

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	Alapképzés
1.6 Szak / Képesítés	Informatika

2. A tantárgy adatai

2.1 A tantárgy neve	Tervezői és fejlesztői környezetek						
2.2 Az előadásért felelős tanár neve	Dr. Sulyok Csaba, tanársegéd						
2.3 A szemináriumért felelős tanár neve	Dr. Sulyok Csaba, tanársegéd						
2.4 Tanulmányi év	3	2.5 Félév	5	2.6. Értékelés módja	Vizsga	2.7 Tantárgy típusa	Kötelező - szaktárgy
2.8 A tantárgy kódja	MLM5015						

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	56	Melyből: 3.5 előadás	28	3.6 szeminárium/labor	28
A tanulmányi idő elosztása:					óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					17
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					17
Szemináriumok / laborok, házi feladatok, portofóliók, referátumok, esszék kidolgozása					32
Egyéni készségfejlesztés (tutorálás)					-
Vizsgák					3
Más tevékenységek:					-
3.7 Egyéni munka össz-óraszama	69				
3.8 A félév össz-óraszama	125				
3.9 Kreditszám	5				

4. Előfeltételek (ha vannak)

4.1 Tantervi	<ul style="list-style-type: none"> ● Bevezetés a programozásba és algoritmikába ● Haladó programozási módszerek ● Adatbázisok
4.2 Kompetenciabeli	<ul style="list-style-type: none"> ● Java programozási nyelv középszintű ismerete ● Webprogramozási alapok

5. Feltételek (ha vannak)

5.1 Az előadás lebonyolításának feltételei	<ul style="list-style-type: none">• Videoprojektorral felszerelt előadóterem• Megbízható hálózati csatlakozási lehetőség
5.2 A szeminárium / labor lebonyolításának feltételei	<ul style="list-style-type: none">• Számítógépekkel és megbízható vezeték nélküli hálózattal felszerelt laborterem• Áramellátási lehetőség hordozható számítógépeknek

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<p>C1.1. Programozási paradigmák és nyelvspecifikus mechanizmusok kielégítő leírása, valamint a szemantikai és szintaktikai közötti különbségek azonosítása.</p> <p>C1.3. Forráskód megfelelő szintű készítése és komponensek unit-tesztelése egy ismert programnyelven, adott feladatspecifikáció alapján.</p> <p>C2.1. A szoftverrendszerek megfelelő fejlesztési módszereinek beazonosítása</p> <p>C2.3. Módszerek, specifikációs mechanizmusok és fejlesztési környezetek alkalmazása az informatikai alkalmazások fejlesztéséhez</p>
Transzverzális kompetenciák	<p>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</p> <p>CT2 Interdiszciplináris csoportban szervezett tevékenységek hatékony lebonyolítása és az interperszonális kommunikáció, a különféle csoportokhoz való viszony és együttműködés empátiás képességének fejlesztése</p> <p>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</p>

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

7.1 A tantárgy általános célkitűzése	<ul style="list-style-type: none">• Komplex szoftverrendszerek tervezésével és megvalósításával kapcsolatos ismeretek elmélyítése, rendszerezése• Osztott rendszerek fejlesztésével kapcsolatos ismeretek elsajátítása• Többretegű szoftverarchitektúrák fejlesztése, szolgáltatás-orientált architektúrák tervezése
7.2 A tantárgy sajátos célkitűzései	<ul style="list-style-type: none">• Modern build rendszerek alapvető funkcionalitásainak megismerése• Az MVC minta ismertetése, thin és rich kliensek megkülönböztetése• Perzisztenciával kapcsolatos tervezési minták és magasabb szintű technológiák megismerése: ORM (Object Relational Mapping), JPA (Java Persistence API) specifikáció, JPA-t implementáló keretrendszerek (pl. Hibernate)• Enterprise keretrendszerek (Java EE, Spring) ismertetése• Az Inversion of Control és Dependency Injection tervezési minták ismertetése.

