

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	Informatika

2. A tantárgy adatai

2.1 A tantárgy neve	Alapvető algoritmusok						
A tantárgy kódja	MLM5104						
2.2 Az előadásért felelős tanár neve	dr. Ionescu Klára						
2.3 A szemináriumért felelős tanár neve	dr. Sándor Réka						
2.4 Tanulmányi év	1	2.5 Félév	1	2.6 Értékelés módja	vizsga	2.7 Tantárgy típusa	kötelező-alap

3. Teljes becsült idő (az oktatási tevékenység féléves óraszámja)

3.1 Heti óraszám	6	melyből: 3.2 előadás	2	3.3 szeminárium/labor	4
3.4 Tantervben szereplő összóraszám	84	melyből: 3.5 előadás	28	3.6 szeminárium/labor	56
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, portfóliók, referátumok, esszék kidolgozása					25
Egyéni készségfejlesztés (tutorálás)					5
Vizsgák					6
Más tevékenységek:					
3.7 Egyéni munka össz-óraszám	66				
3.8 A félév össz-óraszám	150				
3.9 Kreditszám	6				

4. Előfeltételek (ha vannak)

4.1 Tantervi	Nincs
4.2 Kompetenciabeli	Feladatok kijelentéseinek megértése

5. Feltételek

5.1 Az előadás lebonyolításának feltételei	Táblával és videoprojektorral felszerelt előadó
5.2 A szeminárium / labor lebonyolításának feltételei	Számítógépes terem, a gépeken C/C++

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<ul style="list-style-type: none"> • C1.1 A programozási paradigmák és a specifikus nyelvi mechanizmusok megfelelő leírása, valamint a szemantikai és a szintaktikai vonatkozások közötti különbség meghatározása • C1.3 Megfelelő forráskód fejlesztése egy ismert programozási nyelvben és a komponensek egységes tesztelése adott tervezési specifikáció alapján • C1.4 Alkalmazások tesztelése adott tesztelési terv alapján • C1.5 A programegységek fejlesztése és a kapcsolódó dokumentáció megvalósítása
-----------------------------	--

Transzverzális kompetenciák	<ul style="list-style-type: none"> CT1 A szervezett és hatékony munka szabályainak, a didaktikai-tudományos területhez való felelősségteljes hozzáállás alkalmazása a saját potenciál kreatív értékesítéséhez, a szakmai etika alapelveinek és normáinak tiszteletben tartásával CT3 Hatékony módszerek és technikák használata tanulásra, információszerzésre, kutatásra és a tudásszerzési kapacitások fejlesztésére, egy dinamikus társadalom igényeinek való megfelelésre, román és egy nemzetközi nyelven történő kommunikációra
------------------------------------	---

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. Programozási módszerek elsajátítása és gyakorlása. A szoftvertervezés alapszabályainak megismerése.

8. A tantárgy tartalma

8.1 Előadások	Didaktikai módszerek	Megjegyzések
1. A számítógépes feladatmegoldás lépései <ul style="list-style-type: none"> 1.1. A programozói tevékenység 1.2. A feladatmegoldás lépései számítógépes környezetben 1.3. Alkalmazások minőségi szempontjai 2. Az algoritmusok ábrázolása <ul style="list-style-type: none"> 2.1. Algoritmusok <ul style="list-style-type: none"> 2.1.1. Az algoritmus fogalma 2.1.2. Az algoritmusok leírásánál használt elemek 2.2. Algoritmusok ábrázolása folyamatábrák és pszeudokód nyelvek segítségével 2.3. A strukturált programozás alapelvei <ul style="list-style-type: none"> 2.3.1. Lineáris struktúrák 2.3.2. Elágazási struktúrák 2.3.3. Ismétlődő struktúrák 2.3.4. Az alapstruktúrák jelölése pszeudokódban 	1. Előadás	[3] pp: 13-34
2.4. A feladatok számítógépes megoldásához fűződő általános kérdések <ul style="list-style-type: none"> 2.4.1. Algoritmusok helyessége 2.4.2. Az algoritmus végrehajtásához szükséges idő 2.4.3. Az algoritmus által feldolgozott adatok számára szükséges memória mérete 2.4.4. Algoritmusok egyszerűsége 2.4.5. Optimális algoritmusok 2.4.6. Algoritmusok létezése 	2. Előadás	[3] pp: 35-44
3. Lépések finomítása <ul style="list-style-type: none"> 3.1. Bevezetés és megoldott feladatok 	3. Előadás	[3] pp: 51-60 [4] pp: 9-35
4. Programozási tételek <ul style="list-style-type: none"> 4.1. Egyszerű programozási tételek (összeg és szorzat, döntés, kiválasztás, szekvenciális (lineáris) keresés, megszámlálás, minimum- és maximumkiválasztás, kiválogatás) 	4. Előadás	[3] pp: 73-85 [9] pp: 1-34
4.2. Összetett programozási tételek (Szétválogatás, Sorozat Halmazzá alakítása, Keresztmetszet, Egyesítés, Összefésülés)	5. Előadás	[3] pp: 87-96 [10] pp: 1-41

