

A TANTÁRGY ADATLAPJA

1. A képzési program adatai

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

A tantárgy adatlap közös az informatika és matematika-informatika szakok számára.

2. A tantárgy adatai

2.1 A tantárgy neve		Haladó funkcionális programozás (hu) / Metode avansate de programare funcțională (ro) / Advanced methods in functional programming (en)					
2.2 Az előadásért felelős tanár neve			Horváth Zoltán				
2.3 A szemináriumért felelős tanár neve			Csató Lehel				
2.4 Tanulmányi év	3	2.5 Félév	2	2.6. Értékelés módja	kollokvium	2.7 Tantárgy típusa	választható – szak

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 össz-óraszám	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					48
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					26
Szemináriumok / laborok, házi feladatok, portfóliók, referátumok, esszék kidolgozása					14
Egyéni készségfejlesztés (tutorálás)					16
Vizsgák					4
Más tevékenységek:					-
3.7 Egyéni munka össz-óraszama	108				
3.8 A félév össz-óraszama	150				
3.9 Kreditszám	6				

4. Előfeltételek (ha vannak)

4.1 Tantervi	Nincs
4.2 Kompetenciabeli	Alap-programozási készségek, matematikai logika alaptudás, elemi funkcionális programozási ismeretek

5. Feltételek (ha vannak)

5.1 Az előadás lebonyolításának feltételei	<ul style="list-style-type: none">• Az előadásokhoz video-projektor szükséges.• A példák kifejtéséhez és illusztráció számára tábla szükséges.
5.2 A szeminárium / labor lebonyolításának feltételei	<ul style="list-style-type: none">• A laboratóriumi órák alatt a diákok a számítógépet, az oktató a táblát használja;• A szemináriumok során példákat oldunk meg, melyekhez tábla szükséges.

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<ul style="list-style-type: none">• A funkcionális programmodell jellemzői, a magasabb-rendű függvények használata,• Betekintés a modern funkcionális programnyelvek jellegzetességeibe, a típusvezetés fogalma,• A típusosztályok bemutatása, új típusok írása,• Hatékony magas szintű funkcionális programmodulok írása.
Transzverzális kompetenciák	<ul style="list-style-type: none">• Feladatmegoldó készségek fejlesztése,• Hatékony programozási módszerek elsajátítása,• Programok/algorithmok bonyolultságának a becslése és csökkentése.

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

7.1 A tantárgy általános célkitűzése	<ul style="list-style-type: none">• A funkcionális nyelvek más gondolkodási és programmegoldó sémákat követelnek, ezen programmegoldó stratégiák tanulása és elsajátítása a cél.• A funkcionális programmodell alkalmazása modern algoritmusokban,• A típusokról történő gondolkodás elmélyítése, a típusok hasznának a tudatosítása.
7.2 A tantárgy sajátos célkitűzései	<p>Az előadások során a következő témákat érintjük:</p> <p>A funkcionális programmodell: rekurzív algoritmusok, rekurzív algoritmusok gépi átírása, típusvezetés</p> <p>A típusrendszerek definíciói: A Haskell típusrendszere, a Haskell/Clean típusvezetése, a template fogalmának a helyettesítése, a Scala típusai, a Scala típusvezetési mechanizmusa, mint haladó (?) Java kód írásának egy módja</p>

