

<b>MI</b>	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

**2015. január 5.**  
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 feladatlagra 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 feladatlagra 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.

<b>Főspecializáció</b>	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.

<b>Mellékspecializáció</b>	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)	

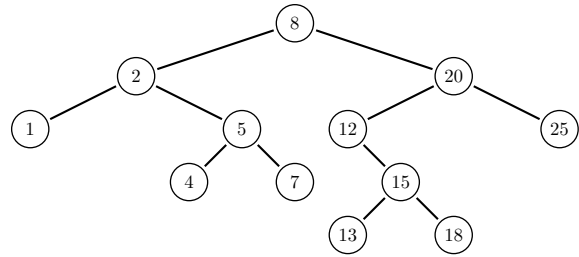
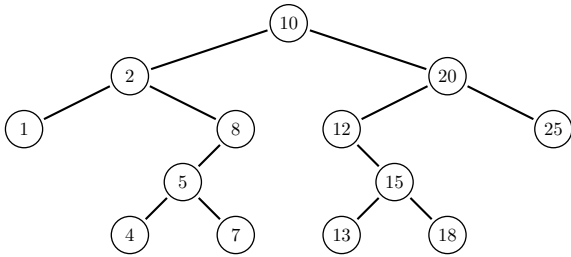
<b>AL</b>	Név, felvételi azonosító, Neptun-kód:	pont(15):
	MEGOLDÁS	

1. Legyen  $f(n) = 4^n + (-4)^n + 2^n$  és  $g(n) = 3^n$ . Ekkor igaz-e, hogy  $f(n) = O(g(n))$ , illetve, hogy  $g(n) = O(f(n))$ ?

Megoldás: egyik sem igaz

pont(1):

2. Az alábbi bináris keresőfára alkalmazza a TÖRÖL(10) eljárást!



pont(1):

3. Az 1, 2, ..., 20 számok közül hányféleképpen tudunk öt különbözőt kiválasztani úgy, hogy az összegük páratlan legyen?

Megoldás:  $\frac{1}{2} \cdot \binom{20}{5} = \binom{10}{5} + \binom{10}{3} \cdot \binom{10}{2} + \binom{10}{1} \cdot \binom{10}{4}$

pont(2):

4. Egy irányított gráfban Floyd algoritmusát használva határozzuk meg a legrövidebb út hosszát az összes pontpárra. Kezdetkor az úthosszakat tároló  $F = F_0$  tömböt az élek súlyaival töltöttük fel. A 3. menet után az itt látható  $F = F_3$  tömböt kaptuk. Mi lesz a következő menet után kapott  $F_4$  tömbben az alábbi két érték?

$$\begin{pmatrix} 0 & \infty & 5 & 2 & \infty & \infty \\ 1 & 0 & 6 & 3 & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 4 & 0 & \infty & 2 \\ 2 & 1 & 7 & 4 & 0 & \infty \\ \infty & \infty & 1 & 5 & \infty & 0 \end{pmatrix}$$

$F_4[2,4] = 3$

$F_4[2,6] = 5$

pont(2):

5. A  $G = (V, E)$  irányított gráf egy város úthálózatát írja le, a csúcsok a csomópontok, az élek a köztük levő útszakaszok. Amikor a városban egy adott  $x \in V$  csomópont lezárásra kerül (az összes ezt érintő útszakasszal együtt), egy  $(a, b) \in E$  útszakasz egyirányúból kétirányúvá tételével próbáljuk a közlekedést fenntartani.

Melyik ismert algoritmussal milyen bemeneten ellenőrizhetjük hatékonyan, hogy adott  $x, a, b$  esetén ily módon megoldható-e, hogy a városban bármely csomópontból bármely csomópontba el tudjunk jutni (természetesen az egyirányú utcákon is szabályosan közlekedve)?

*Megoldás:* Módosítjuk a gráfot: kihagyjuk az  $x$  csúcsot, és hozzávesszük a  $(b, a)$  élt is. DFS egy  $y$  pontból ezen a  $G'$  gráfon és annak megfordítottján. Ha  $y$ -ből mindkétyszer mindent elérünk, akkor igen a válasz, egyébként nem. (De ezen a gráfon lehet  $|V|$  db bejárás is.)

pont(2):

6. Legyen  $G = (V, E)$  egy egyszerű, irányítatlan gráf. A  $\mathcal{T}$  tulajdonság jelentse azt, hogy bármely  $a, b, c \in V$  esetén ha  $\{a, b\} \in E$ , akkor  $\{b, c\} \notin E$  vagy  $\{a, c\} \notin E$ . Fogalmazza meg, milyen gráftulajdonságot ír le  $\mathcal{T}$ !

