

## A TANTÁRGY ADATLAPJA

### 1. A képzési program adatai

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

### 2. A tantárgy adatai

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

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

3.1 Heti óraszám	<b>4</b>	Melyből: 3.2 előadás	<b>2</b>	3.3 szeminárium/labor	<b>2</b>
3.4 Tantervben szereplő össz-óraszám	<b>56</b>	Melyből: 3.5 előadás	<b>28</b>	3.6 szeminárium/labor	<b>28</b>
A tanulmányi idő elosztása:					óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					<b>8</b>
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					<b>8</b>
Szemináriumok / laborok, házi feladatok, portofóliók, referátumok, esszék kidolgozása					<b>28</b>
Egyéni készségfejlesztés (tutorálás)					-
Vizsgák					<b>4</b>
Más tevékenységek: .....					-
3.7 Egyéni munka össz-óraszama	<b>44</b>				
3.8 A félév össz-óraszama	<b>100</b>				
3.9 Kreditszám	<b>4</b>				

### 4. Előfeltételek (ha vannak)

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

## 5. Feltételek (ha vannak)

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

## 6. Elsajátítandó jellemző kompetenciák

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

## 8. A tantárgy tartalma

8.1. Előadás	Didaktikai módszerek	Megjegyzések
1. Bevezető <ul style="list-style-type: none"> <li>• Enterprise keretrendszerek körülírása</li> <li>• Enterprise minták, architektúrák</li> </ul>	Előadás, magyarázat, példák	
2. Modern build rendszerek <ul style="list-style-type: none"> <li>• Build és függőségkezelési rendszerek szükségessége</li> <li>• Maven: működési elv, telepítés-leíró, konvencionális struktúra, függőségek, projekt életciklus, pluginok</li> </ul>	Előadás, magyarázat, példák	
3. Modern build rendszerek és naplózás <ul style="list-style-type: none"> <li>• Gradle: működési elv, telepítés-leíró, függőségek, taskok, pluginok</li> <li>• Multimodul projektek</li> <li>• Naplózás: slf4j, logback</li> </ul>	Előadás, magyarázat, példák	
4. Java-alapú webalkalmazások <ul style="list-style-type: none"> <li>• Bevezető a többrétegű architektúrákba</li> <li>• Java webkonténerek: Tomcat, Servlet API</li> </ul>	Előadás, magyarázat, példák	
5. Model-View-Controller minta <ul style="list-style-type: none"> <li>• Az MVC minta leírása, thin és rich kliensek megkülönböztetése</li> <li>• MVC megvalósítása Java-ban: JSP, Thymeleaf</li> </ul>	Előadás, magyarázat, példák	
6. Többrétegű architektúrák és adatelérési réteg <ul style="list-style-type: none"> <li>• Perzisztencia alapok: JDBC</li> <li>• Az Abstract DAO Factory tervezési minta</li> <li>• Connection pooling</li> </ul>	Előadás, magyarázat, példák	
7. Java SE kiegészítések <ul style="list-style-type: none"> <li>• Konfigurációs állományok: Properties</li> <li>• Profilozás</li> <li>• Nemzetköziesítés (i18n)</li> </ul>	Előadás, magyarázat, példák	
8. Futási idejű befolyásolás <ul style="list-style-type: none"> <li>• Reflection és a Java Reflection API</li> <li>• Biztonsággal kapcsolatos kérdések</li> </ul>	Előadás, magyarázat, példák	
9. IoC konténerek <ul style="list-style-type: none"> <li>• Könyvtár és keretrendszer közötti különbségek tisztázása</li> <li>• A Dependency Injection tervezési minta</li> <li>• A Spring IoC konténer: beanek, hatókörök, factory</li> </ul>	Előadás, magyarázat, példák	
10. Vállalati keretrendszerek: Spring Boot <ul style="list-style-type: none"> <li>• Konfigurációs állományok és profilok használata</li> <li>• Integráció build rendszerekkel</li> </ul>	Előadás, magyarázat, példák	
11. Megjelenítési réteg vállalati keretrendszerrel <ul style="list-style-type: none"> <li>• Spring Web és Spring Web MVC</li> </ul>	Előadás, magyarázat, példák	
12. Magasszintű perzisztencia <ul style="list-style-type: none"> <li>• Objektum-relációs leképezés (ORM)</li> <li>• A Java Persistence API (JPA) s ennek implementációi (Hibernate)</li> </ul>	Előadás, magyarázat, példák	

