Kérem, a túloldalon található táblázatokban jelölje meg, mely fő-, illetve mellékspecializáción kívánja tanulmányait folytatni. FIGYELEM! A fő- és mellékspecializációkat külön-külön kell sorrendbe állítani!

Főspecializáció választása
(Csak felvételi vizsga esetén kell kitölteni)

A táblázatban a főspecializáció neve mellett számmal jelölje a sorrendet: 1-es szám az első helyen kiválasztott specializációhoz, 2-es a második helyen kiválasztotthoz tartozik stb. Nem kell az összes főspecializáció mellé számot írni, de legalább egy főspecializációt jelöljön meg.

Főspecializáció	sorrend
Alkalmazott informatika (AUT)	
Internetarchitektúra és szolgáltatások (TMIT)	
Kritikus rendszerek (MIT)	
Mobil hálózatok és szolgáltatások integrációja (HIT)	
Vizuális informatika (IIT)	

Mellékspecializáció választása
(Csak felvételi vizsga esetén kell kitölteni)

A táblázatban a mellékspecializáció neve mellett számmal jelölje a sorrendet: 1-es szám az első helyen kiválasztott specializációhoz, 2-es a második helyen kiválasztotthoz tartozik stb. Nem kell az összes mellékspecializáció mellé számot írni, de legalább egy mellékspecializációt jelöljön meg.

Mellékspecializáció	sorrend
Adat- és médiainformatika (TMIT)	
IT biztonság (HIT)	
IT rendszerek fizikai védelme (HVT)	
Intelligens rendszerek (MIT)	
Mobilszoftver-fejlesztés (AUT)	
Számításelmélet (SZIT)	
Számítási felhők és párhuzamos rendszerek (IIT)	

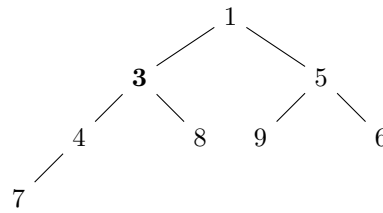
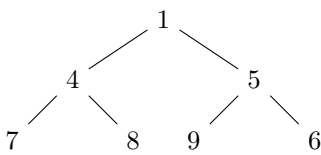
AL	Név, felvételi azonosító, Neptun-kód:	pont(15):
	MEGOLDÁS	

1. Legyen $f(n) = 8n\sqrt{n} + 10(\log_2 n)^2 + 312$ és $g(n) = 15\log_2(n^2) + 3\binom{n}{2}$. Igaz-e, hogy $f(n) = O(g(n))$, illetve, hogy $g(n) = O(f(n))$?

Megoldás: igen; nem

pont(1):

2. Az alábbi kupacon hajtsa végre a BESZÚR(3) műveletet és rajzolja le az eredményt!



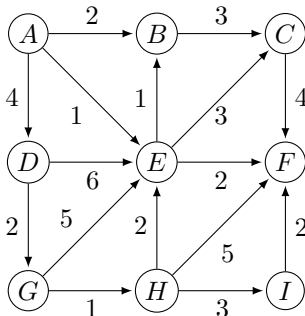
pont(1):

3. Az $1, 2, \dots, 100$ számoknak hány olyan permutációja van, amelyikben az 1 a 2 előtt van, és van közöttük legalább egy másik szám? (Nem szükséges kiszámolni, elegendő egy formulát megadni.)

Megoldás: $98! \cdot \binom{99}{2}$

pont(2):

4. Az alábbi gráfon a Dijkstra-algoritmust használjuk a **D** csúcsból induló legrövidebb utak hosszának meghatározására. A kapott úthosszakat tartalmazó táblázat első 3 sorát feltüntettük. Töltse ki a következő sort!



A	B	C	D	E	F	G	H	I
∞	∞	∞	0	6	∞	2	∞	∞
∞	∞	∞	0	6	∞	2	3	∞
∞	∞	∞	0	5	8	2	3	6
∞	6	8	0	5	7	2	3	6
⋮								

pont(2):

5. Legyen $G = (V, E)$ egy egyszerű, irányítatlan gráf. A \mathcal{T} tulajdonság jelentse a következőt:

Minden $x_1, x_2, x_3, x_4, x_5 \in V$ esetén, van olyan $1 \leq i < j \leq 5$, hogy

- vagy $x_i = x_j$
- vagy $x_i \neq x_j$ és $\{x_i, x_j\} \notin E$.

