

Fișa disciplinei

1. Date despre program

1.1. Instituția de învățământ	ACADEMIA DE STUDII ECONOMICE
1.2. Facultatea	CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ
1.3. Departamente	(Departament) INFORMATICĂ ȘI CIBERNETICĂ ECONOMICĂ
1.4. Domeniul de studii	Cibernetică, statistică și informatică economică
1.5. Ciclul de studii	LICENȚĂ
1.6. Forma de învățământ	CU FRECVENȚĂ
1.7. Programul de studii	Informatică economică
1.8. Limba de studiu	Română
1.9. Anul universitar	2019-2020

2. Date despre disciplina

2.1. Denumire	Programare multiparadigmă - Java									
2.2. Cod	19.0205IF2.2-0005									
2.3. Anul de studii	2	2.4. Semestrul	2	2.5. Forma de evaluare	Examen	2.6. Regimul disciplinei	O (Obligatori)	2.7. Nr. credite ECTS	4	
2.8. Titulari	C(C)	conf.univ.dr. TOMA Cristian -Valeriu					cristian.toma@ie.ase.ro			
	C(C)	conf.univ.dr. FURTUNĂ Titus Felix					felix.furtuna@ie.ase.ro			
	C(C)	conf.univ.dr. IONIȚĂ Cristian					cristian.ionita@ie.ase.ro			
	S(S)	conf.univ.dr. TOMA Cristian -Valeriu					cristian.toma@ie.ase.ro			
	S(S)	conf.univ.dr. FURTUNĂ Titus Felix					felix.furtuna@ie.ase.ro			
	S(S)	lect.univ.dr. IANCU M BOGDAN					bogdan.iancu@ie.ase.ro			
	S(S)	conf.univ.dr. IONIȚĂ Cristian					cristian.ionita@ie.ase.ro			
	S(S)	Asociat drd. CARTAS C COSMIN - GABRIEL					cosmin.cartas@csie.ase.ro			
	S(S)	lect.univ.dr. VINȚE C CLAUDIU					claudiu.vinte@ie.ase.ro			
	S(S)	Asociat drd. MARALES D RĂZVAN - CRISTIAN					razvan.marales@csie.ase.ro			

3. Timp total estimat

3.1. Număr de săptămâni	14.00
3.2. Număr de ore pe săptămână	4.00 din care
	C(C) 2.00
	S(S) 2.00
3.3. Total ore din planul de învățământ	56.00 din care
	C(C) 28.00
	S(S) 28.00
3.4. Total ore de studiu pe semestru (număr ECTS*25 ore)	100.00
3.5. Total ore studiu individual	44.00

<i>Distribuția fondului de timp pentru studiu individual</i>	
Studiu după manual, suport de curs, bibliografie și notițe	20.00
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	8.00
Pregătire seminarii, laboratoare, teme, referate, portofolii și eseuri	10.00
Tutoriat	3.00
Examinări	3.00
Alte activități	

4. Preconții

4.1. de curriculum	Algoritmi in programare, Structuri de date, Programare orientata obiect
4.2. de competențe	Programare orientata obiect - C++, elemente de sisteme de operare si compilatoare, inetrpretoare, -masini virtuale

5. Condiții desfășurare activități

pentru C(C)	Prelegerile de curs se desfasoara în sali cu echipament de predare multimedia.
pentru S(S)	Seminariile se desfasoara în sali cu acces la Internet si PC. Mediul de dezvoltare utilizat este JDK cu IDE Eclipse sau IntelliJ IDEA, si masini virtuale Linux Ubuntu OS sau nativ MS Windows OS, plus tool-uri asociate.

6. Competențe specifice acumulate

PROFESIONALE	C4	Dezvoltarea de componente pentru produse software, folosind structuri de date, algoritmi , tehnici și limbaje de programare evaluate
PROFESIONALE	C5	Dezvoltarea de aplicații informatice care utilizează baze de date, resurse multimedia și tehnologii client-server/servicii web

7. Obiectivele disciplinei

7.1. Obiectivul general	Transferul tehnologic de la profesor/universitate catre studenti a cunosntelor teoretice si practice cu privire la dezvoltarea aplicatiilor software si a programarii multi-paradigma, precum si dezvoltarea capacitatii studentilor de rezolvarea a problemelor practice prin elaborarea de programe dezvoltate in regim orientat obiect si multi-paradigma, open source si cross-platform utilizand platforma Java.
7.2. Obiective specifice	-Elaborarea de programe ce rezolva probleme practice de catre studenti prin utilizarea de noțiuni specifice de programare orientata obiect si multi-paradigma; -Înșușirea de către studenți a sintaxei limbajului Java și a elementelor de logica a elaborarii programelor orientate obiect in regim open source si open-source cross-platform.

8. Conținuturi

8.1. C(C)		Metode de predare/ lucru	Recomandări
1	1. Explicare notiuni elementare JVM, JDK, JRE si compilare in linie de comanda si cu unelte facile pentru procesul de automatizare – ANT/MavenGradle si IDE Eclipse/Netbeans/IntelliJ IDEA. Introducere in sintaxa Java SE (if, switch, for, while), metode, masive, POO - Programare Orientata Obiect si Similitudini cu C++ / Ubuntu Linux sau MS Windows, rulare Java byte-code class fara modificari / recompilari pe diverse sisteme de operare. Lista tipurilor de programare multi-paradigma.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.