13. Magasszintű perzisztencia <ul style="list-style-type: none"> <li>Leképezési nyelvezet: Java Persistence Query Language (JPQL)</li> <li>Adatelérési réteg generálása/absztraktizálása: Spring Data JPA</li> </ul>	Előadás, magyarázat, példák	
14. Ismétlés	Előadás, magyarázat, példák	
<b>Könyvészet</b>		
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"> <li>multimodul projekt elkészítése</li> <li>memóriában tárolt adatok megjelenítése</li> <li>naplózás logback segítségével</li> </ul>	Példaprogramok tanulmányozása, laborfeladatok értékelése	
4-5. hét: Webalkalmazás <ul style="list-style-type: none"> <li>Java webalkalmazás kitelepítése webkonténerre</li> <li>Servlet API és sablonmotor használata</li> </ul>	Példaprogramok tanulmányozása, laborfeladatok értékelése	
6-7. hét: Adatelérési réteg <ul style="list-style-type: none"> <li>JDBC beépítése</li> <li>Abstract DAO Factory minta</li> <li>Konfigurációs állományok használata</li> </ul>	Példaprogramok tanulmányozása, laborfeladatok értékelése	
8-9. hét: Dependency Injection <ul style="list-style-type: none"> <li>A Spring IoC konténer bevezetése</li> <li>Meglévő rendszerben levő beanek Spring beanné alakítása</li> </ul>	Példaprogramok tanulmányozása, laborfeladatok értékelése	
10-11. hét: Megjelenítési réteg <ul style="list-style-type: none"> <li>Spring Web bevezetése</li> <li>DTO/Mapper</li> </ul>	Példaprogramok tanulmányozása, laborfeladatok értékelése	
12-14. hét: Többrétegű alkalmazás kialakítása vállalati keretrendszerrel <ul style="list-style-type: none"> <li>Java Persistence API bevezetése</li> <li>Váltható moduláris adatelérési réteg</li> <li>Spring Data JPA</li> </ul>	Példaprogramok tanulmányozása, laborfeladatok értékelése	
<b>Könyvészet</b>		
1. Introduction to Gradle: <a href="https://gradle.com/training/introduction-to-gradle/">https://gradle.com/training/introduction-to-gradle/</a> 2. Java EE 7 Specification APIs: <a href="https://docs.oracle.com/javaee/7/api/overview-summary.html">https://docs.oracle.com/javaee/7/api/overview-summary.html</a> 3. Spring   Learn: <a href="https://spring.io/learn">https://spring.io/learn</a> 4. Baeldung   Java, Spring and Web Development tutorials: <a href="https://www.baeldung.com">https://www.baeldung.com</a>		

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

<ul style="list-style-type: none"><li>• A tantárgy tematikája nagy átfedést mutat az egyetemi oktatásban a fontosabb egyetemeken oktatott hasonló tematikájú tantárgyak tartalmával.</li><li>• A tananyagok kidolgozása a nemzetközileg legelismertebb szerzők munkái alapján történt, az ajánlott könyvészet szintén a terület legrelevánsabb munkái alapján van összeállítva.</li><li>• A tantárgy keretein belül oktatott témák szükségesek a szoftverfejlesztői iparban történő elhelyezkedéshez, a cégek elvárják az ilyen jellegű ismereteket.</li></ul>
--

## 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"><li>• Az írásbeli vizsga (V) eredménye minimum 50%.</li><li>• Az értékelések összegzése (V+L) minimum 50%.</li><li>• Maximum 2 (indokolatlan) laboróráról való hiányzás.</li></ul>			

Kitöltés dátuma

.....2023. április 11.....

Előadás felelőse

Dr. Sulyok Csaba, egyetemi adjunktus

Szeminárium felelőse

Dr. Sulyok Csaba, egyetemi adjunktus

Az intézeti jóváhagyás dátuma

.....2023. április 28.....

Intézetigazgató

Dr. András Szilárd Károly, egyetemi docens