

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		Prof. dr. Horváth Zoltán					
2.3 A szemináriumért felelős tanár neve		drd. Pável Szaboles					
2.4 Tanulmányi év	3	2.5 Félév	2	2.6. Értékelés módja	kollokvium	2.7 Tantárgy típusa	választható – szak
2.8 A tantárgy kódja		MLM5047					

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 a másodéves „Logikai és Funkcionális Programozás” tárgyból.
--	---

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 típuslevezetés 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/algorithmusok 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	Az előadások során a következő témákat érintjük:

	<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
1. hét - A funkcionális paradigma ismételése, különböző funkcionális programnyelvek bemutatása, összehasonlítások, funkcionális jellemzők a gyakran használt programnyelvekben.	tanári magyarázat, rávezetés, interaktív kódírás, munkáltatás, teszt-feladatok oldása	
2. hét - A Haskell nyelv elemei, típusok fogalma, a Haskell programozói környezet, rekurencia.		
3. hét - Listakezelés Haskell-ben listák ábrázolása, listakonstruktorok, listaműveletek alkalmazása,		
4.hét - A magasabb-rendű függvények fogalma, a map , a filter , az until függvények.		
5. hét - Operátorok Haskell-ben, függvénytípusok, típusok levezetése, lambda-függvények.		
6. hét - Nagyhatékonyságú programok írása, bonyolultság csökkentése, a zip , zipwith , függvények.		
7. hét - A foldr és foldl függvények, a map+fold = map-reduce paradigma. A pipeline programmodell és alkalmazásai.		
8. hét - A lambda-kalkulus, mint függvények/programok értékének a kiszámítása.		
9. hét - Haskell példák a lambda-kalkulusra, elméleti kitekintő.		
10. hét - A lambda-kalkulus és a matematikai programmodell. A programmodulok könnyű tesztelése és helyesség-bizonyítása.		
11. hét - Érdekes feladatok, kitekintések más programnyelvek felé. A Scala, mint funkcionális programnyelv.		
12. hét - Kitekintések más programnyelvek felé: az F# programozási nyelv.		
13. hét - Feladatok bemutatása és feladatok megoldása, a funkcionális programmodell használatával.		
14. hét - Ismétlő feladatok, ellenőrzőfeladatok megbeszélése.		
Könyvészet		

- [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
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 epiztemikus 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	Kollokvium a félév végén	Írásbeli	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: <ul style="list-style-type: none">• A magasabb rendű függvények használata.• A funkcionális kompozíció használata,• A típusok ismerete a Haskell programnyelvben. Az átmenő jegy feltételei: <ul style="list-style-type: none">• Az évközi tevékenység során a pontok 60%-ának az összegyűjtése (24 pont; kizáró jellegű).• A kollokviumon a pontok legalább felének az összegyűjtése (kizáró jellegű).• A végső pontszám minimálisan 70 (=5; kizáró jellegű).			

Kitöltés dátuma

2022.04.04.

Előadás felelőse

Dr. Horváth Zoltán egyetemi tanár

Labor felelőse

drd. Pével Szabolcs

Az intézeti jóváhagyás dátuma

2022.04.22.

Intézetigazgató

Dr. András Szilárd, egyetemi docens