*Megoldás:* A gráfban nincs háromszög.

pont(2):

7. Tekintsük a következő eldöntési problémákat!

$\mathcal{A}$ : Adott egy  $G$  irányítatlan gráf és  $k, \ell > 0$  egész számok.  
Van  $G$ -nek pontosan  $k$  csúcsú és  $\ell$  élű részgráfja?

$\mathcal{B}$ : Adott egy  $G$  irányítatlan gráf.  
Van  $G$ -nek 2014 csúcsú 4102 élű részgráfja?

Az alábbi következtetések közül melyik helyes, melyik nem:

Ha  $P=NP$ , akkor  $\mathcal{A}$  Karp-redukálható (polinomiálisan visszavezethető)  $\mathcal{B}$ -re.

igen – nem

Ha  $\mathcal{A}$  Karp-redukálható (polinomiálisan visszavezethető)  $\mathcal{B}$ -re, akkor  $P=NP$ .

igen – nem

Ha  $\mathcal{B}$  Karp-redukálható (polinomiálisan visszavezethető)  $\mathcal{A}$ -ra, akkor  $P=NP$ .

igen – nem

pont(2):

8. Egy város úthálózatát a  $G = (V, E)$  irányítatlan gráf írja le. A gráf élei a megfelelő útszakaszok hosszával vannak súlyozva. Az élek között meg van jelölve az a 10 darab, amelyek egy-egy alagutat reprezentálnak. Adott  $a \in V$  helyről adott  $b \in V$  helyre szeretnénk a legrövidebb olyan úton eljutni, amely legfeljebb 2 alagúton vezet át.

Vázzon egy algoritmust, ami ha a gráf a szomszédossági mátrixával adott, akkor  $O(|V|^2)$  lépésben megtalál egy, a feltételeknek megfelelő útvonalat!

*Megoldás:*

Dijkstra-algoritmust futtatunk  $a$ -ból azon a gráfon, amelyben: nincs alagút, 1 alagút van (10 eset), 2 alagút van ( $\binom{10}{2}$  eset). Az eredmények közül a legrövidebb út a megoldás.

pont(3):

<b>H</b>	Név, felvételi azonosító, Neptun-kód:  MEGOLDÁS	pont(7,5):
----------	---	------------

1. Miért kell az IPv4 fejrész Header Checksum mezőjének a tartalmát minden továbbítási lépésben újraszámolni?
- a) Egyáltalán nem kell, sőt az hibát okozhat.
  - b) Csupán biztonsági okból, hogy frissítsük a biteket.
  - c) Mert a fejlécben mindig megváltoztatunk valamit a továbbítás során.
  - d) Mert menetközben a csomag adatrésze sérülhetett.

Megoldás: c)

pont(1):

2. Az alábbi állítások közül mely(ek) igaz(ak) a DNS-szerverre?

- a) A TCP 53-as porton figyel a bejövő zónaeltöltési kéréseket.
- b) Az ICMP 53-as porton figyel a bejövő névfeloldási kéréseket.
- c) Csak egy zónát szolgálhat ki.
- d) A másodlagos DNS-szerver írható és olvasható is.
- e) A DNS szervereket az MX rekord jelöli.

Megoldás: a)

pont(1):

3. Az alábbi routingtáblát figyelembe véve melyik interfészen kerül továbbításra a következő cél-IP-címet tartalmazó IP-csomag?

11001000 00010111 00011001 10100101

Prefix	Interfész
11001000 00010111 00010	0
11001000 00010111 00011	1
11001000 00010111 00011000	2
egyébként	3

Megoldás: Az 1-es jelűn.

pont(1):

4. A Bellman-Ford-algoritmus alkalmazása során egy adott időpontban a hálózat  $A$  csomópontja a következő állapotvektort tartja nyilván:

$B, 1$	$C, 2$	$D, 3$	$E, 4$	$F, 1$
--------	--------	--------	--------	--------

Megérkezik  $B$ -től a következő állapotvektor:

$A, 1$	$C, 3$	$D, 2$	$E, 2$	$F, 3$
--------	--------	--------	--------	--------

Mely bejegyzéssel/bejegyzésekkel bővíti ill. módosítja  $A$  az állapotvektorát?

Megoldás:  $E, 3$

pont(1):

5. Egészítse ki az alábbi, levelező rendszerekről szóló szövegrészt a megfelelő rövidítéssel!

„A felhasználók a levelező programjuk segítségével a helyi levelező kiszolgálónak továbbítják a levelet az SMTP protokoll segítségével, amely a címből a levél további útját a DNS protokoll segítségével a(z) ..... rekord alapján határozza meg.”

