

Programozási technológiák (INDK321)

Jeszenszky Péter

2016. február 21.

1. Gyakorlati követelmények

1.1. Hiányzások

A gyakorlatok teljesítéséhez elengedhetetlen az órák rendszeres látogatása. A gyakorlatvezetők minden alkalommal névsort írnak, minden hallgató számára **legfeljebb három hiányzás** engedélyezett a félév folyamán. Négy vagy több hiányzás esetén a félév végi érdemjegy automatikusan elégtelen.

Elmulasztott gyakorlatok pótlására korlátozott létszámban lehetőség van valamely gyakorlatvezető az aktuális héten tartott más Programozási technológiák gyakorlatain. Ehhez azonban sem ülőhelyet, sem számítógépet nem köteles biztosítani a gyakorlatvezető. Más mód nincs a hiányzások pótlására, ez a lehetőség is **kizárólag néhány alkalommal** vehető igénybe a félév során.

1.2. Zárthelyi dolgozat

A félév során **egy zárthelyi dolgozat** lesz, melyben egy **programozási feladatot** kell megoldani **számítógépnél**. A dolgozatírás egy adott hét gyakorlatain történik. A dolgozatot minden egyes hallgató annak a gyakorlatnak az időpontjában kell, hogy megírja, melyet a Neptun szerint látogat. (A dolgozat egy másik gyakorlati csoportban történő megírására a gyakorlatvezető(k) kivételes esetben adhat(nak) engedélyt.) A dolgozatírás hete később kerül kihirdetésre.

1.3. Önálló projekt

A félév során minden hallgató egy, a tárgy oktatójával **előzetesen egyeztetett** témájú projektet kell, hogy megvalósítson. A projekt tartalmi követelményei:

- Tartalmazzon olyan funkcionalitást (üzleti logikát), amely **tesztelhető**, és ehhez a tesztek is el kell készíteni.
- Rendelkezzen értelmes **grafikus felhasználói felülettel**, melynek megvalósítása a **JavaFX** keretrendszerrel történjen.
- A megvalósítás a **modell-nézet-vezérlő** (MVC) architekturális tervezési minta szerint kell, hogy történjen.
- Az adatok kezelését XML vagy JSON formátumban, vagy relációs adatbázisban a Java Persistence API (JPA) révén valósítsa meg. (Kezeljen a program inputot és outputot is valamelyik technológia segítségével.)

A projekteket az órákon kívül, **önállóan** kell megvalósítani. Bemutatásuk és értékelésük a vizsgaidőszak első hetében (május 29. és június 2. között) történik, ehhez időpontot foglalni a szorgalmi időszak utolsó hetében lehet majd a gyakorlatvezetőnél.

A bemutatás során a hallgató számot kell, hogy adjon arról, hogy a projekt a **saját munkája**, és hogy ismeri a projekthez felhasznált fejlesztőeszközök használatát. Automatikusan elégtelen osztályzatot jelent nem saját munka beadása.

A kiemelkedő önálló projekteket készítő hallgatók **kedvezményben** részesülnek a jegymegajánló dolgozatnál és a szóbeli vizsgán. A félév végén minden gyakorlatvezető jelzi az előadónak, hogy mely hallgatókat tart érdemesnek a kedvezményre.

1.4. Gyakorlati aláírás szerzésének feltételei

A gyakorlati aláírás megszerzésének szükséges és elégséges feltételei:

- Legfeljebb 3 hiányzás a gyakorlatokról.
- A zárthelyi dolgozat legalább 60%-os teljesítése.
- Az önálló projekt elkészítése, sikeres bemutatása és megvédése.

A dolgozat javítására egyetlen alkalommal lesz lehetőség egy **órán kívüli időpontban**. Ha valaki nem írta meg a dolgozatot, akkor ez lesz egyben számára is a pótlási lehetőség. A javító zárthelyi dolgozat helye és ideje később kerül kihirdetésre.

Nincs lehetőség azonban sem elmulasztott projekt bemutatás utólagos pótlására, sem pedig annak megismétlésére sikertelen teljesítés esetén.