Fogalmazza meg, milyen ismert gráftulajdonságot ír le \mathcal{T} !

Megoldás: a G gráfban nincs K_5 .

pont(2):

6. Egy f fokú létrán bizonyos fokok annyira rozogák, hogy ha rálépünk, leszakadnak. Szerencsére tudjuk, hogy melyik fokok ilyenek, hova nem szabad lépni. Egy lépéssel legfeljebb 3 fokot tudunk lépni. Legyen $L[i]$ az a szám, hogy hányféleképpen lehet az i -edik fokra feljutni, $L[0] = 1$. Ekkor $L[1] = 1$, ha az első fok jó, egyébként 0. Legyen $4 \leq i \leq f$. Hogyan lehet $L[i]$ értékét meghatározni az $L[0], L[1], \dots, L[i-1]$ segítségével?

Megoldás: $L[i] = 0$, ha az i -edik fok rossz. Egyébként $L[i] = L[i-1] + L[i-2] + L[i-3]$.

pont(2):

7. Tekintsük a következő eldöntési problémákat!

Jelöljön G egy egyszerű irányítatlan gráfot, melynek élei pozitív számokkal vannak súlyozva.

\mathcal{A} : Adott a G gráf és egy k pozitív egész szám. Van G -ben legfeljebb k súlyú feszítőfa?

\mathcal{B} : Adott a G gráf és egy k pozitív egész szám. Van G -ben legalább k súlyú feszítőfa?

\mathcal{C} : Adott a G gráf és egy k pozitív egész szám. Van G -ben legalább k súlyú (egyszerű) út?

Tegyük fel, hogy $P \neq NP$. Jelölje be, melyik igaz az alábbi lehetőségek közül!

	P-beli	NP-teljes
\mathcal{A}	X	
\mathcal{B}	X	
\mathcal{C}		X

pont(2):

8. Egy színháznak két játszóhelye van, a kisszínpad és a nagyszínpad. A színház D féle darab játszik és mindegyikre adott, hogy arra kizárólag a kisszínpad vagy kizárólag a nagyszínpad alkalmas. Műsortervet kell készítenünk az elkövetkező $N \leq D/10$ napra a következő feltételekkel:

- egy napon mindegyik színpadon csak egy darab játszható,
- egy nap minden színész legfeljebb egy darabban szerepelhet (adott, hogy melyikben ki játszik),
- ez idő alatt minden darabból legfeljebb három előadás legyen.

Azt szeretnénk eldönteni, hogy van-e olyan beosztás, amikor az N nap mindegyikén mindkét színpadon van előadás.

Melyik ismert gráfelméleti algoritmussal és azt milyen gráfon futtatva lehet polinom időben megválaszolni a kérdést?

Megoldás:

Legyen G egy páros gráf, melynek csúcsai a lehetséges előadások, azaz minden darab 3 példányban (a kis- és nagyszínpadiak a két osztály). Két csúcsot összekötünk, ha nincs közös színész bennük. A kérdés, hogy van-e N élű párosítás. Ehhez a magyar módszert (maximális párosítást találó algoritmust) használhatjuk. Ha az eredmény legalább N élből áll, akkor van megoldás.

pont(3):

H	Név, felvételi azonosító, Neptun-kód: MEGOLDÁS	pont(7,5):
----------	---	------------

1. Az alábbiak közül mely állítás(ok) igaz(ak) NAT (Network Address Translation) útvonalválasztó használata esetén?

- a) Az IP-címek alapján az állomás nevét adja meg.
- b) Egyetlen, vagy nagyon kevés publikus IP-címmel megoldható a teljes magánhálózat címzése.
- c) Minden alkalmazási rétegbeli protokoll változtatás nélkül használható.
- d) A többi válasz közül egyik sem helyes.

