

## A TANTÁRGY ADATLAPJA

### 1. A képzési program adatai

1.1 Felsőoktatási intézmény	Babeş-Bolyai Tudományegyetem
1.2 Kar	Matematika és Informatika
1.3 Intézet	Magyar Matematika és Informatika
1.4 Szakterület	informatika
1.5 Képzési szint	alap
1.6 Szak / Képesítés	Matematika, Matematika-informatika és Informatika

### 2. A tantárgy adatai

2.1 A tantárgy neve	<b>Adatszerkezetek és algoritmusok</b>						
2.2 Az előadásért felelős tanár neve	Ionescu Klára						
2.3 A szemináriumért felelős tanár neve	Ionescu Klára						
2.4 Tanulmányi év	1	2.5 Félév	2	2.6Értékelés módja	vizsga	2.7 Tantárgy típusa	kötelező –alap

### 3. Teljes becsült idő (az oktatási tevékenység féléves óraszama)

3.1 Heti óraszám	3	melyből: 3.2 előadás	2	3.3 szeminárium/labor	1
3.4 Tantervben szereplő összórászám	42	melyből: 3.5 előadás	28	3.6 szeminárium/labor	14
A tanulmányi idő elosztása:					óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					25
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					5
Szemináriumok / laborok, házi feladatok, portofóliók, referátumok, esszék kidolgozása					30
Egyéni készségfejlesztés (tutorálás)					3
Vizsgák					6
Más tevékenységek: .....					
3.7 Egyéni munka össz-órászama	70				
3.8 A félév össz-órászama	112				
3.9 Kreditszám	5 (M), 5 (MI), 5 (I)				

### 4. Előfeltételek (ha vannak)

4.1 Tantervi	Nincs
4.2 Kompetenciabeli	Elemi algoritmusok ismerete

### 5. Feltételek

5.1 Az előadás lebonyolításának feltételei	Táblával és videoprojektossal felszerelt előadó
5.2 A szeminárium / labor lebonyolításának feltételei	Táblával és videoprojektossal felszerelt szemináriumi terem

### 6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<ul style="list-style-type: none"> <li>• Az <b>algoritmus fogalmának</b> megértése, az algoritmusok ábrázolási módozatainak elsajátítása</li> <li>• Az <b>algoritmusok tervezéséhez</b> szükséges készségek kialakítása, a fegyelmezett, logikus és <b>algoritmikus gondolkodás</b> kialakítása</li> <li>• A <b>strukturált programozás, a moduláris programtervezés</b>, valamint a top-down és bottom-up programtervezés alapszabályainak megismerése és elsajátítása</li> <li>• Adott <b>feladatosztályokhoz</b> tartozó feladatok megoldási <b>algoritmusainak</b> és a szükséges <b>adatszerkezeteknek</b> megismerése és elsajátítása: számok, karakterláncok feldolgozása, sorozatok, kétdimenziós tömbök, keresés, összefésülés, rendezés stb.</li> <li>• A megtervezett algoritmusok implementálása <b>egyszerű Pascal/C/C++ programok</b> segítségével</li> <li>• A legfontosabb <b>programozási módszerek</b> (visszalépéses keresés, oszd meg és uralkodj, mohó algoritmusok) elsajátítása és a megfelelő <b>feladatmegoldási készség</b> kialakítása</li> <li>• Helyes, átlátható programozási stílus kialakítása, a <b>dokumentálás</b> alapszabályainak megismerése</li> <li>• <b>Szoftver-komponensek fejlesztése adatstruktúrák, algoritmusok</b>, technikák és fejlett programozási nyelvek felhasználásával</li> <li>• A programozási nyelvek, technikák és módszerek frissítése, úgy hogy a fejlesztett szoftver-komponensek tükrözzék az Információs Technológia és Kommunikáció fejlettségi állapotát</li> <li>• A fejlesztési folyamat szakaszaira vonatkozó követelmények meghatározása, annak érdekében, hogy nagy teljesítményű szoftver-komponenseket kapjunk, modern technológiák alkalmazásával</li> <li>• A szoftverfejlesztési folyamatra jellemző tevékenységek kidolgozása, mennyiségi, minőségi és a gazdasági hatékonyság szempontokat követve</li> <li>• Komplex rendszerekbe beépíthető szoftver-komponensek megvalósítására alkalmas adatstruktúrák, utasítások és probléma-osztályok ismertetése</li> </ul>
Transzverzális kompetenciák	<ul style="list-style-type: none"> <li>• Matematikai problémák megoldása informatikai eszközökkel</li> <li>• A diák elemző és szintetizáló képességének fejlesztése.</li> <li>• A szakmai etika elveinek, normáinak és értékeinek alkalmazása egy felelős, hatékony és igényes munkastratégia kialakításában.</li> <li>• A képzési lehetőségek beazonosítása és a tanulási módszerek és erőforrások hatékony felhasználása a hallgató fejlődésének érdekében.</li> </ul>

