

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

2. A tantárgy adatai

2.1 A tantárgy neve	Közös projekt						
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	2	2.6. Értékelés módja	kollokvium	2.7 Tantárgy típusa	kötelező – szak

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	24	melyből: 3.5 előadás	0	3.6 szeminárium/labor	24
A tanulmányi idő elosztása:					Óra
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					15
Egyéni készségfejlesztés (tutorálás)					0
Vizsgák					3
Más tevékenységek:					0
3.7 Egyéni munka össz-óraszámja	26				
3.8 A félév össz-óraszámja	50				
3.9 Kreditszám	2				

4. Előfeltételek (ha vannak)

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

5. Feltételek (ha vannak)

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

6. Elsajátítandó jellemző kompetenciák

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

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

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

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"> 1. Szoftverfejlesztési módszerek és stratégiák a szoftverfejlesztés folyamata <ul style="list-style-type: none"> • tipikus szerepek (kliens, projektvezető, tervezők, fejlesztők, tesztelők, stb.) • szoftverfejlesztési stratégiák • UML alapok 		

- | | | |
|--|--|--|
| <ol style="list-style-type: none">2. Integrált fejlesztői környezetek<ul style="list-style-type: none">• általános tudnivalók• javasolt tervezési eszközök, környezetek• javasolt programozási környezetek3. Analízis<ul style="list-style-type: none">• követelmény specifikáció• használati eset analízis• domain analízis4. Tervezés<ul style="list-style-type: none">• az architektúra megtervezése• modern fejlesztési technológiák• tervezési minták• diagramok (osztály, szekvencia, állapot)5. Csapatmunka<ul style="list-style-type: none">• verziókövetés és forráskód menedzsment• revision controll terminológia• CVS (Concurrent Versioning System)• SVN (Subversion)• rétegek• fájlrendszer és tranzakciók• verziókövetés Eclipse környezetben: a Subclipse és Subversive bemutatása6. Implementációs részletek<ul style="list-style-type: none">• standardok, konvenciók• forráskód dokumentálása7. Fejlesztési technológiák<ul style="list-style-type: none">• különböző projekt típusok fejlesztéséhez javasolt API-k• különböző projekt típusok fejlesztéséhez javasolt technológiák8. Keretrendszerek<ul style="list-style-type: none">• különböző projekt típusok fejlesztéséhez javasolt keretrendszerek9. Grafikus felhasználói interfész<ul style="list-style-type: none">• javasolt GUI eszköztárak• fejlesztői eszközök• szoftver-ergonómiai szempontok10. Hibakövetés<ul style="list-style-type: none">• általános tudnivalók• hibakövető rendszerek• a Bugzilla bemutatása11. Verifikáció és validáció (V&V)<ul style="list-style-type: none">• a V&V folyamat, módszerek, általános tudnivalók | | |
|--|--|--|

<ul style="list-style-type: none"> • szoftvertesztelés • tesztelés típusok, alkalmazott módszerek • tesztervek • eredmények kiértékelése <p>12. Unit tesztelés</p> <ul style="list-style-type: none"> • általános tudnivalók • unit testing Java-ban, a JUnit • a Fixture fogalom • Test Suite létrehozása, alkalmazása • az Eclipse által biztosított JUnit támogatás • JUnit alapú tesztelés Eclipse környezetben (példa) <p>13. A projekt átadása</p> <ul style="list-style-type: none"> • felhasználói dokumentáció • a végtermék kézbesítése (csomagolás, telepítő csomag) • a rendszer bemutatása <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>Könyvészet</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"> • A projektnek átmenő osztályzatot kell kapnia. • A végső jegy minimálisan 5-ös. 			

Kitöltés dátuma

Előadás felelőse

Szeminárium felelőse

2014.04.25

-

dr. Csató Lehel

Az intézeti jóváhagyás dátuma

Intézetigazgató

.....

dr. Szenkovits Ferenc