Megoldás: b)

pont(1):

2. Az alábbiak közül mely(ek) *elosztott* többszörös hozzáférési módszer(ek)?

- a) Polling (körbekérdezés)
- b) Probing (csoportos lekérdezés)
- c) CSMA/CD
- d) Reservation (helyfoglalás)
- e) A többi válasz közül egyik sem helyes.

Megoldás: c)

pont(1):

3. Hogy hívjuk azt a HTTP kapcsolatot, ahol egy TCP kapcsolaton belül több HTTP kérés/válasz pár is lehet?

Megoldás: perzisztens

pont(1):

4. IPv4 tördelés esetén ki végezheti el a töredékek összeállítását (ha nem használunk NAT-ot)? A csomópont nevét várjuk válaszként.

Megoldás: címzett

pont(1):

5. A 198.51.100.128/26 hálózatra adja meg a broadcast címet!

Megoldás: 198.51.100.10|11 1111, azaz 198.51.100.191

pont(1):

6. Egészítse ki az alábbi szöveget!

A(z) AS egy olyan hálózatrész, amelyen belül egységes routing módszert alkalmaznak. Ezek között

a(z) csoportba tartozó routing protokollokat használhatjuk, melyeknek leggyakrabban használt változata a BGP.

Megoldás: EGP

pont(1):

7. Az *A* és *B* végpontok közötti kommunikáció során az *A* végpont utolsóként elküldött TCP PDU-jában a sorszám (sequence number) 8350, a hasznos adatrész 1400 byte. A *B* válaszként küldött TCP PDU-jában az ACK-szám 8050. Hány byte-nyi adatot küldhet még *A* a következő nyugta megérkezéséig, ha *B* vételi ablakmérete 2000?

Megoldás: 300

pont(1,5):

O	Név, felvételi azonosító, Neptun-kód: MEGOLDÁS	pont(7,5):
----------	--	------------

Figyelem! Minden feladatnál csak egy helyes válasz van!

1. Az alábbi állítások közül melyik *igaz* a rendszerhívásokkal kapcsolatban?

- a) A rendszerhívásokat az alkalmazásokat programozók direkt módon használják.
- b) Alkalmazói programok a be- és kiviteli műveleteket rendszerhívásokon keresztül érik el.
- c) A rendszerhívás egy függvényhívás.
- d) Az aszinkron rendszerhívás a műveletek befejezése után tér vissza.

Megoldás: b) (a) – hibás, mert a rendszerkönyvtárakon keresztül használják. b) – a műveletek megvalósítva a kernelben vannak, és az alkalmazói programok direkt módon nem érhetik el az I/O-t. c) – hibás, mert lényegében egy megszakításról van szó. d) – hibás, mert az aszinkron hívás csak elindítja a végrehajtást.)

pont(1):

2. Az alábbi állítások közül melyik *hamis* az operációs rendszerek belső szerkezetével kapcsolatban?

- a) Az operációs rendszer magja felelős a feladatok ütemezéséért.
- b) Az operációs rendszer magja kezeli a memóriát.
- c) Az operációs rendszer magja többnyire platformfüggetlen módon kerül megvalósításra.
- d) Az operációs rendszer magja tartalmazza a rendszerhívásokat fogadó réteget.

Megoldás: d) (a rendszerhívásokat fogadó réteg a kernel felett található, a kernel és az alkalmazások között teremti meg a kapcsolatot.)

pont(1):

3. Az alábbi állítások közül melyik *hamis* az egyszerű ütemezési algoritmusokkal (FIFO, RR, SJF, SRTF) kapcsolatban?

- a) A FIFO algoritmus nem preemptív.
- b) Az SJF algoritmus preemptív.
- c) Az RR algoritmus preemptív.
- d) Az SRTF algoritmus preemptív.

Megoldás: b)

pont(1):

4. Az alábbi állítások közül melyik *igaz* a folyamatokkal (process) és a szálakkal (thread) kapcsolatban?

- a) A szál a folyamathoz rendelt CPU-n fut.
- b) A folyamatnak saját halomja (heap) van.
- c) A szálnak saját halomja (heap) van.
- d) Egy folyamat egy szál kontextusában fut.