2. Vizsga

A tárgyból **szóbeli vizsgát** kell tenni, melynek előfeltétele a gyakorlati aláírás megszerzése. Egyetlen alkalommal lehetőség lesz a félévben a szóbeli vizsgára jogosult hallgatóknak egy írásbeli dolgozat megírására, melyre **megajánlott vizsgajegy**et kapnak. A jegymegajánló dolgozat időpontja később kerül kihirdetésre.

3. Etikai elvárások

A hallgatókkal kapcsolatos etikai normákra A Debreceni Egyetem Etikai Kódexe az irányadó. Az etikai normákat megsértők automatikusan nem kapnak gyakorlati aláírást/elégtelen vizsgajegyét kapnak a tárgyból, nevük az Informatikai Kar oktatói között nyilvánosságra kerül.

4. Ajánlott irodalom

- [1] Christian Bauer, Gavin King, and Gary Gregory. *Java Persistence with Hibernate*. 2nd ed. Manning, 2015. URL: <https://www.manning.com/books/java-persistence-with-hibernate-second-edition>.
- [2] Christian Crumlish and Erin Malone. *Designing Social Interfaces: Principles, Patterns, and Practices for Improving the User Experience*. 2nd ed. O'Reilly Media, 2015. URL: <http://www.designingsocialinterfaces.com/>.
- [3] Norman Fenton and James Bieman. *Software Metrics: A Rigorous and Practical Approach*. 3rd ed. CRC Press, 2014. URL: <https://www.crcpress.com/Software-Metrics-A-Rigorous-and-Practical-Approach-Third-Edition/Fenton-Bieman/p/book/9781439838228>.

- [4] Martin Fowler et al. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 1999. URL: <http://martinfowler.com/books/refactoring.html>.
- [5] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
- [6] Paul C. Jorgensen. *Software Testing: A Craftsman's Approach*. 4th ed. CRC Press, 2013. URL: <https://www.crcpress.com/Software-Testing-A-Craftsmans-Approach-Fourth-Edition/Jorgensen/9781466560680>.
- [7] Lasse Koskela. *Effective Unit Testing: A guide for Java developers*. Manning, 2013. URL: <https://www.manning.com/books/effective-unit-testing>.
- [8] Jeff Langr, Andy Hunt, and Dave Thomas. *Pragmatic Unit Testing in Java 8 with JUnit*. Pragmatic Bookshelf, 2015. URL: <https://pragprog.com/book/utj2/pragmatic-unit-testing-in-java-8-with-junit>.
- [9] Van Lindberg. *Intellectual Property and Open Source: A Practical Guide to Protecting Code*. O'Reilly, 2008. URL: <http://shop.oreilly.com/product/9780596517960.do>.
- [10] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008. URL: <https://www.pearsonhighered.com/program/Martin-Clean-Code-A-Handbook-of-Agile-Software-Craftsmanship/PGM63937.html>.
- [11] Glenford J. Myers, Corey Sandler, and Tom Badget. *The Art of Software Testing*. 3rd ed. Wiley, 2011. URL: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118031962.html>.
- [12] Dave Nicolette. *Software Development Metrics*. Manning, 2015. URL: <https://www.manning.com/books/software-development-metrics>.
- [13] Yogesh Prajapati and Vishal Ranapariya. *Java Hibernate Cookbook*. Packt Publishing, 2015. URL: <https://www.packtpub.com/application-development/java-hibernate-cookbook>.
- [14] Kishori Sharan. *Learn JavaFX 8: Building User Experience and Interfaces with Java 8*. Apress, 2015. URL: <http://www.apress.com/9781484211434>.

- [15] Ian Sommerville. *Software Engineering*. 10th ed. Pearson, 2015. URL: <https://www.pearsonhighered.com/program/Sommerville-Software-Engineering-10th-Edition/PGM35255.html>.
- [16] Mitesh Soni. *Jenkins Essentials*. Packt Publishing, 2015. URL: <https://www.packtpub.com/application-development/jenkins-essentials>.
- [17] Rod Stephens. *Beginning Software Engineering*. Wrox, 2015. URL: <http://www.wrox.com/WileyCDA/WroxTitle/Beginning-Software-Engineering.productCd-1118969146.html>.
- [18] Jenifer Tidwell. *Designing Interfaces: Patterns for Effective Interaction Design*. 2nd ed. O'Reilly Media, 2010. URL: <http://designinginterfaces.com/>.