

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	Posztgraduális
1.6 Szak / Képesítés	Posztgraduális informatika és szoftverfejlesztés képzés

2. A tantárgy adatai

2.1 A tantárgy neve		Szoftverrendszerek verifikációja és validációja, automatizált tesztelés					
2.2 Az előadásért felelős tanár neve			Dr. Csató Lehel				
2.3 A szemináriumért felelős tanár neve			Dr. Csató Lehel				
2.4 Tanulmányi év	1	2.5 Félév	2	2.6. Értékelés módja	vizsga	2.7 Tantárgy típusa	Kötelező
2.8 Tantárgy azonosító		MLM5131					

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

3.1 Heti óraszám	4	Melyből: 3.2 előadás	2	3.3 szeminárium/labor	2
3.4 Tantervben szereplő össz-óraszám	40	Melyből: 3.5 előadás	20	3.6 szeminárium/labor	20
A tanulmányi idő elosztása					óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					30
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					15
Szemináriumok / laborok, házi feladatok, portofóliók, referátumok, esszék kidolgozása					30
Egyéni készségfejlesztés (tutorálás)					6
Vizsgák					4
Más tevékenységek:					0
3.7 Egyéni munka össz-óraszama		85			
3.8 A félév össz-óraszama		125			
3.9 Kreditszám		5			

4. Előfeltételek (ha vannak)

4.1 Tantervi	<ul style="list-style-type: none"> Programozás alapok és algoritmika, Programozás és adatszerkezetek, objektumorientált programozás, haladó programozási módszerek és környezetek
4.2 Kompetenciabeli	<ul style="list-style-type: none"> Programozási ismeretek egy magas szintű objektumorientált programozási nyelvben

5. Feltételek (ha vannak)

5.1 Az előadás lebonyolításának feltételei	<ul style="list-style-type: none"> Videoprojektorral és táblával felszerelt előadóterem
5.2 A szeminárium/labor	<ul style="list-style-type: none"> Számítógépekkel felszerelt laborterem, a gépeken a Java

lebonyolításának feltételei	programozásra megfelelő szoftverekkel (fejlesztői csomag és környezet)
-----------------------------	--

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<p>C1.1. Programozási paradigmák és nyelvspecifikus mehanizmusok kielégítő leírása valamint a szemantika és szintaktika közötti különbségek azonosítása.</p> <p>C1.2. Meglévő szoftver alkalmazás magyarázata különböző absztrakciós szinteken (architektúra, csomagok, osztályok, metódusok), a megszerzett alapismereteket felhasználva.</p> <p>C1.3. Forráskód megfelelő szintű készítése és komponensek unit-tesztelése egy ismert programnyelven, adott feladatspecifikáció alapján.</p> <p>C1.4. Alkalmazások tesztelése adott tesztelési terv alapján</p> <p>C1.5. A progamegységek fejlesztése és a kapcsolódó dokumentáció megvalósítása</p>
Transzverzális kompetenciák	<p>CT1. Hatékony és precíz munkavégzés szabályainak alkalmazása, az oktatási és tudományos területtel szembeni felelősségteljes hozzáállás tanúsítása egy adott helyzetben rejlő személyes potenciál optimális és kreatív kiaknázása érdekében a szakmai etikai normák és irányelvek betartása mellett.</p> <p>CT3. Hatékony tanulási, informálódási és kutatási technikák használata valamint tudás felhasználási képességek, a dinamikus társadalom kívánalmaihoz való alkalmazkodóképesség és román valamint egy nemzetközi nyelven történő kommunikációs készségek fejlesztése.</p>

7. A tanárgy célkitűzései (az elsajátítandó kompetenciáknak megfelelően)

7.1 A tantárgy általános célkitűzései	<ul style="list-style-type: none"> • Algoritmusok teljes és részleges helyességének megértése • Algoritmusok helyességének bizonyítása • Szoftvertesztelési és ellenőrzési módszerek megismerése • Helyes algoritmusok tervezése a specifikáció alapján • A gyakorlati szempontoknak megfelelő programozási stílus elsajátítása
7.2 A tantárgy sajátos célkitűzései	<ul style="list-style-type: none"> • Statikus verifikációs módszerek elsajátítása, kódátvizsgálás • Tesztesetek megtervezése • A tesztelési folyamat menedzselésére használt eszközök megismerése • Tesztelési módszerek elsajátítása • Automatizált tesztek implementálása