Megoldás: b) (az a) hamis, mert a folyamathoz nem rendelünk CPU-t, azt a végrehajtás egységéhez, a szálnak rendeljük, a c) verem esetén lenne igaz, a d) meg pont fordítva van.)

pont(1):

5. Az alábbi állítások közül melyik *igaz* a szemaforra vonatkozóan?

- a) A szemafor kezdeti értéke minden alkalmazásban 1, vagyis a szemaforhoz tartozó erőforrás nem foglalt.
- b) Szemaforral nem lehet randevút megvalósítani.
- c) Szemafor alkalmazásával elkerülhetjük a holtpontra létrejöttét.
- d) A számláló (counter) típusú szemafor alkalmazása esetén a szemaforhoz rendelt erőforrás egy időben több párhuzamos feladat által használható.

Megoldás: d) (a – hamis, mert szinkronizáció megvalósítása esetén foglalva inicializáljuk a szemaforra, b – hamis, mert a szemafor alkalmazható szinkronizációra, c – hamis, mert a szemafor alkalmazása esetén a holtpontra létrejöttének szükséges feltételeit nem zárjuk ki.)

pont(1):

6. Az alábbi mondatok közül melyik *hamis* a lapszervezéssel kapcsolatban?

- a) Lapszervezés esetén a logikai lapok és a fizikai memória keretek mérete azonos.
- b) A laptáblát általában a fizikai memóriában tárolják.
- c) A laptábla minden bejegyzése egy fizikai memóriakeretre mutat.
- d) A lapszervezés esetén nincs külső tördelődés.

Megoldás: c) (mert csak azok a lapok kerülnek leképzésre, amiket tényleg használunk (valid/invalid bit), az a) és a d) egymásból következik, és ez az egyik legnagyobb előny, ami miatt használjuk a lapozást, a b) igaz, a TLB-be azt csak cache-elik.)

pont(1):

7. Az alábbi állítások közül melyik *hamis* a fájlrendszerekkel kapcsolatban?

- a) A fájlrendszer feladata az, hogy az információt tároló logikai blokkokat a fizikai fájllokhoz rendelje.
- b) Láncolt listás tárolás esetén az egyes blokkok tárolják a fájlban következő blokkok helyét.
- c) Indexelt tárolás esetén speciális blokkok tárolják a fájl tárolására használt blokkok helyét.
- d) Indexelt tárolás esetén a fájl tetszőleges részének gyors elérése lehetséges.

Megoldás: a) (a fizikai blokkokat kell a logikai fájllokhoz rendelni)

pont(1):

8. Az alábbi két állítás közül melyik *igaz* a PRAM memória-hozzáférési modell esetén?

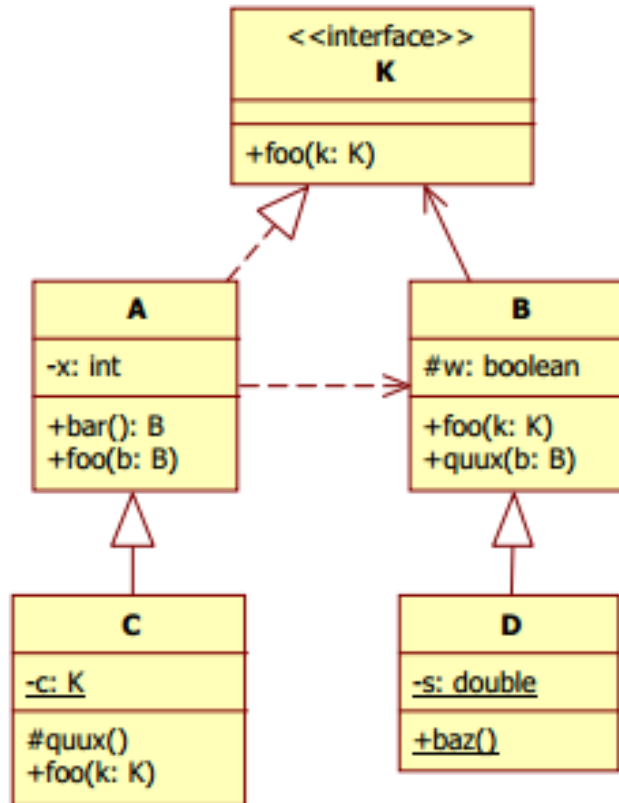
- a) Az olvasás-olvasás ütközés esetén nincs versenyhelyzet.
- b) Az olvasás-írás ütközés esetén nincs versenyhelyzet.