Megoldás: MX

pont(1):

6. Adja meg annak az autonóm rendszernek a típusát, amelyre a következő állítások igazak:

- egynél több autonóm rendszerrel van összekötve
- meghibásodás esetén sem lesz elvágva
- nem enged átmenő forgalmat más autonóm rendszerek között

*Megoldás:* 6. multihomed

pont(1):

---

7. Az  $A$  és  $B$  végpontok közötti kommunikáció során az  $A$  végpont utolsóként elküldött TCP PDU-jában a sorszám (sequence number) 6740, a hasznos adatrész 130 byte.  $B$  válaszként küldött TCP PDU-jában az ACK-szám 6250. Hány bájtnyi adatot küldhet még  $A$  a következő nyugta megérkezéséig, ha az ablakméret 720?

*Megoldás:* 100

pont(1,5):

<b>O</b>	Név, felvételi azonosító, Neptun-kód:  MEGOLDÁS	pont(7,5):
----------	---	------------

*Figyelem! Minden feladatnál csak egy helyes válasz van!*

1. Az alábbi megállapítások közül melyik *igaz* a valósidejű operációs rendszerekkel kapcsolatban?

- a) A valósidejű operációs rendszerek gyorsak.
- b) A kemény valósidejű operációs rendszerek rendszerhívásainak válaszidejére felső korlát adható.
- c) A valósidejű operációs rendszerek kritikus szolgáltatásainak válaszidejére felső korlát adható.
- d) A lágy valósidejű operációs rendszerek mindig prioritásosak.

Megoldás: c)

pont(1):

2. Az alábbi állítások közül melyik *hamis* a korai operációs rendszerekkel kapcsolatban?

- a) A rezidens monitorok az előző munka befejezése után automatikusan elindították a következő munkát.
- b) Az időosztásos rendszereket az on-line felhasználói feladatok kedvezőbb válaszidejének biztosítására fejlesztették ki.
- c) Az időosztásos rendszerekben a batch feladatok is időosztás módon, Round-Robin ütemezéssel futnak.
- d) Az első számítógépek esetén a feladatok ütemezése a rendszert üzemeltető humán operátoroknak volt a feladata.

Megoldás: c)

pont(1):

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

- a) A FIFO algoritmusban jelentkezhet a Konvoj-hatás.
- b) Az RR algoritmusban nem jelentkezhet a Konvoj-hatás.
- c) Az SJF algoritmusban nem jelentkezhet a Konvoj-hatás.
- d) Az SRTF algoritmusban nem jelentkezhet a Konvoj-hatás.

Megoldás: c)

pont(1):

4. Az alábbi állítások közül melyik *igaz* a feladatok tipikus állapot-átmeneti diagramjával kapcsolatban?

- a) A feladatok VÁRAKOZÓ állapotban jönnek létre.
- b) Kooperatív (nem preemptív) operációs rendszerben a FUTÓ feladat nem kerülhet vissza FUTÁSRA KÉSZ állapotba direkt módon (csak a VÁRAKOZÓ állapoton keresztül).
- c) A FUTÁSRA KÉSZ feladatok közül a hosszú távú ütemező választja ki a FUTÓ feladatot.
- d) Ha a rendszerben a FUTÓ feladat kilépésekor nincs FUTÁSRA KÉSZ feladat, akkor az IDLE rendszerfeladat fog futni (ez mindig FUTÁSRA KÉSZ).

Megoldás: d)

pont(1):

---

5. Az alábbi állítások közül melyik *hamis* a közös erőforrásokra vonatkozó kölcsönös kizárás tekintetében?

- a) A feladat által használt közös memóriára biztosítani kell a kölcsönös kizárást.
- b) Az operációs rendszerek mutex szolgáltatása egy példányos közös erőforrások esetén alkalmazható a kölcsönös kizárás megvalósítására.
- c) Többpéldányos közös erőforrások esetén a számláló (counter) típusú szemafor alkalmazható a kölcsönös kizárás megvalósítására.
- d) Ha a feladatnak egy többpéldányos erőforrásból többre van szüksége, akkor az egyesével problémamentesen lefoglalható.

Megoldás: **d)**

pont(1):

---

6. Az alábbi mondatok közül melyik *nem* szükséges feltétele a holtpontnak?

- a) A rendszerben legyen erőszakos erőforrás-elvétel.
- b) A rendszerben alkalmazzunk kölcsönös kizárást.
- c) A rendszerben feladatok foglaljanak új erőforrásokat úgy, hogy eközben további erőforrások foglalására tegyenek kísérletet.
- d) A holtpontnban lévő P0-Pi feladatok egymásra várjanak.