## 7. A tantárgy célkitűzései (az elsajátítandó jellemző kompetenciák alapján)

7.1 A tantárgy általános célkitűzése	<ul style="list-style-type: none"> <li>• Modellezési, feladatmegoldói, informatikai szövegértési készségek, jártasságok fejlesztése.</li> <li>• Az <b>alkotókészség fejlesztése</b>.</li> <li>• <b>Egyéni munkára nevelés</b> és a <b>csapatszellem</b> kialakítása.</li> </ul>
7.2 A tantárgy sajátos célkitűzései	<ul style="list-style-type: none"> <li>• Fegyelmezett, logikus és <b>algoritmikus gondolkodás</b> kialakítása.</li> <li>• <b>Absztrakt adattípusok és adatszerkezetek specifikálása, ábrázolása és implementálása</b>.</li> <li>• A <b>szoftvertervezés</b> alapszabályainak megismerése.</li> </ul>

## 8. A tantárgy tartalma

8.1 Előadások	Didaktikai módszerek	Megjegyzések
---------------	----------------------	--------------

<p><b>1. Bevezetés</b></p> <ul style="list-style-type: none"> <li>1.1. Absztrakt adatszerkezetek</li> <li>1.2. Definíciók</li> <li>1.3. Egy egyszerű példa</li> </ul> <p><b>2. Adattípusok és adatszerkezetek</b></p> <ul style="list-style-type: none"> <li>2.1. Adattípusok</li> <li>2.2. Az adattípusok absztrakciójának szintjei</li> <li>2.3. Absztrakt adattípusok <ul style="list-style-type: none"> <li>2.3.1. Az absztrakt adattípusok specifikálása <ul style="list-style-type: none"> <li>A) Egyszerű típusok</li> <li>B) Összetett típusok</li> </ul> </li> <li>2.3.2. Az absztrakt adattípusok használatának előnyei <ul style="list-style-type: none"> <li>A) Egyszerűség</li> <li>B) Egységesség (integritás)</li> <li>C) Az implementáció függetlensége</li> </ul> </li> </ul> </li> <li>2.4. Az elemek és a szerkezet <ul style="list-style-type: none"> <li>2.4.1. Az adatok elemei</li> <li>2.4.2. A szerkezet</li> <li>2.4.3. Lineáris és rendezett adatszerkezetek</li> <li>2.4.4. A címkiszámolásos és a láncolt ábrázolás</li> </ul> </li> <li>2.5. Virtuális és fizikai adattípusok</li> <li>2.6. Statikus változók. Mutatók. Dinamikus változók <ul style="list-style-type: none"> <li>2.6.1. Mutatók</li> </ul> </li> </ul>	<p>1. Előadás</p>	<p>[3] pp: 11-32</p>
<p><b>3. Adatszerkezetek logikai megközelítése</b></p> <ul style="list-style-type: none"> <li>3.1. Általánosságok</li> <li>3.2. Logikai (absztrakt) adattípusok és fizikai ábrázolásuk</li> <li>3.3. Statikus adatszerkezetek <ul style="list-style-type: none"> <li>3.3.1. A tömb (<i>Array</i>)</li> <li>3.3.2. A tétel (<i>Record</i>)</li> <li>3.3.3. A halmaz (<i>Set</i>)</li> </ul> </li> <li>3.4. Félstatikus adatszerkezetek <ul style="list-style-type: none"> <li>3.4.1. A verem (<i>Stack</i>)</li> <li>3.4.2. Várakozási sor (<i>Queue</i>)</li> <li>3.4.3. A hasítótábla (<i>Hashing Table</i>)</li> </ul> </li> <li>3.5. Dinamikus adatszerkezetek <ul style="list-style-type: none"> <li>3.5.1. A lineáris lista (<i>List</i>)</li> <li>3.5.2. A fa (<i>Tree</i>)</li> <li>3.5.3. A hálózat (<i>Network</i>)</li> </ul> </li> </ul> <p><b>4. A tétel</b></p> <ul style="list-style-type: none"> <li>4.1. Rögzített (fix) tételtípus</li> <li>4.2. A tétel absztrakt adattípus</li> <li>4.3. „Változó” (variánsokkal rendelkező) tételek</li> <li>4.4. Tételek C-ben és C++-ban</li> </ul>	<p>2. Előadás</p>	<p>[3] pp: 33-51</p>
<p><b>5. Tömbök</b></p> <ul style="list-style-type: none"> <li>5.1. Definíciók és tulajdonságok</li> <li>5.2. Ábrázolásmódok <ul style="list-style-type: none"> <li>5.2.1. A tömbök deklarációja különböző programozási nyelvekben</li> <li>5.2.2. Helyfoglalás tömbök számára különböző programozási nyelvekben</li> </ul> </li> <li>5.3. A tömb absztrakt adattípus</li> <li>5.4. Implementálás</li> <li>5.5. Tömbszakasz és részsorozat</li> <li>5.6. A tömbök megfeleltetési függvényei</li> <li>5.7. A tömbök paraméterei</li> </ul>	<p>3. Előadás</p>	<p>[3] pp: 51-71</p>