Megoldás: a) (hiszen minden olvasó a helyes értéket fogja olvasni, míg b) esetén a tényleges végrehajtástól függően az olvasás a régi vagy az új értékez is visszaadhatja (van versenyhelyzet).)

pont(0,5):

S1	Név, felvételi azonosító, Neptun-kód: MEGOLDÁS	pont(5):
-----------	---	----------

1. Az alábbi UML2 diagram alapján – a kulcs felhasználásával – jellemezze az állítást!



- A – mindkét tagmondat igaz és a következtetés is helyes (+ + +)
- B – mindkét tagmondat igaz, de a következtetés hamis (+ + -)
- C – csak az első tagmondat igaz (+ -)
- D – csak a második tagmondat igaz (- +)
- E – egyik tagmondat sem igaz (- -)

A-nak `foo(k:K)` metódusa nem kaphat paraméterül B típusú objektumot, mert B-nek is van `foo(k:K)` metódusa.

Megoldás: **B**

pont(1):

2. Mely tesztek irányulnak a következő fázisok kimeneteinek ellenőrzésére?

Követelmények	validálás (validation)
Specifikáció	rendszereszt (system test)

pont(1):

3. Izidor leveszi kedvenc könyvét a könyvespolcról. Megnézi, hogy ki a könyv kiadója, majd a kiadót SMS-ben megküldi Emerenciának.

(i) Adja meg a fenti történet UML2 modelljében előforduló operációk szignatúráit (használja a történetben szereplő szavakat)!

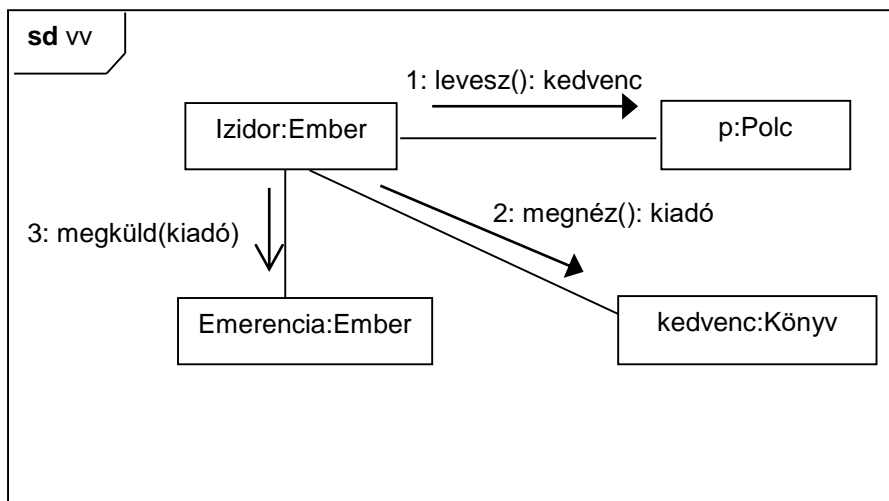
Megoldás:

levesz(): Könyv
megnéz(): Kiadó
oneway megküld(Kiadó)

pont(1):

(ii) A fenti történet alapján rajzoljon UML2 kommunikációs diagramot!

Megoldás:



pont(1):

4. Az alábbiakban a fontosabb szoftver architektúrátípusokat adtuk meg. Válassza ki, hogy a kötegelt feldolgozás (batch) melyik architektúrához illeszkedik!

