

ADATSZERKEZETEK TÁRGY

2009/10 TANÉV 1. FÉLÉV



ÁLTALÁNOS KÉRDÉSEK

Az Adatszerkezetek 2 féléves tárgya, mindkét félévben kollokvium és gyakorlati jegy van.

Követelmények az aláírásért:

- a papíros zárthelyi (✍) megírása,
- beadandó (📄) feladat érdemi megoldása és beadása,
- a (🖨) gépes zárthelyi megírása,
- rendszeres gyakorlatra járás (a vizsgaszabályzatban rögzítettek szerint).

Követelmények a legalább kettes gyakorlati jegyért:

- a papíros zárthelyi legalább 2,
- a beadandó feladat legalább 2,
- a gépes zárthelyi legalább 2.

A gyakorlati jegybe a zárthelyik, beadandók és a házi feladat(ok) eredményei mellett a gyakorlaton való részvétel minősége is beleszámít.

Ha a(z) „elméleti” és géptermi) zárthelyi valamelyikét nem írta meg legalább elégségesre, akkor a pótzárthelyi(ke)t meg **kell** írnia!

A beadandóval szemben támasztott feltétel, hogy

1. a program legyen képes rövid (2-3 oldalnyi!) nyitó **tájékoztató** megjelenítésére,
2. mivel a megoldáshoz néhány „komplex” adatszerkezetre (listára, veremre, sorra stb.-re) van szükség, ezért ezt (ezeket) önálló modulban (unit-ban, include-állományban), elkülönítve kell megvalósítani. Azaz a program algoritmusai és a kódja is **legalább két-két független programegységet** alkot.
3. a program helyes működésének dokumentálásához szükséges jónéhány (legalább 3) elvileg különböző **tesztadatsor**, fájlokban. Ezeket az exe mellett kell elhelyezni az anyagban.

A beadandó feladat beadásáról (📄):

1. a meghatározott **formai feltételeknek** eleget nem tevő dolgozatokat nem értékeljük (azt újra be kell adni helyesen; azonban a határidő nem módosul);
2. a beadás **elektronikusan** történik a gyakorlatvezetőnek ([moodle](#) rendszerén keresztül), a dokumentációt² ettől függetlenül papíron is beadhatja (ami azonban nem helyettesíti az elektronikus beadást!), ekkor kaphat *részletes értékelést* munkájáról;
3. a beadási határidő: **2009.11.29.** vasárnap 24 óra

¹ Az „elfogadható” hiányzások számaránya legfeljebb 25%, aggályos a 25-50% közötti, az 50%-ot meghaladó hiányzás esetén a gyakorlati jegy adását meg kell tagadni.

² A dokumentációhoz mintául szolgálhat az innen (<http://people.inf.elte.hu/szlavi/PrM2felev/ADTBeaMintaDoku.doc>) letölthető anyag, amelyet értelemszerű átalakításokkal kell felhasználni.




4. késedelmes beadás esetén a jegyet **hetente 1 jeggyel** csökkentjük (legfeljebb **3** hetes késéssel fogadjuk el a beadandót);




RÉSZLETES TEMATIKA

A tematikában elsősorban az alábbi irodalmakra építünk és hivatkozunk:

1. Módszeres programozás – Programozási bevezető (*ulógia* 18),
2. Módszeres programozás – Adattípusok (*ulógia* 34),
3. Módszeres programozás – A programkészítés technológiája (*ulógia* 21),
4. Adatszerkezetek példatár (*ulógia* 45),
5. Módszeres programozás – Rekurzív típusok (*ulógia* 27).

A TOVÁBBI KOMMUNIKÁCIÓHOZ AZ **ETR** KURZUS-MAIL SZOLGÁLTATÁSÁT FOGOM HASZNÁLNI!

Sorszám/ Ea.-dátum	Előadás	Gyakorlat
1 09.02.	Regisztrációs hét	
2 09.09. 	Adatokkal kapcsolatos fogalmak, különös tekintettel a típus általános és a mutató típus fogalmára. A típusdefiniálás nyelvi keretei. (Modulok, unit, include.)	Ismétlés – típusdefiniáció, -deklaráció, eljárások, függvények, paraméterezés, unit
3 09.16. 	Modulteszt Tömb <ul style="list-style-type: none"> • memóriamodellek, folytonos és láncolt ábrázolás • vektorok és mátrixok • speciális tömbök • hézagosan kitöltött tömbök 	Mutatótípus – a lényeg, ábrázolás, műveletek, alkalmazás
4 09.23.	Verem, sor láncolt ábrázolással	Hézagosan kitöltött tömbök láncolt ábrázolása – ritkaMátrix-modul, alkalmazás
5 09.30.	Kupac, prioritási sor kupaccal	Verem láncolt ábrázolással – verem-modul („elem típus cseréje” probléma és megoldása), alkalmazás
6 10.07.	Dinamikusan és statikusan láncolt lista	Sor láncoltan – sor-modul, alkalmazás (egy „elemi szimulációs” feladat)
7 10.14.	Táblázat <ul style="list-style-type: none"> • kulcs-transzformáció • táblázatrepresentációk 	Prioritási sor kupaccal, alkalmazása rendezésre
8 10.21.	Rekurzív típusok 1. – alapok <ul style="list-style-type: none"> • a rekurzió mint típuskonstrukciós eszköz  Az önálló feladat kiosztása.	Lista láncolt ábrázolással – lista-modul (keretmodulból kiindulva) Lista láncolt ábrázolással – alkalmazások
9 10.28.	Őszi szünet	
10 11.04.	Rekurzív típusok 2. – bináris fák <ul style="list-style-type: none"> • bináris fák • kereső fák • kiegyensúlyozott fák • rendező fák (heapsort) 	Táblázat <ul style="list-style-type: none"> • kulcsütközések megoldási módjai • kulcsütközési statisztikák • függvények megvalósítása és kipróbálása az elvárt tulajdonságok szempontjából