5.8. A tömbök leírása tételekkel		
5.9. Sajátos tömbök 5.9.1. Háromszögű mátrix 5.9.2. Ritka tömbök A) <i>Háromsoros reprezentáció</i> B) <i>Négy soros reprezentáció</i> 5.10. A polinom absztrakt adattípus 5.10.1. A polinomokról általában 5.10.2. Lehetséges ábrázolások A) <i>A polinom ábrázolása tömbök segítségével</i> B) <i>A polinom ábrázolása dinamikusán tárolt tömbök segítségével</i> C) <i>A polinom ábrázolása monomjainak sorozataként</i>	4. Előadás	[3] pp: 72-82
<b>6. Karakterláncok</b> 6.1. Karakterláncokkal végezhető műveletek 6.2. Mintaillesztés 6.2.1. Egyszerű algoritmus A) <i>Mintaillesztés tömbök segítségével implementált karakterláncokkal</i> B) <i>Mintaillesztés dinamikusán tárolt karakterláncokkal</i> 6.2.2. A Knuth, Morris, Pratt algoritmus 6.2.3. A Boyer-Moore algoritmus <b>7. Halmazok</b> 7.1. Alapfogalmak 7.2. Ábrázolásmódok 7.2.1. A halmazok deklarációja különböző programozási nyelvekben 7.3. A halmaz absztrakt adattípus 7.4. A Halmaz osztály	5. Előadás	[3] pp: 83-100  [3] pp: 101-106
<b>8. Listák</b> 8.1. Dinamikus adatszerkezetek 8.2. Egyszeresen (szimplán) láncolt listák 8.2.1. Listák implementálása tömbökkel 8.2.2. Listák implementálása a dinamikus tárban 8.3. A verem 8.3.1. Műveletek 8.3.2. A verem absztrakt adattípus 8.3.3. Ábrázolásmódok és implementálás A) <i>Statikus implementáció</i> B) <i>Dinamikus implementáció</i>	6. Előadás	[3] pp: 107-116
8.4. A várakozási sor 8.4.1. Műveletek 8.4.2. A várakozási sor absztrakt adattípus 8.4.3. Ábrázolásmódok és implementálás A) <i>Statikus implementáció</i> B) <i>Dinamikus ábrázolás és implementálás</i> 8.5. Tetszőleges listák	7. Előadás	[3] pp: 117-130
8.6. Rendezett lista 8.7. Kétszeresen (duplán) láncolt lista 8.8. Körkörös listák 8.9. A dinamikus tárkezelés alkalmazásai 8.9.1. Polinomok ábrázolása láncolt listák segítségével A) <i>Polinomok összeadása</i> B) <i>Polinomok törlése</i>	8. Előadás	[3] pp: 131-148