- a) objektumorientált (object oriented)
- b) eseményszóró (event-based implicit invocation)
- c) absztrakt gép (interpreter)
- d) adatfolyam (pipes and filters)
- e) adattárolás (blackboard)

Megoldás: d)

pont(1):

S2	Név, felvételi azonosító, Neptun-kód: MEGOLDÁS	pont(5):
-----------	---	----------

1. Milyen általános problémát old meg a Factory Method (Metódusgyár) tervezési minta?

Megoldás: Interfészt definiál az objektum létrehozására, de a leszármaztatott osztályra hagyja annak eldöntését, hogy konkrétan melyik osztályból kell példányt létrehozni. Akkor használjuk, ha egy osztály nem látja előre annak az objektumnak az osztályát, amit létre kell hoznia, valamint ha egy osztály azt szeretné, hogy leszármazottai határozzák meg azt az objektumot, amit létre kell hoznia.

pont(1):

2. Mutasson egy C++, Java vagy C# kódrészletet a Singleton tervezési minta implementálására, és mutasson példát a mintának megfelelő osztály használatára!

Megoldás: Egyszerű C++ nyelvű megoldás:

```
class Singleton {
public:
    static Singleton* Instance()
    {
        if (_instance == 0) {
            _instance = new Singleton;
        }
        return _instance;
    }
    void Print() { ... }
protected:
    Singleton() {}
private:
    static Singleton* _instance;
};

Singleton* Singleton::_instance = NULL;
```

Példa használatra:

```
int main()
{
    Singleton::Instance()->Print();
    ...
}
```

pont(1):

3. Milyen általános problémát old meg a Composite (Összetett) tervezési minta?

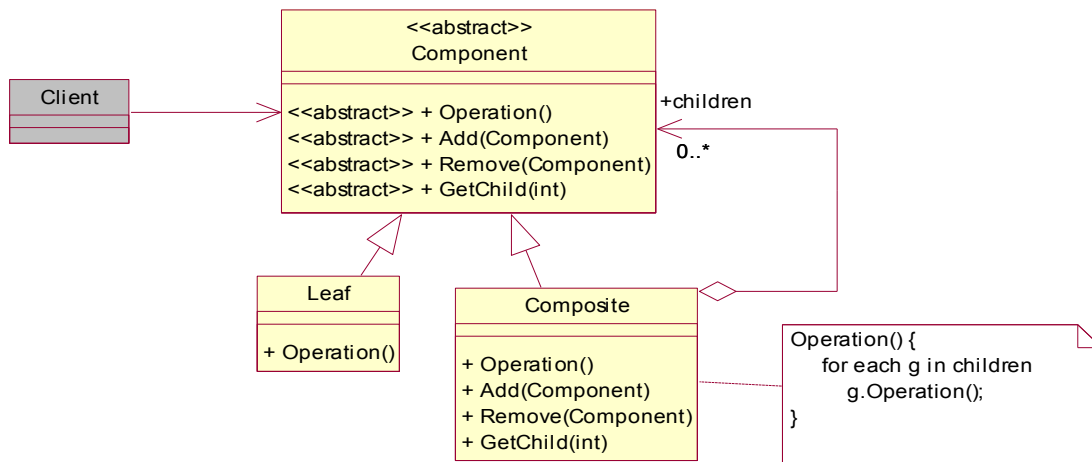
Megoldás: A minta az alábbi problémákra ad megoldást:

- (i) Rész-egész viszonyban álló objektumokat fastruktúrába rendezi.
- (ii) A kliensek számára lehetővé teszi, hogy az egyszerű és összetett objektumokat egységesen kezelje.

pont(1):

4. Mutassa be általánosságában vagy egy példán keresztül a Composite minta működését, ezen belül rajzolja fel a minta osztálydiagramját!