Megoldás: **a)**

pont(1):

---

7. Az alábbi lapszervezéssel kapcsolatos állítások közül melyik *hamis*?

- a) Lapszervezés esetén nincs külső tördelődés.
- b) Lapszervezés esetén a laptábla csökkenti a szabad fizikai memória méretét, hiszen az is ott kerül tárolásra.
- c) Az üres (nem használt) fizikai memória keretek listáját a kerettáblában tároljuk.
- d) A lapszervezés esetén a memória-hozzáférés sebessége nem változik a változó méretű partíciók módszeréhez képest.

Megoldás: **d)**

pont(1):

---

8. Az alábbi két állítás közül melyik *igaz* a permanens táron az egyes fájlokhoz tartozó blokkok azonosítására (allokációs stratégia) szolgáló megoldásokkal kapcsolatban?

- a) A láncolt tárolás esetén a fájl tetszőleges része közvetlenül elérhető.
- b) Az indexelt tárolás esetén a fájl tetszőleges része közvetlenül elérhető.

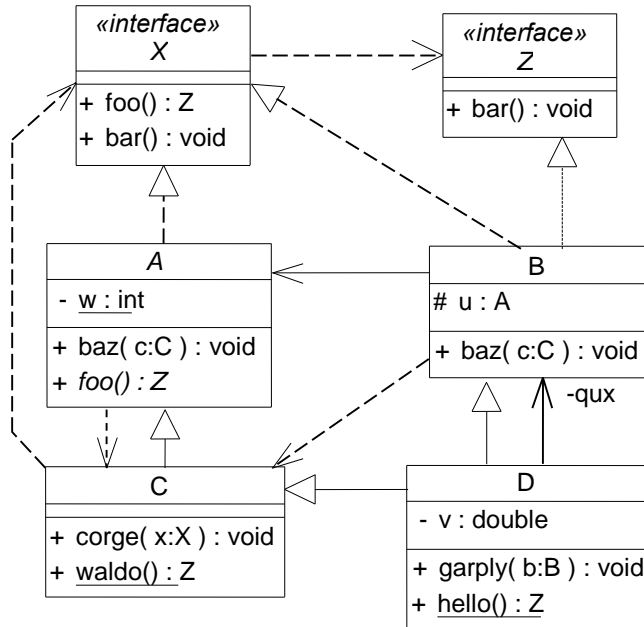
Megoldás: **b)**

pont(0,5):



<b>S1</b>	Név, felvételi azonosító, Neptun-kód:	pont(5):
	MEGOLDÁS	

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 (- -)

C-nek a `corge` metódusa kaphat paraméterül D típusú objektumot, ezért a metódus meghívhatja a kapott objektum `garply` metódusát.

Megoldás: C

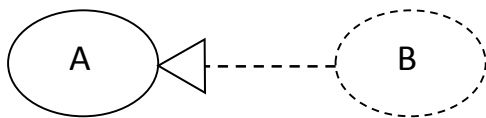
pont(1):

2. A szoftverfejlesztés melyik fázisának célja „a rendszer fő komponenseinek azonosítása és a közöttük fennálló együttműködés definiálása” ?

Megoldás: Architektúrális tervezés

pont(1):

3. Jelölje meg, mi A és B az alábbi UML2 diagramon?



- A:
- operáció
  - állapot
  - use-case
  - processz

- B :
- kollaboráció
  - metódus
  - beágyazott állapot
  - feltételes use-case

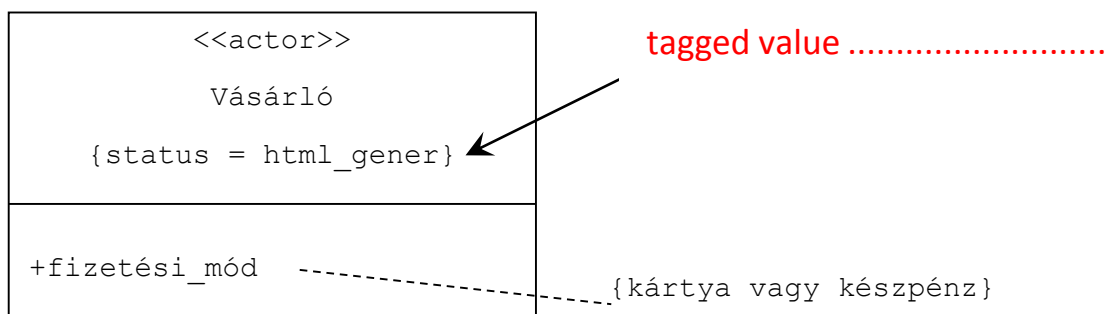
pont(1):