<p><b>9. Fák</b></p> <p>9.1. Motiváció</p> <p>9.2. Bináris fák</p> <p>9.3. A fák ábrázolásmódjai</p> <p>9.3.1. Ábrázolás a memóriában</p> <p>A) Bináris fák ábrázolása</p> <p>B) Tetszőleges fák ábrázolása</p> <p>9.3.2. Kimeneti ábrázolásmódok (megjelenítések)</p> <p>9.3.3. A bemeneti ábrázolásmódok (a fa beolvasása)</p> <p>9.4. Műveletek</p> <p>9.4.1. Tökéletesen egyensúlyozott bináris fa létrehozása</p> <p>9.4.2. Bejárás</p> <p>A) Mélységi bejárás</p> <p>B) Nem rekurzív bejárás</p> <p>C) Konstans tárigényű bejárás</p> <p>D) Tetszőleges fák szélességi bejárása</p>	<p>9. Előadás</p>	<p>[3] pp: 149-166</p>
<p>9.5. Keresőfák</p> <p>9.5.1. Keresés</p> <p>9.5.2. Beszúrás egy keresőfába</p> <p>9.5.3. Törlés</p> <p>9.5.4. A legkisebb (és a legnagyobb) kulcsú elem megkeresése</p> <p>9.5.6. A következő kulcsú elem megkeresése</p> <p>9.6. AVL-egyensúlyozott bináris fák</p> <p>9.6.1. Specifikáció – AVL fák</p> <p>A) Beszúrás AVL-fákba</p> <p>B) Törlés AVL-fákból</p> <p>9.6.2. Implementálás</p>	<p>10. Előadás</p>	<p>[3] pp: 167-178</p>
<p>9.7. „Piros-fekete” fák</p> <p>9.7.1. A piros-fekete fa absztrakt adattípus</p> <p>9.7.2. Forgatások</p> <p>9.7.3. Beszúrás a piros-fekete fákba</p> <p>9.7.4. Törlés a piros-fekete fákból</p>	<p>11. Előadás</p>	<p>[3] pp: 193-209</p>
<p>9.8. Splay-fák (kifordított fák, S-fák)</p> <p>9.9. Binárisan indexelt fák</p> <p>9.9.1. Bevezetés</p> <p>9.9.2. Naiv algoritmus</p> <p>9.9.3. Parciális összegek</p> <p>9.9.4. Binárisan indexelt fa</p> <p>A) Módosítás</p> <p>B) Lekérdezés</p>	<p>12. Előadás</p>	<p>[3] pp: 214-227</p>
<p><b>10. Kupacok</b></p> <p>10.1. A bináris kupac</p> <p>10.1.1. A kupac tulajdonság fenntartása</p> <p>10.1.2. A kupac építése</p> <p>10.1.3. Kupacrendezés</p> <p>10.2. Elsőbbségi sorok</p>	<p>13. Előadás</p>	<p>[3] pp: 235-245</p>
<p><b>11. Hasító táblák</b></p> <p>11.1. Bevezetés</p> <p>11.1.1. A szimbolikus táblázat osztály</p> <p>11.2. Statikus hasító táblázatok</p> <p>11.2.1. Hasító függvények</p> <p>A) Négyzetek közepe</p> <p>B) Osztás (maradékszámítás)</p>	<p>14. Előadás</p>	<p>[3] pp: 283-298</p>