Megoldás:



A Composite minta osztálydiagramja az ábrán látható. Objektumdiagramja fastruktúra, amelyben a Leaf objektum levél, a Composite belső objektum. A Component osztály biztosítja a közös interfészt a két leszármazott használatához. A minta alkalmazása akkor lehetséges, ha a közös interfész absztrahálható. A műveletet mind a Leaf, mind a Composite felülírja, a Composite gyakran az általa tartalmazott Leaf és Composite típusú objektumok azonos nevű függvényeit hívja tovább. Az asszociációkhoz tartozó tömbök kezelését mind az őosztályban, mind a C

pont(1):

5. Tegyük fel, hogy egy adott művelet egy webalkalmazásban kliens (pl. JavaScript) és kiszolgáló (pl. ASPX) oldali kóddal is megvalósítható. Adjon meg egy előnyt a kliens oldali megvalósításra vonatkozóan, és egy tipikus előnyt a kiszolgáló oldali megvalósításra vonatkozóan!

Megoldás:

A kliens oldali szkript (pl. JavaScript) előnye pl.: Gyorsabb, mert nincs szükség interakcióra a kiszolgálóval (vagy ha szükség is van rá, az hatékonyabban, kisebb adatforgalom mellett megtehető).

A kiszolgáló oldali kód előnyei: A kiszolgáló oldali kód általában lefordítható. Így a hibák egy része már fordításkor kiderül, illetve az alkalmazás futása gyorsabb lesz. Kiszolgáló oldali kóddal általában könnyebb böngésző független megvalósítást készíteni.

pont(1):

AD	Név, felvételi azonosító, Neptun-kód: MEGOLDÁS	pont(5):
-----------	---	----------

1. Adott egy $R(ABCDEF)$ séma és az attribútumain egy

$$F = AB \rightarrow CDE, B \rightarrow FA, CD \rightarrow BE, E \rightarrow AD, EF \rightarrow A$$

függőség-halmaz, valamint az R séma $R1(ABC)$, $R2(CDE)$, $R3(ABFD)$, $R4(EAD)$ felbontása. A táblázatos teszt alkalmazásával döntse el, hogy veszteségmentes-e a sémafelbontás!

Megoldás:

	A	B	C	D	E	F
ABC	a	a	a	b₁ a	b₁ a	b₁ a
CDE	b₂ a	b₂ a	a	a	a	b ₂
ABFD	a	a	b₃ a	a	b₃ b₁ a	a
EAD	a	b ₄	b ₄	a	a	b ₄

veszteségmentes

pont(1):

2. Egy 2 000 000 rekordból álló állományt szeretnénk „vödörös hash” szervezéssel tárolni. A rekordhossz 200 byte, egy blokk kapacitása (a fejrészt nem számítva) 2000 byte. A kulcsok 25 byte-osak, egy mutatóhoz 8 byte kell. A rekordok kiolvasására legfeljebb 4 blokkelérési időt engedélyezve számítsa ki, hogy legalább hány byte-os lesz a hash-tábla! (Tételezze fel, hogy a vödörkatalógus kereséskor memóriában tartható és a hash-függvény egyenletesen osztja el a kulcsokat.)

Megoldás: 400 000 byte mert $2000/200 = 10$ rekord/blokk, 200 000 blokk, $8 \cdot 50000$ byte

pont(1):

3. Adjon példát – ha létezik – olyan, legalább 1NF relációs sémára, amely nem 2NF, és nem is bontható fel veszteségmentesen 2NF részsémákba úgy, hogy a sémafelbontás függőségörző is legyen!

Megoldás: Nincs ilyen séma.

pont(1):

4. Az alábbiak közül mely állítások igazak?

Létezik olyan pontosan 3NF séma (nem BCNF), hogy létezik egy rá illeszkedő reláció,

- a) amelynek egyetlen másodlagos attribútumában sincs redundancia.
- b) amely valamennyi elsődleges attribútumában redundáns.
- c) mely az egyik másodlagos attribútumában redundáns.

Megoldás: a), b)

pont(1):

5. Egy relációs adatbázis tervezéséhez mintaadatokat kaptunk az R táblából, ld. alább. A különböző kódok garantáltan különböző értékeket jelölnek. Kijelenthető-e a táblázat alapján, hogy igazak az $R(ABCD)$ sémán a következő funkcionális függőségek?

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b2	c1	d2
a2	b2	c2	d1

a) $A \rightarrow B$

b) $AC \rightarrow B$

Megoldás: Egyik sem jelenthető ki.

pont(1):