

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 | Adatszerkezetek és algoritmusok | | | | | | |
| 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 | 2.6 Értékelés módja | kollokv | 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 (MI) |

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

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

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

8. A tantárgy tartalma

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

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

| | | |
|---|--------------------|------------------------|
| <p>9. Fák</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) <i>Bináris fák ábrázolása</i></p> <p>B) <i>Tetszőleges fák ábrázolása</i></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) <i>Mélységi bejárás</i></p> <p>B) <i>Nem rekurzív bejárás</i></p> <p>C) <i>Konstans tárigényű bejárás</i></p> <p>D) <i>Tetszőleges fák szélességi bejárása</i></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) <i>Beszúrás AVL-fákba</i></p> <p>B) <i>Törlés AVL-fákból</i></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) <i>Módosítás</i></p> <p>B) <i>Lekérdezés</i></p> | <p>12. Előadás</p> | <p>[3] pp: 214-227</p> |
| <p>10. Kupacok</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>11. Hasító táblák</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) <i>Négyzetek közepe</i></p> <p>B) <i>Osztás (maradékszámítás)</i></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"> 1. Cormen T., Leiserson C., Rivest R., Stein, C. – <i>Új algoritmusok</i>, Sclar, Budapest, 2003. 2. Horowitz E. – <i>Fundamentals of Data Structures in C++</i>, Computer Science Press, 1995. 3. Ionescu K. – <i>Adatszerkezetek</i>, Egyetemi Könyvkiadó, Kolozsvár, 2007 4. Preiss B. R. – <i>Data Structures and Algorithms with Object-Oriented Design Patterns in C++</i>, 1997 (http://www.brpreiss.com/books/opus4/). 5. Rónyai, L., Ivanyos, G., Szabó, R. – <i>Algoritmusok</i>, Typotex, Budapest, 1999. 6. Wirth N. – <i>Algorithms + Data Structures = Programs</i>, Prentice Hall Inc., 1976. 7. Storer, J.A. – <i>An Introduction to Data Structures and Algorithms</i>, Birkhauser Springer 2002. 8. Stubbs D. F., Webre N., W. – <i>Data Structures</i>, Brooks/Cole Publishing Company Monterey, California, 1985. | | |
| 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"> 1) Cormen T., Leiserson C., Rivest R., Stein, C. – <i>Új algoritmusok</i>, Sclar, Budapest, 2003. 2) Horowitz E. – <i>Fundamentals of Data Structures in C++</i>, Computer Science Press, 1995. 3) Ionescu K. – <i>Adatszerkezetek</i>, Egyetemi Könyvkiadó, Kolozsvár, 2007 4) Preiss B. R. – <i>Data Structures and Algorithms with Object-Oriented Design Patterns in C++</i>, 1997 (http://www.brpreiss.com/books/opus4/). 5) Rónyai, L., Ivanyos, G., Szabó, R. – <i>Algoritmusok</i>, Typotex, Budapest, 1999. 6) Wirth N. – <i>Algorithms + Data Structures = Programs</i>, Prentice Hall Inc., 1976. | | |

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">• Az elemi algoritmusok ismerete, a programozási tételek alkalmazása• Egyszerű rekurzív algoritmusok ismerete• Az elemi adatszerkezetek ismerete, az ezeket feldolgozó algoritmusok implementálása és alkalmazása• A fejlett adatszerkezetek ismerete• Projekt megvalósítása és megvédése | | | |

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

.....