**Tantárgyi tematika és félévi követelményrendszer**

**BPI1113L/BMI1302L Programozási technológiák**

**Oktató:** Vályi Sándor Zoltán (PhD), illetve az Epam Systems Rt. debreceni irodájának oktatói, elsődlegesen Guta Norbert

Az órák online formában, a Teams-en, az Progtech-NL csoportban mennek. A forráskódokat, prezentációkat a külsős Epam-oktatók a <https://github.com/epam-nye-cooperation/epam-nye-progtech> webhelyre helyezik el.

**A foglalkozásokon történő részvétel:**

* A gyakorlati foglalkozásokon a részvétel kötelező. **Online órák lesznek a NyE Teams-rendszerében, a Progtech-NL csoportban.**
* ***A tárgy ismeretádasának módszere: a tárgy nappalis kurzusán menő meetingek felvételeinek – lehetőség hÍján offline – megtekintése, anyagának követése. Szóval ez még az első órák előtt is teendőket ró a hallgatókra. Utána a levelezős órákon is – kevésbé részletesen – még átveszik az anyagot.***

**Félévi követelmény:** gyakorlati jegy

**A félévközi ellenőrzések követelményei:**

* Beadandó otthoni készítése és 2 ütemben történő megvédése. A beadandó tartalma: egy asztali (desktop) parancssoros (command line) játékprogram, amely a kurzuson átvett témák nagy részét használja, előre definiált mélységben.
* Valamint egy elméletibb jellegű beadandó készítése, ami a programtervezési mintákról szól, és kritériumkövetelmény, pontszám nem jár hozzá, csak sikeres/nem sikeres értékelést kap. Ezt nem kell védeni. dolgozat cime ez*: Tervezési minták egy OO programozási nyelvben. MVC, mint modell-nézet-vezérlő minta és néhány másik tervezési minta.*

**Az értékelés módja, ütemezése:**

* Beadandó otthoni készítése és online megvédése. A védések időpontjai az alábbi óraütemezésben kiolvashatók. 3 alkalomból áll a levelezős kurzus. Két részből áll a védés, az első védésrész a második alkalommal, a második védésrész a harmadik alkalomkor jön sorra.
* A programtervezési mintás beadandó beadási határideje a 2023. december 4.

**Az érdemjegy kialakításának módja:**

A félévi gyakorlati jegyet az összes pontszám alapján állapítjuk meg: 70% -- elégséges, 77% -- közepes, 85% -- jó, 93% -- jeles. Elégtelen gyakorlati jegy javítása a TVSz szerint lehetséges, csak egyszer a vizsgaidőszakban, a beadandó-védés pótlásával.

**Féléves tematika:**

| Alkalom | **Témakör** | **Megjegyzés** |
| --- | --- | --- |
| 1. | Bemutatkozás, Eszközök telepítése | * JDK telepítés (11) * IDE (IntelliJ) * Verzió kezelés / Git alapok   + Git telepítés   + IDE / Tortoise Git   + GitHub regisztráció   + repo létrehozás   + clone, pull, commit, push * Maven telepítés |
| 1. | Maven alapok | * Megmutatni hogy build tool nélkül nehéz a fejlesztés (javac, JAR) * Maven életciklusok (clean, package, test, install) * Alap maven projekt létrehozása * pom.xml * Függőség kezelés * Pluginek |
| 1. | Prog2 (Java / OO alapok) ismétlés | * Java ismétlés   + Exception handling   + Collection API   + Generikusok * OO alapok   + Absztrakció, Polimorfizmus, Öröklődés, Enkapszuláció * OO irányelvek   + magas kohézió, alacsony kötés,   + SOLID   + KISS   + YAGNI   + DRY * Clean Code   + elnevezési konvenciók, beszédes változónevek, kommentek (inline comments, JavaDoc) * Checkstyle Maven plugin * Gyakorlati feladat: egy szándékosan rossz (állatorvosi ló) forráskódot bemutatni és megkeresni a hibáit |
| 1. | Tesztelés | * Tesztelés céljának bemutatása   + tesztelési módszerek és szintek   + testing pyramid * Elméleti alapozás:   + Egységtesztelés (fontosságának kihangsúlyozása, fejlesztés alatt már írni kell, TDD)   + Mockolás   + FIRST * Gyakorlat   + JUnit5   + Mockito   + Given-When-Then struktúra   + teszt metódusok elnevezési konvenciója (testMethodShouldDoSomethingWhenCondition) * Jacoco, Surefire Maven plugins (80% coverage check felkonfigurálása) |
| 1. | Komplexebb közös programozási feladat | * Command Line játék (például torpedó) alkalmazás vázának elkezdése * Logolás * VO objektumok (Object Methods fontossága: equals, hashCode, toString) * Java Packaging (alap megközelítés model, ui, service, persistance package-ek) |
| 1. | Komplexebb közös programozási feladat | * Design Pattern (például: Singletone, Observer, Builder) * Immutability |
| 2. | **Beadandó védés part 1** | * Elvárások az első beadandó védésre:   + Maven project összerakása (pluginokkal)   + Model osztályok leimplemnetálása   + Command line interakciók kezelése (tesztekkel) |
| 2. | OO Tervezési minták |  |
| 2. | Stream API |  |
| 2. | JDBC |  |
| 2. | XML Feldolgozás (tartalék téma) |  |
| 2. | Komplexebb közös programozási feladat |  |
| 3. | **Beadandó védés, 2. rész** |  |