<p>C) Partícionálás  D) Számjegy vizsgálat  E) Szorzásos módszer  F) Univerzális hasítási technika</p> <p>11.3. A túlcordulás kezelése  11.3.1. Nyitott címzés  11.3.2. Kipróbálási technikák  11.3.3. Láncolás</p> <p>11.4. Dinamikus hasító táblázatok  11.4.1. Alkönyvtárakat felhasználó dinamikus hasítás</p>		
<p>Könyvészet</p> <ol style="list-style-type: none"> <li>1. <b>Cormen T., Leiserson C., Rivest R., Stein, C.</b> – <i>Új algoritmusok</i>, Sclar, Budapest, 2003.</li> <li>2. <b>Horowitz E.</b> – <i>Fundamentals of Data Structures in C++</i>, Computer Science Press, 1995.</li> <li>3. <b>Ionescu K.</b> – <i>Adatszerkezetek</i>, Egyetemi Könyvkiadó, Kolozsvár, 2007</li> <li>4. <b>Preiss B. R.</b> – <i>Data Structures and Algorithms with Object-Oriented Design Patterns in C++</i>, 1997 (<a href="http://www.brpreiss.com/books/opus4/">http://www.brpreiss.com/books/opus4/</a>).</li> <li>5. <b>Rónyai, L., Ivanyos, G., Szabó, R.</b> – <i>Algoritmusok</i>, Typotex, Budapest, 1999.</li> <li>6. <b>Wirth N.</b> – <i>Algorithms + Data Structures = Programs</i>, Prentice Hall Inc., 1976.</li> <li>7. <b>Storer, J.A.</b> – <i>An Introduction to Data Structures and Algorithms</i>, Birkhauser Springer 2002.</li> <li>8. <b>Stubbs D. F., Webre N., W.</b> – <i>Data Structures</i>, Brooks/Cole Publishing Company Monterey, California, 1985.</li> </ol>		
8.2 Szeminárium / Labor	Didaktikai módszerek	Megjegyzések
1. Példák, statikus változókkal, mutatókkal, dinamikus változókkal.	1. Szeminárium	
2. Példák rögzített (fix) tételtípussal (Pascalban és C++-ban), példák változó rekordokkal (Pascalban).	2. Szeminárium	
3. Gyakorlatok a tömbök megfeleltetési függvényeivel, tömbök leírása tételekkel. Az egy- és kétdimenziós tömb AAT implementálása.	3. Szeminárium	
4. A háromszögű mátrix AAT és a ritka tömb AAT implementálása. A polinom ábrázolása tömbök segítségével és monomjainak sorozataként.	4. Szeminárium	
5. Mintaillesztés tömbök segítségével implementált karakterláncokkal, illetve dinamikusán tárolt karakterláncokkal. Knuth, Morris, Pratt algoritmus, Boyer-Moore algoritmus	5. Szeminárium	
6. Listák implementálása statikusan és dinamikusán	6. Szeminárium	
7. Várakozási sorok implementálása statikusan és dinamikusán	7. Szeminárium	
8. Rendezett listák implementálása statikusan és dinamikusán	8. Szeminárium	
9. Parciális vizsga	9. Szeminárium	
10. Tökéletesen egyensúlyozott fák implementálása	10. Szeminárium	
11. Keresőfák implementálása	11. Szeminárium	
12. Piros-fekete fák implementálása	12. Szeminárium	
13. Splay-fák és binárisan indexelt fák implementálása	13. Szeminárium	
14. Kupacok implementálása. Alkalmazások. Az elsőbbségi sor implementálása.	14. Szeminárium	
<p>Könyvészet</p> <ol style="list-style-type: none"> <li>1) <b>Cormen T., Leiserson C., Rivest R., Stein, C.</b> – <i>Új algoritmusok</i>, Sclar, Budapest, 2003.</li> <li>2) <b>Horowitz E.</b> – <i>Fundamentals of Data Structures in C++</i>, Computer Science Press, 1995.</li> <li>3) <b>Ionescu K.</b> – <i>Adatszerkezetek</i>, Egyetemi Könyvkiadó, Kolozsvár, 2007</li> <li>4) <b>Preiss B. R.</b> – <i>Data Structures and Algorithms with Object-Oriented Design Patterns in C++</i>, 1997 (<a href="http://www.brpreiss.com/books/opus4/">http://www.brpreiss.com/books/opus4/</a>).</li> <li>5) <b>Rónyai, L., Ivanyos, G., Szabó, R.</b> – <i>Algoritmusok</i>, Typotex, Budapest, 1999.</li> <li>6) <b>Wirth N.</b> – <i>Algorithms + Data Structures = Programs</i>, Prentice Hall Inc., 1976.</li> </ol>		

**9. A tantárgy tartalmának összhangba hozása az episztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásaival.**

- A tantárgy tartalma megegyezik az egyetemi oktatásban a fontosabb egyetemeken oktatott algoritmusok és programozás bevezető tárgy hagyományos tartalmával.
- A tárgy keretében figyelembe vesszük a számítógép használata nyújtotta lehetőségeket a matematikai problémák vizsgálatában.

**10. Értékelés**

Tevékenység típusa	10.1 Értékelési kritériumok	10.2 Értékelési módszerek	10.3 Aránya a végső jegyben
10.4 Előadás	Alapfogalmak és algoritmusok ismerete	A félév közepén parciális írásbeli vizsga	33 %
10.5 Szeminárium / Labor	Egyéni projekt megvalósítása és bemutatása (helyesség, stílus, dokumentáció, indentálás, tesztelés) 33%	A vizsgaidőszakban írásbeli vizsga	33 %
10.6 A teljesítmény minimumkövetelményei			
<ul style="list-style-type: none"><li>• Az elemi algoritmusok ismerete, a programozási tételek alkalmazása</li><li>• Egyszerű rekurzív algoritmusok ismerete</li><li>• Az elemi adatszerkezetek ismerete, az ezeket feldolgozó algoritmusok implementálása és alkalmazása</li><li>• A fejlett adatszerkezetek ismerete</li><li>• Projekt megvalósítása és megvédése</li></ul>			

Kitöltés dátuma  
2015. május. 5.

Előadás felelőse  
dr. Ionescu Klára

Szeminárium felelőse  
dr. Ionescu Klára

Az intézeti jóváhagyás dátuma

Intézetigazgató,  
Dr. Szenkovits Ferenc, egyet. docens

.....