4. Az UML2-ben a gyűjteményeknek (kollektcióknak) két fontos tulajdonsága van: rendezettség (ordered) és egyediség (unique). Írja be a táblázatba a megfelelő UML2 kollektció nevét!

rendezett	egyedi	UML2 kollektció neve
nem	nem	bag

pont(1):

5. Adja meg, hogy a jelölt elem melyik UML2 meta-modell elem példánya!



pont(1):

<b>S2</b>	Név, felvételi azonosító, Neptun-kód:  MEGOLDÁS	pont(5):
-----------	---	----------

1. Adja meg két-három pontban, miben és hogyan segítenek a tervezési minták a szoftvertervezés során!  
Figyelem: *ne* a tervezési minta definícióját adja meg!

*Megoldás:*

- Növelik a rendszer karbantarthatóságát, módosíthatóságát;
- Növelik az egyes részek újrafelhasználhatóságát;
- Segítenek megtalálni a nem maguktól értetődő osztályokat.

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?

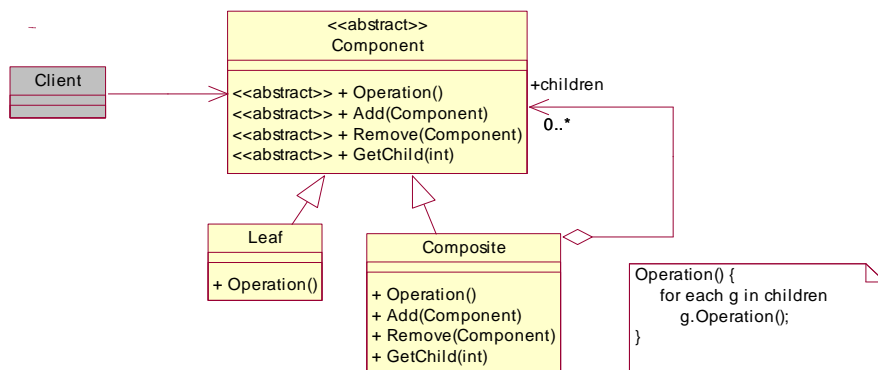
*Megoldás:*

- (i) Rész-egész viszonyban álló objektumokat fastruktúrába rendez.
- (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 Composite-ben megvalósíthatjuk.

pont(1):

5. Tegyük fel, hogy egy adott műveletet 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) előny pl.:

– Gyorsabb, mert nincs szükség interakcióra a kiszolgálóval (vagy ha szükség is van rá, az hatékonyabban, kisebb adatforgalom mellett megtehető).

A kiszolgáló oldali kód előnyök:

– 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):

<b>AD</b>	Név, felvételi azonosító, Neptun-kód:  MEGOLDÁS	pont(5):
-----------	---	----------

1. Mit veszíthetünk, ha egy relációs adatbázis tervezése során nem veszteségmentes sémafelbontásokat készítünk?

*Megoldás:* információt

pont(1):

2. Egy táblában 100 000 db 100 byte-os rekord található, amelynek a kulcsán B\*-fa indexet definiáltak. A kulcs 20 byte-os, egy mutató 5 byte, a blokkméret 1000 byte. Számítsa ki, hogy hány százalékkal nő a kulcs alapján történő keresés ideje abban az esetben, ha minden (adat- és index)blokk csak félig van tele, ahhoz képest, amikor minden blokk tele van!

*Megoldás:* 25%-kal

pont(1):

3. Az  $R(A, B, C, D)$  séma melyik attribútumaiban valósulhat meg redundáns adattárolás funkcionális függés következtében, ha az attribútumok között az  $ABD \rightarrow BC$  és  $C \rightarrow BC$  függések ismertek?

*Megoldás:* A B attribútumban.

pont(1):

4. Minimalizálja az alábbi függéshalmazt!

$$F = \{A \rightarrow BA, C \rightarrow ABC, D \rightarrow DF, CD \rightarrow EAF\}$$

*Megoldás:*  $F_{\min} = \{A \rightarrow B, C \rightarrow A, D \rightarrow F, CD \rightarrow E\}$

pont(1):

5. Adott egy  $R$  relációs séma, és az attribútumainak  $X$  és  $Y$  részhalmazai. Mik a legtágabb feltételei annak, hogy az  $(XY, R \setminus Y)$  sémafelbontás az  $X \rightarrow Y$  funkcionális függés esetén veszteségmentes legyen?

*Megoldás:* Ne legyen  $Y$ -nak olyan attribútuma, amely triviálisan függ  $X$ -től.

pont(1):