8. A tantárgy tartalma

8.1 Előadás	Didaktikai módszerek	Megjegyzések
-------------	----------------------	--------------

<p>1. hét - A funkcionális paradigma ismételése, különböző funkcionális programnyelvek bemutatása, összehasonlítások, funkcionális jellemzők a gyakran használt programnyelvekben.</p>	<p>tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás, teszt-feladatok oldása</p>	
<p>2. hét - A Haskell nyelv elemei, típusok fogalma, a Haskell programozói környezet, rekurencia.</p>		
<p>3. hét - Listakezelés Haskell-ben listák ábrázolása, listakonstruktorok, listaműveletek alkalmazása,</p>		
<p>4.hét - A magasabb-rendű függvények fogalma, a map, a filter, az until függvények.</p>		
<p>5. hét - Operátorok Haskell-ben, függvénytípusok, típusok levezetése, lambda-függvények.</p>		
<p>6. hét - Nagyhatékonyságú programok írása, bonyolultság csökkentése, a zip, zipwith, függvények.</p>		
<p>7. hét - A foldr és foldl függvények, a map+fold = map-reduce paradigma. A pipeline programmodell és alkalmazásai.</p>		
<p>8. hét - A lambda-kalkulus, mint függvények/programok értékének a kiszámítása.</p>		
<p>9. hét - Haskell példák a lambda-kalkulusra, elméleti kitekintő.</p>		
<p>10. hét - A lambda-kalkulus és a matematikai programmodell. A programmodulok könnyű tesztelése és helyesség-bizonyítása.</p>		
<p>11. hét - Érdekes feladatok, kitekintések más programnyelvek felé. A Scala, mint funkcionális programnyelv.</p>		
<p>12. hét - Kitekintések más programnyelvek felé: az F# programozási nyelv.</p>		
<p>13. hét - Feladatok bemutatása és feladatok megoldása, a funkcionális programmodell használatával.</p>		
<p>14. hét - Ismétlő feladatok, vizsgafeladatok megbeszélése.</p>		
<p>Könyvészet</p> <p>[1]. Bird R (2011) Pearls of Functional Algorithm Design,</p> <p>[2]. Bird R. (2015) Thinking Functionally with Haskell, Cambridge University Press</p> <p>[3]. Reede, C. (1989) Elements of Functional Programming, Addison Wesley.</p> <p>[4]. Petricek T, Skeet J (2009) Real-World Functional Programming, With examples in F# and C#, Manning Publications.</p> <p>[5]. Field A. (1988) Functional Programming, Addison Wesley, New York.</p> <p>[6]. Horváth Zoltán (ELTE programnyelvek tanszék) Funkcionális programozás előadása.</p> <p>[7].Graham Hutton (2007) Programming in Haskell, Cambridge University Press.</p> <p>[8]. Miran Lipovaca (2011) Learn you a Haskell for Great Good, No Starch Press, San Francisco.</p>		
<p>8.2 Szeminárium / Labor</p>	<p>Didaktikai módszerek</p>	<p>Megjegyzések</p>

Labor		
1. hét Az első feladatcsoport kitűzése (Haskell alapok, nyelvi konstrukciók), a funkcionális stílus gyakorlása.	Munkáltatás, feladatok kitűzése és megoldása. Tesztelések különböző esetekre.	
2. hét Feladatok ellenőrzése, második feladatcsoport kitűzése (MAP, FOLD, FILTER)		
3. hét Második feladatcsoport megbeszélése		
4. hét A második feladatcsoport ellenőrzése, a harmadik bemutatása (Hajtogatások alkalmazása).		
5. hét A harmadik feladatcsoport ellenőrzése, a negyedik bemutatása és megbeszélése (típusosztály, definíciók).		
6. hét A harmadik feladatcsoport bemutatása.		
7. hét Feladatok ellenőrzése.		

9. Az episztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásainak összhangba hozása a tantárgy tartalmával.

- Az előadás követi a neves angliai (University College London, St.Andrews-i Egyetem) és amerikai egyetemek előadásainak a struktúráját (MIT, Stanford).
- A feladatokat a fenti egyetemek – Stanford, MIT, UCL – kurzusainak a segítségével állítottuk össze.

10. Értékelés

Tevékenység típusa	10.1 Értékelési kritériumok	10.2 Értékelési módszerek	10.3 Aránya a végső jegyben
10.4 Előadás	Írásbeli vizsga a félév végén	Írásbeli vizsga	60%
10.5 Labor	Programozási feladatok bemutatása és/vagy szemináriumi bemutató	A megoldások pontozása	40%
	Opcionális feladatok	A megoldások pontozása	+20%

10.6 A teljesítmény minimumkövetelményei

Az elvárt minimális tudás:

- A magasabb rendű függvények használata.
- A funkcionális kompozíció használata,
- A típusok ismerete a Haskell programnyelvben.

Az átmenő jegy feltételei:

- Az évközi tevékenység során a pontok 60%-ának az összegyűjtése (24 pont; kizáró jellegű).
- A vizsgán legkevesebb a pontok felének összegyűjtése (kizáró jellegű).
- A végső pontszám minimálisan 70 (=5; kizáró jellegű).

Kitöltés dátuma

2019.10.22

Előadás felelőse

prof. dr. Horváth Zoltán

Labor felelőse

prof. dr. Csató Lehel

Az intézeti jóváhagyás dátuma

.....

Intézetigazgató

dr. András Szilárd, egyetemi docens