

TANTÁRGYI TEMATIKA ÉS TELJESÍTÉSI KÖVETELMÉNYEK
2018/2019. II. FÉLÉV

<i>Cím</i>	<i>Programming 2</i>
<i>Tárgykód</i>	IVB305ANMI
<i>Heti óraszám: ea/gy/lab</i>	2 / 0 / 3
<i>Kreditpont</i>	5
<i>Szak(ok)/ típus</i>	Computer Science Engineering BSc
<i>Tagozat</i>	Full-time
<i>Követelmény</i>	Term mark
<i>Meghirdetés féléve</i>	2018-19/2
<i>Előzetes követelmény(ek)</i>	Programming I
<i>Oktató tanszék(ek)</i>	Systems and Software Technologies
<i>Tárgyfelelős és oktatók</i>	Tamas Storcz

COURSE OBJECTIVES

The purpose of this course is to introduce the students to the fundamental concepts of object-oriented paradigm and event driven programming in Java environment. While solving application development tasks, meet the modern design and development methods, tools and their integration into development process and teamwork.

COURSE CONTENTS

Short description:

Students would see generalization and transition of their previously acquired programming knowledge. They would meet and learn the principals of object oriented programming, the most commonly used programming paradigm. With appropriate tools, these principals would be applied to produce sustainable source code, even with team work. The iterative application of OOP principles is done in the most popular environment: Java.

Students during their teamwork (as homework) practice programming skills, learn to professionally evaluate own and college's work. Finally they would see a method to measure developer performance.

Issues:

Lecture:

1. Java history, platform structure, comparison of basic elements of C and Java language
2. Object oriented programming – encapsulation, often used types
3. Static components, arrays, often used classes: handling the console, Math, Random
4. Memory management, (Garbage Collector), Source version management

5. Methods in detail, Inheritance, Polymorphism, Class cast
6. Type parameter, generics, Agile development
7. Abstract classes, interfaces, cohesion and decoupling
8. Exception handling, text file management, logging, serialization
9. Frameworks, IoC, DI, Unit tests, dependency simulation, compilation tools (MAVEN)
10. GUI, event driven programming, common controls
11. Accessing database, Design patterns
12. Threading, concurrency
13. Theoretical exam

Lab:

1. Java and IntelliJ IDEA installation and handling; short examples, procedural implementation in Java, running, debugging
2. Designing and implementing objects, code refactoring, OOP implementation of sample code
3. Static classes and components, array creation, fulfilment and management, using console
4. git repository management, integration, issue tracking, peer-review, merge request
5. Method overload, accessibility, inheritance and polymorphism, class cast
6. Using type parameter, create generic classes, using generic collections, backlog, scrum poker, burndown
7. Creation of abstract methods and classes, define and implement interfaces, refactor code for decoupling
8. First practical exam
9. Handle possible exceptions while accessing data in files, write logs, serialize data
10. Creating and using frameworks, unit tests using JUnit and Mockito
11. using GUI, define custom events and event handlers, trigger event
12. Accessing H2 database with JDBC
13. Second practical exam

EXAM AND EVALUATION

Participation: Lecture 70%, Laboratory 80%

Conditions of teacher's signature:

- required participation
- average of 3 exams above 35%
- working homework submitted in time

Conditions of term mark:

- teacher's signature
- theoretical and second practical exams both above 50%;

Final exam: no

Evaluation:

Obtainable in the semester: 110%

Weights of activities:

- Theoretical exam 40%
- First practical exam 10%
- Second practical exam 30%
- Tests on practice (total) 10%
- Homework 10%
- Tests on lecture max. +10%

Calculation of term mark:

with the conditions above, the weighted average of activities

- 0%-35%: signature denied
- 36%-50%: 1 (fail)
- 51%-62%: 2 (pass)
- 63%-74%: 3 (satisfactory)
- 75%-86%: 4 (good)
- 87%-110%: 5 (excellent)

Retake skipped or failed exam:

- When using prohibited support, exam terminated and counted as 0%
- Test cannot be retaken
- Skipped exam counted as 0%
- Retaking exam will contain the contents of the whole semester (replaces all 3 exams with weight of 70%)
- Retaking exam will be in the first two weeks of the exam period

RECOMMENDED LITERATURE

- [1st] Java in Nutshell: 6th edition, D. Flanagan, O'Reilly, 2014
- [2nd] An Introduction to Object-Oriented Programming, T. Budd, Addison-Wesley, 2004.
- [3rd] Design Patterns: Elements of Reusable Object-Oriented Software, E. Gamma, R. Helm, R. Johnson and John Vlissides, Addison Wesley, Reading, Mass., 1995.
- [4th] The Unified Modeling Language Reference Manual, J. Rumbaugh, I. Jacobson, G. Booch, Addison-Wesley, 1999

ÜTEMEZÉS

		SZORGALMI IDŐSZAK, OKTATÁSI HETEK															VIZSGAIDŐSZAK						
2018/2019. II. FÉLÉV		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	1.	2.	3.	4.	5.		
Előadás tematika sorszáma			1	2	3	4	5	6	7	8		9	10	11	12	13			Aláírás, félévközi jegy már nem pótolható				
Gyakorlat/Labor sorszáma			1	2	3	4	5	6	7	8		9	10	11	12	13							
Zárhelyi dolgozat										P1					T P2								
Otthoni munka	kiadása			1		2		3				4											
	beadási határidők													1-4									
Jegyző- könyvek	beadási határidők																						
Egyebek	pl. beszámolók,																						
	stb.																						
Aláírás / Félévközi jegy megadása																a /fj							
Vizsgák tervezett időpontjai																							

2019. február 05.

.....
tantárgyfelelős