8. A tantárgy tartalma

8.1. Előadás	Didaktikai módszerek	Megjegyzések
1. Modern build rendszerek <ul style="list-style-type: none"> • Verziókövetés: git, git flow • Build és függőségkezelési rendszerek szükségessége • Legacy build rendszerek: Make és Ant 	Előadás, magyarázat, példák	
2. Modern build rendszerek <ul style="list-style-type: none"> • Maven: működési elv, telepítés-leíró, konvencionális struktúra, függőségek, projekt életciklus, pluginok • Multimodul Maven projektek 	Előadás, magyarázat, példák	
3. Modern build rendszerek és naplózás <ul style="list-style-type: none"> • Gradle: működési elv, telepítés-leíró, függőségek, taskok, pluginok • Minőségbiztosítás build rendszerekkel • Naplózás: slf4j, logback 	Előadás, magyarázat, példák	
4. Java-alapú webalkalmazások <ul style="list-style-type: none"> • Java webkonténerek: Tomcat • A Servlet API • HTTP kérések feldolgozása, session-kezelés 	Előadás, magyarázat, példák	
5. Model-View-Controller minta <ul style="list-style-type: none"> • Az MVC minta leírása, thin és rich kliensek megkülönböztetése • MVC megvalósítása Java-ban: JSP, Thymeleaf 	Előadás, magyarázat, példák	
6. Többrétegű architektúrák és adatelérési réteg <ul style="list-style-type: none"> • Perzisztencia alapok: JDBC • Az Abstract DAO Factory tervezési minta • Connection pooling 	Előadás, magyarázat, példák	
7. Java SE kiegészítések <ul style="list-style-type: none"> • Konfigurációs állományok: Properties • Profilozás • Nemzetköziesítés (i18n) 	Előadás, magyarázat, példák	
8. Futási idejű befolyásolás <ul style="list-style-type: none"> • Reflection és a Java Reflection API • Biztonsággal kapcsolatos kérdések 	Előadás, magyarázat, példák	
9. Magasszintű perzisztencia <ul style="list-style-type: none"> • Objektum-relációs leképezés (ORM) • A Java Persistence API (JPA) s ennek implementációi (Hibernate) 	Előadás, magyarázat, példák	
10. Magasszintű perzisztencia <ul style="list-style-type: none"> • Leképezési nyelvezet: Java Persistence Query Language (JPQL) • Lekérdezések optimalizálása és biztonsága • Öröklődés kezelése ORM keretrendszerekben 	Előadás, magyarázat, példák	
11. IoC konténerek <ul style="list-style-type: none"> • Könyvtár és keretrendszer közötti különbségek tisztázása • A Dependency Injection tervezési minta 	Előadás, magyarázat, példák	