<p>5. Alprogramok</p> <p>5.1. Bevezetés</p> <p>5.2. Algoritmusok és programok fejlesztési módozatai</p> <p>5.2.1. A top-down típusú (fentről lefele) programozás</p> <p>5.2.2. A bottom-up (lentől felfele) programozás</p> <p>5.2.3. Moduláris algoritmustervezés</p> <p>5.3. A moduláris programozás alapszabályai</p> <p>5.3.1. Moduláris dekompozíció</p> <p>5.3.2. Moduláris kompozíció</p> <p>5.3.3. Modulok tulajdonságai</p> <p>5.3.4. A modularitás alapelvei</p> <p>5.4. Algoritmusok tesztelése</p> <p>5.4.1. A fekete doboz módszere</p> <p>5.4.2. Az átlátszó doboz módszere</p>	<p>6. Előadás</p>	<p>[3] pp: 111-119</p> <p>[4] pp: 42-66</p>
<p>6. Rendezési algoritmusok</p> <p>6.1. Bevezetés</p> <p>6.2. Összehasonlításos rendezési módszerek</p> <p>6.2.1. Buborekrendezés</p> <p>6.2.2. Egyszerű felcseréléses rendezés</p> <p>6.2.3. Válogatásos rendezés</p> <p>6.2.4. Minimum/maximum kiválasztásra épülő rendezés</p> <p>6.2.5. Beszűrő rendezés</p> <p>6.3. Rendezések lineáris időben</p> <p>6.3.1. Leszámláló rendezés (ládarendezés)</p> <p>6.3.2. Számjegyes rendezés</p>	<p>7. Előadás</p>	<p>[3] pp: 131-140</p>
<p>7. Rekurzió</p> <p>7.1. Bevezetés és megoldott feladatok</p> <p>7.2. Közvetlen rekurzió</p>	<p>8. Előadás</p>	<p>[3] pp: 141-153</p>
<p>8. A visszalépéses keresés módszere (backtracking)</p> <p>8.1. Bevezetés</p> <p>8.2. A visszalépéses keresés általános bemutatása</p> <p>8.2.1. Iteratív algoritmus</p> <p>8.2.2. Rekurzív algoritmus</p> <p>8.3. A visszalépéses keresés bővítése</p> <p>8.4. Visszalépéses keresés a síkban</p>	<p>9. Előadás</p>	<p>[3] pp: 161-184</p>
<p>9. Az oszd meg és uralkodj módszer (divide et impera)</p> <p>9.1. Bevezetés</p> <p>9.2. Az oszd meg és uralkodj módszer általános bemutatása</p> <p>9.3. Megoldott feladatok</p>	<p>10. és 11. Előadás</p>	<p>[3] pp: 191-209</p>
<p>10. Mohó algoritmusok (greedy módszer)</p> <p>10.1. Bevezetés</p> <p>10.2. A mohó algoritmus általános bemutatása</p>	<p>12. Előadás</p>	<p>[3] pp: 213-235</p>
<p>10.3. Heurisztikus mohó algoritmusok</p>	<p>13. Előadás</p>	<p>[3] pp: 236-240</p>
<p>11. A dinamikus programozás módszere</p> <p>11.1. A dinamikus programozás módszerének általános bemutatása</p> <p>11.2. Megoldott feladatok</p>	<p>14. Előadás</p>	<p>[1] pp: 323-356</p>

Könyvészet

1. **Cormen T., Leiserson C., Rivest R., Stein, C.** – *Új algoritmusok*, Sclar, Budapest, 2003.
2. **Horowitz E.** – *Fundamentals of Data Structures in C++*, Computer Science Press, 1995.
3. **Ionescu K.** – *Bevezetés az algoritmikába*, Egyetemi Könyvkiadó, Kolozsvár, 2007
4. **Kása Z.** – *Algoritmusok tervezése*, Stúdium Könyvkiadó, Kolozsvár, 1994.
5. **Knuth D. E.** – *A számítógép-programozás művészete, I, II, III kötet*, 1992.
6. **Rónyai, L., Ivanyos, G., Szabó, R.** – *Algoritmusok*, Typotex, Budapest, 1999.
7. **Wirth N.** – *Algorithms + Data Structures = Programs*, Prentice Hall Inc., 1976.