11 11.11.	<i>✍ Papíros zárthelyi</i>	Különböző típusok alkalmazásai (gyakorlás a gépes zh-ra)
12 11.18.	Rekurzív típusok 3. – nem bináris fák <ul style="list-style-type: none"> • ábrázolásuk • B-fák 	BinFa modul megvalósítása
13 11.25.	 <i>Géptermi zárthelyi: 8-11 (KisLovi)</i>  <i>Az önálló feladat beadása: 11.29. éjfélig</i>	Bejárások és keresések a BinFa modulra építve
14 12.02.	Konzultáció	Rekurzív algoritmusok fákra
15 12.09.	<i>Vizsga zárthelyi 8-11 (KisLovi)</i>	<i>✍ Pót papíros /  pót gépes zárthelyi: 12.16. 8-10 / 10-13 (KisLovi)</i>

Gyakorlati jegy UV: 2009.12.21. hétfő 8-12 (KisLovi)

Tudnivalók:

- Egy [komplex feladat](#) megoldása papíron és számítógéppel.
- Minden saját papíralapú és elektronikus anyag használható.

Vizsga³: 2010.01.06. szerda 8-12 (KisLovi)

Utóvizsga: 2010.01.20. szerda 8-12 (KisLovi)

Vizsgatételek:

#	Tétel
1.	Adatokkal kapcsolatos fogalmak, különös tekintettel a típus általános és a mutató típus fogalmára. A típusdefiniálás nyelvi keretei. (Modulok, unit, include.)
2.	A modulteszt célja. Egy lehetséges kiinduló pont: a típus(konstrukció) axiomatikus leírása. Egy-két jellegzetes példa a teszteljárás készítésére.
3.	A tömb-típuskonstrukció: lényege (axiómák). Megvalósítása: memóriamodellek, folytonos és láncolt ábrázolás. Speciális és hézagosan kitöltött tömbök.
4.	A verem-típuskonstrukció: lényege (axiómák). Megvalósítása láncolt ábrázolással. Példa.
5.	A sor-típuskonstrukció: lényege (axiómák). Megvalósítása láncolt ábrázolással. Példa.
6.	A kupac mint típuskonstrukció(axiómák). Alkalmazás – kupacrendezés.
7.	A prioritási sor típuskonstrukció: lényege (axiómák). Megvalósítása kupaccal.
8.	A lista-típuskonstrukció: lényege (axiómák). Megvalósítása láncolással. Példa: egy-két programozási tétel újrafogalmazása listára.
9.	A táblázat-típuskonstrukció: lényege (axiómák). Megvalósítás legfontosabb fogalmai: kulcs-transzformációk, táblázatrepresentációk.
10.	A rekurzió-típuskonstrukció: lényege példákon bemutatva, a használat kétféle „filozófiája”. Megvalósítás láncoltan.
11.	A bináris fák-típuskonstrukció: lényege (axiómák). Megvalósítás rekurzióval. Példa: bejárások, egy-két programozási tétel bináris fára.

³ Azok számára, akik nem szerezték meg a december 16-diki vizsga-zárthelyin a vizsgajegyét, azaz nem írták azt meg vagy nem fogadták el a megajánlott jegyet.

12.	A kereső fák-típuskonstrukció lényege: bináris fa speciális műveletekkel. Keresés, törlés, kiegyensúlyozás.
13.	A rendező fák-típuskonstrukció lényege: kereső fa speciális ábrázolásokkal (kitaposott úttal / kupaccal). Keresés, törlés.

A tárgy 2. félévének tematikája:

http://people.inf.elte.hu/szlavi/Adatszerk/Adatszerk_2felev.htm.

A tárgy 1. félévének „hivatalos” weblapja:

http://adatszerk1.elte.hu/?%C3%81tal%C3%A1nos_inform%C3%A1ci%C3%B3k
(még feltöltés alatt).

A tárgy 2. félévének „hivatalos” weblapja:

http://adatszerk2.elte.hu/?%C3%81tal%C3%A1nos_inform%C3%A1ci%C3%B3k
(még feltöltés alatt).