

## A TANTÁRGY ADATLAPJA

### 1. A képzési program adatai

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

A tantárgy adatlap közös az informatika, matematika-informatika, illetve az információ-mérnöki szakok számára. A kreditszámok különbözősége az előzetes tudás különbözőségére vezethető vissza (a matematika-informatika szakosok nem tanulnak logikai programozást, mely egyik alapja a mesterséges intelligenciának).

### 2. A tantárgy adatai

2.1 A tantárgy neve		<b>Funkcionális nyelvek, típusrendszerek és programozási módszerek / Sisteme de tipuri si metode de programare in limbaje functionale / Functional type systems and programming paradigms</b>					
2.2 Az előadásért felelős tanár neve		<b>Horváth Zoltán</b>					
2.3 A szemináriumért felelős tanár neve		<b>Csató Lehel</b>					
2.4 Tanulmányi év	<b>3</b>	2.5 Félév	<b>2</b>	2.6. Értékelés módja	<b>kollokvium</b>	2.7 Tantárgy típusa	<b>választható – szak</b>

### 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	<b>42</b>	melyből: 3.5 előadás	<b>28</b>	3.6 szeminárium/labor	<b>14</b>
A tanulmányi idő elosztása:					Óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					<b>32</b>
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					<b>26</b>
Szemináriumok / laborok, házi feladatok, portfóliók, referátumok, esszék kidolgozása					<b>14</b>
Egyéni készségfejlesztés (tutorálás)					<b>7</b>
Vizsgák					<b>4</b>
Más tevékenységek:					-
3.7 Egyéni munka össz-óraszama					<b>83</b>
3.8 A félév össz-óraszama					<b>125</b>
3.9 Kreditszám					<b>5</b>

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

## 6. Elsajátítandó jellemző kompetenciák

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

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

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

## 8. A tantárgy tartalma

8.1 Előadás	Didaktikai módszerek	Megjegyzések
<b>1. hét</b> A funkcionális paradigma ismertetése, különböző funkcionális programnyelv bemutatása, összehasonlítások, funkcionális jellemzők a gyakran használt programnyelvekben.	tanári magyarázat, rávezetés, interaktív kódírás	
<b>2. hét</b>	tanári magyarázat,	

A Haskell nyelv elemei, típusok fogalma, a Haskell programozói környezet, rekurencia.	rávezetés, interaktív kódírás, munkáltatás	
<b>3. hét</b> Listakezelés Haskell-ben listák ábrázolása, listakonstruktorok, listaműveletek alkalmazása,	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás	
<b>4.hét</b> A magasabb-rendű függvények fogalma, a <b>map</b> , a <b>filter</b> , az <b>until</b> függvények.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás	
<b>5. hét</b> Operátorok Haskell-ben, függvénytípusok, típusok levezetése, lambda-függvények.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás, feladatok kitűzése	
<b>6. hét</b> Nagyhatékonyságú programok írása, bonyolultság csökkentése, a <b>zip</b> , <b>zipwith</b> , függvények.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás, teszt-feladatok	
<b>7. hét</b> A <b>foldr</b> és <b>foldl</b> függvények, a <b>map+fold</b> = map-reduce paradigma. A pipeline programmodell és alkalmazásai.	tanári magyarázat, rávezetés, munkáltatás	
<b>8. hét</b> A lambda-kalkulus, mint függvények/programok értékének a kiszámítása.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás	
<b>9. hét</b> Haskell példák a lambda-kalkulusra, elméleti kitekintő.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás	
<b>10. hét</b> A lambda-kalkulus és a matematikai programmodell. A programmodulok könnyű tesztelése és helyesség-bizonyítása.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás	
<b>11. hét</b> Érdekes feladatok, kitekintések más programnyelvek felé. A Scala nyelv, mint funkcionális programnyelv.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás	
<b>12. hét</b> Kitekintések más programnyelvek felé: az F# programozási nyelv.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás	
<b>13. hét</b> Feladatok bemutatása és feladatok megoldása, a funkcionális programmodell használatával.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás, teszt-feladatok	
<b>14. hét</b> Ismétlő feladatok, vizsgafeladatok megbeszélése.	munkáltatás, teszt-feladatok	
<b>Könyvészet</b>		
[1]. Bird R (2011) Pearls of Functional Algorithm Design,		
[2]. Bird R. (2015) Thinking Functionally with Haskell, Cambridge University Press		
[3]. Reede, C. (1989) Elements of Functional Programming, Addison Wesley.		
[4]. Petricek T, Skeet J (2009) Real-World Functional Programming, With examples in F# and C#, Manning		

Publications.

[5]. Field A. (1988) Functional Programming, Addison Wesley, New York.

[6]. Horváth Zoltán (ELTE programnyelvek tanszék) Funkcionális programozás előadása.

[7].Graham Hutton (2007) Programming in Haskell, Cambridge University Press.

[8]. Miran Lipovaca (2011) Learn you a Haskell for Great Good, No Starch Press, San Francisco.

8.2 Szeminárium / Labor	Didaktikai módszerek	Megjegyzések
<b>Labor</b>		
<b>1. hét</b> Az első feladatcsoport kitűzése (Haskell alapok, nyelvi konstrukciók), a funkcionális stílus gyakorlása.	munkáltatás, individuális feladatok	
<b>2. hét</b> Feladatok ellenőrzése, második feladatcsoport kitűzése (MAP, FOLD, FILTER)	munkáltatás, individuális feladatok	
<b>3. hét</b> Második feladatcsoport megbeszélése	munkáltatás, individuális feladatok	
<b>4. hét</b> A második feladatcsoport ellenőrzése, a harmadik bemutatása (Hajtogatások alkalmazása).	munkáltatás, individuális feladatok	
<b>5. hét</b> A harmadik feladatcsoport ellenőrzése, a negyedik bemutatása és megbeszélése (típusosztály, definíciók).	munkáltatás, individuális feladatok	
<b>6. hét</b> A harmadik feladatcsoport bemutatása.	munkáltatás, individuális feladatok	
<b>7. hét</b> Feladatok ellenőrzése.	munkáltatás, individuális feladatok	

### 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 – publikus kurzus-oldalai 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

2015.04.29

Előadás felelőse

Horváth Zoltán

Labor felelőse

Csató Lehel

Az intézeti jóváhagyás dátuma

.....

Intézetigazgató

Szenkovits Ferenc