

## 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</b>

### 2. A tantárgy adatai

2.1 A tantárgy neve	<b>Közös projekt</b>						
2.2 Az előadásért felelős tanár neve							
2.3 A szemináriumért felelős tanár neve	dr. Csató Lehel						
2.4 Tanulmányi év	3	2.5 Félév	5	2.6. Értékelés módja	kollokvium	2.7 Tantárgy típusa	kötelező – alap

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

3.1 Heti óraszám	2	melyből: 3.2 előadás	0	3.3 szeminárium/labor	2
3.4 Tantervben szereplő össz-óraszám	28	melyből: 3.5 előadás	0	3.6 szeminárium/labor	28
A tanulmányi idő elosztása:					<b>Óra</b>
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					4
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					4
Szemináriumok / laborok, házi feladatok, portofóliók, referátumok, esszék kidolgozása					35
Egyéni készségfejlesztés (tutorálás)					0
Vizsgák					4
Más tevékenységek: .....					0
3.7 Egyéni munka össz-óraszámja	47				
3.8 A félév össz-óraszámja	75				
3.9 Kreditszám	3				

### 4. Előfeltételek (ha vannak)

4.1 Tantervi	<ul style="list-style-type: none"> <li>Nincs</li> </ul>
4.2 Kompetenciabeli	<ul style="list-style-type: none"> <li>Fejlett programozási módszerek és környezetek, tervezési minták, software engineering</li> </ul>

### 5. Feltételek (ha vannak)

5.1 Az előadás lebonyolításának feltételei	-
5.2 A szeminárium / labor lebonyolításának feltételei	<ul style="list-style-type: none"> <li>Számítógépekkel felszerelt laborterem, megfelelő szoftverek, hozzáférés a megfelelő szerverekhez (alkalmazáserver, verziókövető, projekt menedzsment)</li> </ul>

## 6. Elsajátítandó jellemző kompetenciák

<b>Szakmai kompetenciák</b>	<ul style="list-style-type: none"> <li>• modern szoftverfejlesztési módszerek és stratégiák elsajátítása</li> <li>• a modern szoftverfejlesztésben alkalmazott tervezési eljárások, minták megismerése</li> <li>• az operatív csapatmunkához szükséges kvalitások elsajátítása, tapasztalatok szerzése</li> <li>• modern integrált fejlesztői környezetek, API-k, technológiák és keretrendszerek megismerése</li> <li>• szoftver verifikációs és validációs módszerek alkalmazása</li> </ul>
<b>Transzverzális kompetenciák</b>	<ul style="list-style-type: none"> <li>• Önálló tanulás, kritikus gondolkodás, problémabeazonosítás és feladatmegoldás</li> <li>• Összetettebb projekteknél (pl. vállalati alkalmazások) alkalmazott módszerek, stratégiák, technológiák elsajátítása</li> <li>• Csapatmunkához szükséges kvalitások elsajátítása, projektmenedzsmenttel kapcsolatos alapismeretek</li> </ul>

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

7.1 A tantárgy általános célkitűzése	<ul style="list-style-type: none"> <li>• modern szoftverfejlesztési módszerek és stratégiák elsajátítása</li> <li>• a modern szoftverfejlesztésben alkalmazott tervezési eljárások, minták megismerése</li> <li>• az operatív csapatmunkához szükséges kvalitások elsajátítása, tapasztalatok szerzése</li> <li>• modern integrált fejlesztői környezetek, API-k, technológiák és keretrendszerek megismerése</li> <li>• szoftver verifikációs és validációs módszerek alkalmazása</li> </ul>
7.2 A tantárgy sajátos célkitűzései	<ul style="list-style-type: none"> <li>• Szoftverfejlesztési módszerek és stratégiák elsajátítása</li> <li>• A szoftverfejlesztési folyamat (analízis, tervezés, implementáció, verifikáció és validáció, a projekt átadása) megismerése</li> <li>• Rendszertervezés elsajátítása, tervezési minták, UML alkalmazása</li> <li>• Integrált fejlesztői környezetek megismerése</li> <li>• Csapatmunkát támogató rendszerek megismerése (verziókövetés, hibajelentés, projekt menedzsment stb.)</li> <li>• Javasolt API-k, technológiák, keretrendszerek használata</li> </ul>

## 8. A tantárgy tartalma

8.1 Előadás - nincs	Didaktikai módszerek	Megjegyzések
8.2 Szeminárium / Labor	Didaktikai módszerek	Megjegyzések
Szeminárium - nincs		
Labor + projekt:		
<p>A bevezető előadások és laborórák keretein belül tárgyalt témakörök:</p> <ol style="list-style-type: none"> <li>1. Szoftverfejlesztési módszerek és stratégiák a szoftverfejlesztés folyamata <ul style="list-style-type: none"> <li>• tipikus szerepek (kliens, projektvezető, tervezők, fejlesztők, tesztelők, stb.)</li> <li>• szoftverfejlesztési stratégiák</li> <li>• UML alapok</li> </ul> </li> </ol>		