8. Sedgewick R. – <i>Algorithms in C++</i> , Addison-Wesley, 1992.		
9. Szlávi P., Zsakó L. – <i>Elemi programozási tételek</i> , Neumann János Számítógép-tudományi Társaság, Budapest, 2001.		
10. Szlávi P., Zsakó L. – <i>Összetett programozási tételek</i> , Neumann János Számítógép-tudományi Társaság, Budapest, 2001.		
8.2 Szeminárium / Labor	Didaktikai módszerek	Megjegyzések
1. Lineáris struktúrák, Elágazási struktúrák, Ismétlődő struktúrák, Böhm és Jacopini tétele	1.Szeminárium	[1] pp:23-35
2. Elemi algoritmusok (Felcserélés, Maximumérték, Legnagyobb, Palindromszám), érdekes algoritmus: Elnökválasztás	2.Szeminárium	[1] pp:45-50
3. Elemi algoritmusok 2 (Eukleidész algoritmus, Prímszámok, Fibonacci-számok, Háromszög, Fordított szám, Törzstényezők, Konverzió, Gyors hatványozás)	3.Szeminárium	[1] pp:51-72
4. Egyszerű programozási tételek (Összeg és szorzat, Döntés, Kiválasztás, Szekvenciális (lineáris) keresés, Megszámlálás, Minimum- és maximumkiválasztás, Kiválogatás)	4.Szeminárium	[1] pp:73-85 [1] pp:97-100 [3] pp: 1-34
5. Összetett programozási tételek (Szétválogatás, Sorozat halmazra alakítása, Keresztmetszet, Egyesítés, Összefésülés)	5.Szeminárium	[1] pp:87-96 [1] pp:101-104 [4] pp: 1-41
6. Alprogramok (polinomok, mátrixok, determináns)	6.Szeminárium	[1] pp:114-116 [1] pp:120-130
7. Rendező algoritmusok (Buborékredezés, Egyszerű felcseréléses rendezés, Válogatásos rendezés, Minimum/maximum kiválasztásra épülő rendezés, Beszűrő rendezés, Leszámláló rendezés, Számjegyes rendezés)	7.Szeminárium	[1] pp:131-136
8. Parciális vizsga	8.Szeminárium	
9. Rekurzív algoritmusok (Egy szó betűinek megfordítása, Szavak sorrendjének megfordítása, Faktoriális, Legnagyobb közös osztó, Számjegyösszeg, Descartes-szorzat, k elemű részhalmazok, Konverzió, Fibonacci-sorozat, Minden részhalmaz, Partíciók, Halmazpartíciók, Kamatos kamat)	9.Szeminárium	[1] pp:145-160
10. Visszalépéses kereséssel megoldandó feladatok 1: 8 királynő a sakkasztalon, Variációk, Zárójelek, Legrövidebb utak, Játékok, Szűrjektiv függvények, S pénzösszeg kifizetése, Labirintus, Fénykép, Legnagyobb méretű tárgyak	10. Szeminárium	[1] pp:163-174
11. Oszd meg és uralkodj módszerrel megoldandó feladatok: Minimumszámolás, Hatványozás, Bináris keresés, Összefésülésen alapuló rendezés, Gyorsrendezés, Hanoi tornyok, Úszómedence	11. Szeminárium	[1] pp:192-211
12. Mohó algoritmusokkal megoldandó feladatok: Összeg, Az átlagos várakozási idő minimalizálása, Buszmegálló, Autó bérbeadása, Hátizsák, Minimális feszítőfák (Kruskal és Prim), Minimális hosszúságú utak (Dijkstra algoritmus)	12. Szeminárium	[1] pp:216-235
13. Heurisztikus mohó algoritmusokkal megoldandó feladatok: Utazóügynök, Gráfszínezés, Összegkifizetés legkevesebb számú bankjeggyel	13. Szeminárium	[1] pp:236-245
14. Dinamikus programozással megoldandó feladatok: Legrövidebb lánc, Leghosszabb növekvő részsorozat, Autó bérbeadás, Számháromszög, Kukorica, Dominók, Céllövölde	14. Szeminárium	
Könyvészet		
1) Ionescu K. – <i>Bevezetés az algoritmikába</i> , Egyetemi Könyvkiadó, Kolozsvár, 2007		
2) Kása Z. – <i>Algoritmusok tervezése</i> , Stúdium Könyvkiadó, Kolozsvár, 1994.		
3) Szlávi P., Zsakó L. – <i>Elemi programozási tételek</i> , Neumann János Számítógép-tudományi Társaság, Budapest, 2001.		
4) Szlávi P., Zsakó L. – <i>Összetett programozási tételek</i> , Neumann János Számítógép-tudományi Társaság, Budapest, 2001.		

9. A tantárgy tartalmának összhangba hozása az episztémikus 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 és gyakorlati vizsga	33 %
10.5 Szeminárium / Labor	Házi feladatok (helyesség, stílus, dokumentáltság, indentálás, tesztelés) 33%	A vizsgaidőszakban írásbeli és gyakorlati 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• Tudjon megoldani feladatokat visszalépéses kereséssel, oszd meg és uralkodj módszerrel, mohó algoritmussal és a dinamikus programozás módszerével			

Kitöltés dátuma
2020. április 28.

Előadás felelőse
dr. Ionescu Klára

Szeminárium felelőse
dr. Sándor Réka

Az intézeti jóváhagyás dátuma
.....

Intézetigazgató,
Dr. András Szilárd, egyet. docens