8. A tantárgy tartalma

8.1 Előadás	Didaktikai módszerek	Megjegyzések
1. Szoftverek verifikációjával és validációjával (V&V) kapcsolatos alapfogalmak áttekintése	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
2. Statikus V&V módszerek: kódátvizsgálás, code review és code inspection módszerek, kapcsolódó eszközök, automatizált statikus elemzés	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
3. Formális V&V módszerek: algoritmusok	<ul style="list-style-type: none"> • Magyarázat 	

helyességének bizonyítása, a Floyd és Hoare módszerek, a Cleanroom módszer	<ul style="list-style-type: none"> • Párbeszéd • Példák • Interaktív bemutatók 	
4. Dinamikus V&V módszerek: tesztelési módszerek áttekintése	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
5. Automatizált tesztelés (JUnit, Hamcrest), tesztek elkülönítése és mockign keretrendszerek (Mockito)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
6. Automatizált integrációs tesztelés (Arquillian)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
7. Automatizált UI tesztelés (Selenium)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
8. Automatizált UI tesztelés: mobil applikációk tesztelésére használható keretrendszerek	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
9. V&V și CI/CD (Continuous Integration/Deployment), alkalmazások monitorizálására használható eszközök	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
10. V&V a QA (Quality Assurance/Minőségbiztosítás) kontextusában, ismétlés	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	

Könyvészet

1. Frentiu, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
2. R. S. Pressman, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010
3. L. Crispin, J. Grecory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
4. M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
5. K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
6. J. P. Katoen, Principles of Model Checking, MIT Press, May 2008
7. R. Patton, Software Testing, Sams Publishing, 2005
8. Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004
9. I. Bernstein, Practical software testing, Springer, 2002
10. Balanescu T., Corectitudinea programelor, Editura tehnica, Bucuresti 1995.
11. Morgan, C., Programing from Specifications, Prentice Hall, NewYork, 1990.

8.2 Szeminárium / labor	Didaktikai módszerek	Megjegyzések
1. Gyakorlatok: manuális tesztelés és debugging	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
2. Gyakorlatok: code review & code inspection	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd 	

	<ul style="list-style-type: none"> • Példák • Interaktív bemutatók 	
3. Automatizált statikus elemzésre használható eszközök kipróbálása (pl. SonarQube)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
4. Gyakorlatok: tesztesetek tervezése	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
5. Automatizált unit tesztek létrehozása (JUnit+Hamcrest+Mockito)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
6. Automatizált integrációs tesztek létrehozása (Arquillian)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
7. Automatizált UI tesztek létrehozása (Selenium)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
8. Automatizált tesztek létrehozása mobil platformokon (Espresso/Apium)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
9. Az elsajátított V&V módszereket beépítő CI/CD pipeline-ok létrehozása (automatizált tesztek futtatása, automatikus kitelepítés tesztszerverekre, automatizált statikus elemzések automatikus futtatása)	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	
10. Tesztek dokumentálása, a projektmenedzsment/ issue tracking rendszerek használata a V&V / QA kontextusában	<ul style="list-style-type: none"> • Magyarázat • Párbeszéd • Példák • Interaktív bemutatók 	

Könyvészet

1. Frentiu, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
2. R. S. Pressman, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010
3. L. Crispin, J. Grecory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
4. M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
5. K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
6. J. P. Katoen, Principles of Model Checking, MIT Press, May 2008
7. R. Patton, Software Testing, Sams Publishing, 2005
8. Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004
9. I. Bernstein, Practical software testing, Springer, 2002
10. Balanescu T., Corectitudinea programelor, Editura tehnica, Bucuresti 1995.
11. Morgan, C., Programing from Specifications, Prentice Hall, New York, 1990.

9. A tantárgy tartalmának összhangba hozása az episztemikus közösség képviselő, a szakmai egyesületek és a szakterület reprezentatív munkáltatóinak elvárásaival

- A tantárgy követi az IEEE és ACM informatikai kurrikulumát
- A tantárgy része a fontos romániai és külföldi egyetemek oktatási programjának
- A tananyagot a szoftverfejlesztő cégek többsége kiemelten fontosnak tartja a programozói képességek megszerzésének szempontjából

10. Értékelés

Tevékenység típusa	10.1 Értékelési kritériumok	10.2 Értékelési módszerek	10.3 Arány a végső jegyben
10.4 Előadás	Az elsajátított ismeretek, azok felidézésének helyessége és pontossága. V&V módszerek alkalmazásának képessége.	Írásbeli vizsga	50%
10.5 Szeminárium/labor	V&V módszerek gyakorlati alkalmazásának képessége	Gyakorlati vizsga	50%
10.6 A sikeres vizsga minimális követelményei			
<ul style="list-style-type: none">• Minden részeredménynek átmenő osztályzatnak (min. 5) kell lennie			

Az előadás felelőse

A szeminárium felelőse

Dr. Csató Lehel, egyetemi professzor

Dr. Csató Lehel, egyetemi professzor

Kitöltés dátuma

Az intézeti jóváhagyás dátuma

Intézetvezető

Dr. András Szilárd Károly, egyetemi docens