

# GEIAL31H-B2, GEIAL31H-BL2

## Szoftvertesztelés

Mérnökinformatikus alapszak (BSc), Programtervező informatikus alapszak (BSc)

### Tantárgy feladata és célja:

A tárgy célja a szoftvertesztelés főbb módszereinek és technikájának a bemutatása. A tesztelési szintek megismerése. Test First és Test Driven Development módszerek ismertetése. A különböző szoftvertesztelési technikák gyakorlatban történő megismerése.

### Tárgy tematikus leírása:

Szoftver-minőségi kritériumok áttekintése és minőségbiztosítási folyamat megvalósulásának vizsgálata szoftverprojektekben. Tesztelési alapelvek. Funkcionális tesztelés, Unit, Integrációs, Alrendszer és Rendszer tesztek ismertetése. JUnit keretrendszer ismertetése. Agilis fejlesztési módszertanok átfogó ismertetése, Test First Development fejlesztési módszer empirikus vizsgálata

### Ütemterv

# Hét	Előadás	Gyakorlat
1. hét	Követelmények ismertetése	Labor használati rend, oktatási anyagok elérése, szoftverkörnyezet beállítása, Használt technológiák ismertetése, Git
2. hét	Szoftver minőség jellemzők	
3. hét	Szoftver hibák és forrásaik	Unit teszt alapok
4. hét	Tesztelés alapjai. Tesztelés céljai, tesztelés és hibakezelés, teszttevékenységek, munkatermékek	
5. hét	Tesztelés a sw életciklusában	Kódolási gyakorlat
6. hét	Statikus tesztelés	
7. hét	Teszttechnikák <b>TDD és BDD</b>	Kódolási gyakorlat
8. hét	Őszi szünet	
9. hét	Tesztmenedzsment	UI tesztelés kódolási gyakorlat 1
10. hét	Teszt támogató eszközök	
11. hét	Refactoring 1	UI tesztelés kódolási gyakorlat 2
12. hét	Refactoring 2	
13. hét	Teljesítmény tesztelés, biztonsági tesztelés	Pótlások
14. hét	Összefoglalás	

## Aláírás megszerzése:

A zárthelyi dolgozaton legalább elégséges = 40% eredmény elérése

Pótlások: az utolsó tanulmányi héten

Az előadások legalább 60%-án való részvétel, a gyakorlatok legalább 70%-án való részvétel

## Számonkérés módja:

Kollokvium A vizsgaidőszakban írásbeli + szóbeli vizsga

## Gyakorló feladatok

A 2025. évben az alábbi feladatok szerepelnek

1. Készítsd egy github repozitóriumot. Használd a következő utasításokat:

- git init
- git status
- git add
- git diff
- git commit
- git log
- git pull
- git push

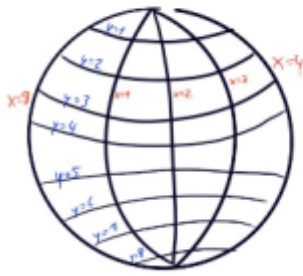
Oszd meg a github repozitóriumodat. Ellenőrizd a megosztást.

2. Test Driven development elveket alkalmazva implementáld arab szám - római szám konvertert! A kódot töltsd fel egy github repozitóriumba!

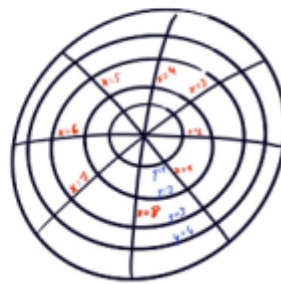
3. Készíts olyan kódot ami a holdjáróval feltérképezi a bolygót!

- Meg kell adni a kezdő (x,y) koordinátát, és azt, hogy merre néz a holdjáró (N, E, S, W)
- A holdjáró karakteres utasításokat kap
- Implementáld az előre/hátra parancsokat (f,b)
- Implementáld a balra/jobbra parancsokat (l,r)
- Kezeld le a térkép "széleit"
- Kezeld le, hogy a bolygók gömbök!
- A bolygón akadályok is vannak. Minden lépés előtt a holdjáró ellenőrzi, van-e akadály. Ha a holdjáró akadályt érzékel, akkor visszatér az utolsó pontra és jelenti az akadályt
- Szigorúan TDD elveket kell követni, nincs kivétel!
- A kódot töltsd fel egy github repozitóriumba! Minden teszteset egy külön commit legyen!

Oldalnézet



Felülnézet



Tipp:

## Ajánlott irodalom:

- Mauro Pezzé, Michal Young: [Software Testing and Analysis](#)
- International Software Testing Qualifications Board: [Certified Tester – Foundation Level Syllabus \(2024\)](#)
- Ficsor Lajos, Dr. Kovács László, Dr. Krizsán Zoltán, Dr. Kuser Gábor - , [Szoftvertesztelés jegyzet](#)
- Dorothy Graham, Erik van Veenendaal, Isabel Evans és Rex Black - A szoftvertesztelés alapjai
- Fehér Krisztián - Szoftvertesztelési alapismeretek
- Robert C. Martin – Tiszta kód
- Hornyák Olivér - Szoftverminőség-jellemzők előadás fóliák  
swtest2020\_2ea.ppt
- Hornyák Olivér - Szoftverminőség  
swmin\_2024\_2ea\_hu.pptx

## Mock gyakorlat

- [Mock Fóliák](#)
- Letölthető kód:

2024\_09\_uni-sw-unit-testing-main.zip

### Feladat:

1. Nézd át a beimportált függőségeket és frissítsd a legfrissebb verziókra
2. Írj mock tesztet a Triangle osztály mind az 5 metódusához
3. Írj mock tesztet, ami ellenőrzi a Fibonacci sorozat 15. és 20. elemét ( nem kell a tényleges kalkulációt implementálni)
4. Írj Unit tesztet, ami ellenőrzi a Fibonacci sorozat 8. és 23. elemét (kell a tényleges kalkuláció implementálás)

## UI tesztelés gyakorlat

- Leírás [UI tesztautomatizálás](#)
- Letölthető kódok

UI tesztautomatizálási kódok

From:  
<https://edu.iit.uni-miskolc.hu/> - **Institute of Information Science - University of Miskolc**

Permanent link:  
<https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szoftvertesztes?rev=1758448244>

Last update: **2025/09/21 09:50**