- |  |  |  |
|--|--|--|
| <ol style="list-style-type: none"><li>2. Integrált fejlesztői környezetek<ul style="list-style-type: none"><li>• általános tudnivalók</li><li>• javasolt tervezési eszközök, környezetek</li><li>• javasolt programozási környezetek</li></ul></li><li>3. Analízis<ul style="list-style-type: none"><li>• követelmény specifikáció</li><li>• használati eset analízis</li><li>• domain analízis</li></ul></li><li>4. Tervezés<ul style="list-style-type: none"><li>• az architektúra megtervezése</li><li>• modern fejlesztési technológiák</li><li>• tervezési minták</li><li>• diagramok (osztály, szekvencia, állapot)</li></ul></li><li>5. Csapatmunka<ul style="list-style-type: none"><li>• verziókövetés és forráskód menedzsment</li><li>• revision controll terminológia</li><li>• CVS (Concurrent Versioning System)</li><li>• SVN (Subversion)</li><li>• rétegek</li><li>• fájlrendszer és tranzakciók</li><li>• verziókövetés Eclipse környezetben: a Subclipse és Subversive bemutatása</li></ul></li><li>6. Implementációs részletek<ul style="list-style-type: none"><li>• standardok, konvenciók</li><li>• forráskód dokumentálása</li></ul></li><li>7. Fejlesztési technológiák<ul style="list-style-type: none"><li>• különböző projekt típusok fejlesztéséhez javasolt API-k</li><li>• különböző projekt típusok fejlesztéséhez javasolt technológiák</li></ul></li><li>8. Keretrendszerek<ul style="list-style-type: none"><li>• különböző projekt típusok fejlesztéséhez javasolt keretrendszerek</li></ul></li><li>9. Grafikus felhasználói interfész<ul style="list-style-type: none"><li>• javasolt GUI eszköztárak</li><li>• fejlesztői eszközök</li><li>• szoftver-ergonómiai szempontok</li></ul></li><li>10. Hibakövetés<ul style="list-style-type: none"><li>• általános tudnivalók</li><li>• hibakövető rendszerek</li><li>• a Bugzilla bemutatása</li></ul></li><li>11. Verifikáció és validáció (V&amp;V)<ul style="list-style-type: none"><li>• a V&amp;V folyamat, módszerek, általános tudnivalók</li></ul></li></ol> |  |  |
|--|--|--|

<ul style="list-style-type: none"> <li>• szoftvertesztelés</li> <li>• tesztelés típusok, alkalmazott módszerek</li> <li>• tesztervek</li> <li>• eredmények kiértékelése</li> </ul> <p>12. Unit tesztelés</p> <ul style="list-style-type: none"> <li>• általános tudnivalók</li> <li>• unit testing Java-ban, a JUnit</li> <li>• a Fixture fogalom</li> <li>• Test Suite létrehozása, alkalmazása</li> <li>• az Eclipse által biztosított JUnit támogatás</li> <li>• JUnit alapú tesztelés Eclipse környezetben (példa)</li> </ul> <p>13. A projekt átadása</p> <ul style="list-style-type: none"> <li>• felhasználói dokumentáció</li> <li>• a végtermék kézbesítése (csomagolás, telepítő csomag)</li> <li>• a rendszer bemutatása</li> </ul> <p>A félév elején a hallgatók 4-5 tagú csoportokat alakítanak. A témák rögzítése után elkezdődik az egész félév alatti folyamatos fejlesztési tevékenység. A laborórák a felsorolt elméleti fogalmak bemutatása mellett a közös egyeztetéseknek szolgálnak keret. A cégekkel történő együttműködések alapuló projektek további specifikus szabályokat követnek.</p> <p>Az egyes projektek fejlesztésének folyamatos követése verziókövető rendszeren keresztül történik, a felmerült problémák, javaslatok a gyakorlati órákon kerülnek megbeszélésre (sürgősebb esetekben más kommunikációs módszerek is alkalmazhatóak, pl. e-mail, chat, stb.).</p> <p>A projekt különböző fázisainak megfelelő anyagok leadása határidőhöz kötött, ezeknek az anyagoknak a kiértékelése, megbeszélése szintén a gyakorlati órákon történik.</p>		
<p><b>Könyvészet</b></p> <p>A megfelelő könyvészeti források beazonosítása csapatonként, az illető projekt specifikumainak megfelelően történik.</p> <p>Általánosan: felhasznált technológiákkal, környezetekkel, fejlesztői és tervezői eszközökkel, csapatmunkát támogató eszközökkel, tervezési mintákkal kapcsolatos dokumentációk.</p>		

**9. Az episztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásainak összhangba hozása a tantárgy tartalmával.**

- A tantárgy tematikája nagy átfedést mutat az egyetemi oktatásban a fontosabb egyetemeken oktatott hasonló tematikájú tantárgyak tartalmával.
- A tananyagok kidolgozása nemzetközileg elismert szerzők munkái alapján történt, az ajánlott könyvészet szintén a terület releváns munkái alapján lesz összeállítva, a különböző projektek specifikumainak megfelelően.

- 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			
10.5 Szeminárium / Labor	Projekt	A projekt pontozása, az egyéni hozzájárulások figyelembevételével. A projekt bemutatása bizottság előtt történik, a végső jegy a bizottsági tagoktól kapott, rögzített kritériumrendszeren alapuló pontszámok összesítése alapján lesz meghatározva.	100%
10.6 A teljesítmény minimumkövetelményei			
Az átmenő jegy feltételei:			
<ul style="list-style-type: none"> <li>• A projektnek átmenő osztályzatot kell kapnia.</li> <li>• A végső jegy minimálisan 5-ös.</li> </ul>			

Kitöltés dátuma

2014.04.25

Az intézeti jóváhagyás dátuma

.....

Előadás felelőse

-

Intézetigazgató

dr. Szenkovits Ferenc

Szeminárium felelőse

dr. Csató Lehel