<ul style="list-style-type: none"> • A Spring IoC konténer: beanek, hatókörök, factory 		
12. Vállalati keretrendszerek: Spring Boot <ul style="list-style-type: none"> • Konfigurációs állományok és profilok használata • Integráció build rendszerekkel 	Előadás, magyarázat, példák	
13. Többrétegű alkalmazás kialakítása vállalati keretrendszerrel <ul style="list-style-type: none"> • Megjelenítési réteg implementációja: Spring Web és Spring Web MVC • Adatelérési réteg generálása/absztraktizálása: Spring Data JPA • Biztonsági szempontok lefedése: Spring Security 	Előadás, magyarázat, példák	
14. Ismétlés	Előadás, magyarázat, példák	
Könyvészet <ol style="list-style-type: none"> 1. Károly Simon. <i>Kenyerünk Java: a Java programozás alapjai</i>. Presa Universitară Clujeană, 2010. 2. Balaji Varanasi, Sudha Belica. <i>Introducing Gradle</i>. Apress, 2015. 3. Margit Antal. <i>Java alapú webtechnológiák</i>. Scientia, 2009. 4. Christian Bauer, Gavin King, Gary Gregory. <i>Java Persistence with Hibernate</i>. Manning Publications, 2015. 5. Madhusudhan Konda. <i>Just Spring Data Access: Covers JDBC, Hibernate, JPA and JDO</i>. O'Reilly Media, 2012. 6. Iuliana Cosmina. <i>Pivotal Certified Professional Spring Developer Exam: A Study Guide</i>. Apress, 2015. 7. Martin Fowler. <i>Patterns of Enterprise Application Architecture</i>. Addison-Wesley, 2002. 		
8.2. Szeminárium / labor	Didaktikai módszerek	Megjegyzések
1-3. hét: Gradle <ul style="list-style-type: none"> • multimodul projekt elkészítése • memóriában tárolt adatok megjelenítése • naplózás logback segítségével 	Példaprogramok tanulmányozása, laborfeladatok értékelése	
4-5. hét: Webalkalmazás <ul style="list-style-type: none"> • Java webalkalmazás kitelepítése webkonténerre • Servlet API és sablonmotor használata 	Példaprogramok tanulmányozása, laborfeladatok értékelése	
6-7. hét: Adatelérési réteg <ul style="list-style-type: none"> • JDBC beépítése • Abstract DAO Factory minta • Konfigurációs állományok használata 	Példaprogramok tanulmányozása, laborfeladatok értékelése	
8-10. hét: Objektum-relációs leképezés <ul style="list-style-type: none"> • Java Persistence API bevezetése • Váltható moduláris adatelérési réteg 	Példaprogramok tanulmányozása, laborfeladatok értékelése	
11-12. hét: Dependency Injection <ul style="list-style-type: none"> • A Spring IoC konténer bevezetése • Meglévő rendszerben levő beanek Spring beanné alakítása 	Példaprogramok tanulmányozása, laborfeladatok értékelése	
13-14. hét: Többrétegű alkalmazás kialakítása vállalati keretrendszerrel <ul style="list-style-type: none"> • Spring Web bevezetése • Spring Data JPA bevezetése 	Példaprogramok tanulmányozása, laborfeladatok értékelése	

Könyvészet

1. Introduction to Gradle: <https://gradle.com/training/introduction-to-gradle/>
2. Java EE 7 Specification APIs: <https://docs.oracle.com/javaee/7/api/overview-summary.html>
3. Spring | Learn: <https://spring.io/learn>
4. Baeldung | Java, Spring and Web Development tutorials: <https://www.baeldung.com>

9. A tantárgy tartalmának összhangba hozása az episztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásaival

- A tantárgy tematikája nagy átfedést mutat az egyetemi oktatásban a fontosabb egyetemeken oktatott hasonló tematikájú tantárgyak tartalmával.
- A tananyagok kidolgozása a nemzetközileg legelismertebb szerzők munkái alapján történt, az ajánlott könyvészet szintén a terület legrelevánsabb munkái alapján van összeállítva.
- A tantárgy keretein belül oktatott témák szükségesek a szoftverfejlesztői iparban történő elhelyezkedéshez, a cégek elvárják az ilyen jellegű ismereteket.

10. Értékelés

Tevékenység típusa	10.1 Értékelési kritériumok	10.2 Értékelési módszerek	10.3 Aránya a végső jegyben
10.4 Előadás	A kurzusokon bemutatott elméleti ismeretek megfelelő elsajátítása	Írásbeli vizsga (V)	50%
10.5 Szeminárium / Labor	A tanult elméleti ismeretek megfelelő gyakorlatba helyezése	Laborfeladatok (L)	50%
10.6 A teljesítmény minimumkövetelményei			
Az átmenő jegy feltételei:			
<ul style="list-style-type: none">• Az írásbeli vizsga (V) eredménye minimum 50%.• Az értékelések összegzése (V+L) minimum 50%.• Maximum 2 (indokolatlan) laboróráról való hiányzás.			

Kitöltés dátuma

.....

Előadás felelőse

Dr. Sulyok Csaba, tanársegéd

Szeminárium felelőse

Dr. Sulyok Csaba, tanársegéd

Az intézeti jóváhagyás dátuma

.....

Intézetigazgató

Dr. András Szilárd Károly, egyetemi docens