2	2. Masive in Java si Introducere POO - clase, obiecte/instante, clase abstracte, interfete. Alternative de instalare si rulare din linia de comanda a claselor si librariilor - JAR si prin unelte specializate de automatizare ANT/Maven/Gradle si Jenkins.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
3	3. Java POO - Clasa, Obiectul + Obiecte nemodificabile, Interfața, Clasa abstractă, Mecanismul de moștenire/Derivare, Polimorfism, relația de tip compoziție "are" versus "este un", interfața ca tip de date, analogie cu C++ polimorfism "pur", clasa ClassCastException + teme din Robocode.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
4	4. Programare utilizand mecanisme speciale in Java, precum Java Generics si JCF - Java Collection Framework: containere, iteratori si algoritmi. JCF – vectori, liste, tabele hash, etc.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
5	5. Java I/O - Input / Output la nivel de fluxuri de caractere si octeti, clasa File si RandomAccessFile. Java Native Interface - JNI (pentru intelegerea metodelor native) si adnotari plus introspectie. Introducere minimala in testare unitara JUnit 4 ca exemplificare a mecanismelor de adnotare si introspectie.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
6	6. Mecanismul Java I/O de serializare, adnotari si introspectie plus JNI. Mecanismul de exceptii si erori plus doua modele de arhitecturi de cod sursa - Factory Methods & Singleton.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
7	7. Caracteristici Java 8 si paradigma programarii functionale: Nashorn JavaScript Engine, Call-back & Inner Classes, Method references, default method, lambda expressions + functional interfaces, processing streams, Optional, Date/Time, & new API (Base64).	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
8	8. Multi-fir versus multi-proces, modele de tip Multi-fir de executie, caracteristici ale paradigmelor de programare concurenta si paralela, operatii atomice, fire de executie JVM si OS; Java API pentru programarea multi-fir, Singleton vs. Immutable Objects - Software Design Patterns; Parallelism/HPC - High Performance Computing.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.

9	9. Elemente avansate de programare multi-fir de executie (java.util.concurrent - ExecutorService + Future-Callable + Lock/Semaphore + Producer/Consumer) si expresii lamda Java 8 pentru mecanismele multi-fir de executie.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
10	10. Java in retele de calculatoare: TCP vs. UDP / IP cu Java Socket. Caz de utilizare TCP cu multi-fir pentru implementarea protocolului FTP si HTTP.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
11	11. Java NIO (New Input/Output), RegEx (Regular Expressions) si module JDK 9 + Caracteristici aditionale Java 9 (ex. HTTP2 Client, try-catch, etc.)	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
12	12. Java XML (JAXB2) si JSON Parsing (JSON.org / Jackson).	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
13	13. Programarea bazelor de date in Java - JDBC si NoSQL.	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.
14	14. Introducere in programarea vizuala GUI Java FX si paradigma programarii orientate pe evenimente. Java FX 8 (MVP - Model View Presenter).	Prelegere bazata pe prezentari multimedia si interactiune cu studentii.	- Suportul de curs este postat pe platforma programului de licenta. - Se recomandă studentilor parcurgerea prealabilă a suportului de curs pentru a putea interacționa mai ușor cu profesorul în timpul predării.

Bibliografie

- Jonathan Knudsen, Patrick Niemeyer, Learning in Java, O'Reilly, 2005, Statele Unite ale Americii
- Bruce Eckel, Thinking in Java, Prentice Hall, 2003, Statele Unite ale Americii

8.2. S(S)

Metode de predare/ lucru

Recomandări

1	1. Exemplu practic pentru implementarea conceptelor de clasa, obiect, copie "superficiala" vs. copie "completa", in Java.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
2	2. Exemplu practic pentru lucru cu masive in Java si utilizarea librariilor JAR in Java.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
3	3. Exemplu practic pentru mostenire si derivare, polimorfism, interfata in Java.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
4	4. Exemplu practic pentru structuri de date in Java si JCF - Java Collection Framework cu List<E> si Map<K,V>	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
5	5. Exemplu practic pentru Java IO DataInputStream, DataOutputStream and RandomAccessFile.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
6	6. Exemplu pentru serializarea obiectelor in Java.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
7	7. Exemplu practic pentru testarea interfetelor functionale si expresiilor lambda.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
8	8. Exemplu practic pentru programarea concurenta utilizand elemente multi-fir pentru accesul la resurse comune.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.

9	9. Programare multi-fir standard si avansata, elemente Executor/Service si Future-Callable din pachetul java.util.concurrent pentru adunarea elementelor unui vector/a doi vectori in paralel.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
10	10. Exemplu practic pentru programare Socket UDP uni-cast/multi-cast/broad-cast.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
11	11. Exemplu de tip Socket TCP in Java pentru implementarea unui simplu server de web/HTTP.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
12	12. Exemplu pentru utilizarea XML and JSON.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
13	13. Exemplu practic pentru programarea bazelor de date relationale (MySQL / SQLite)prin JDBC API si celor de tip NoSQL (MongoDB) prin librerie MongoDB JAR.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.
14	14. Exemplu practic pentru programarea interfetelor grafice si orientata pe evenimente in Java FX.	Interactiune cu studentii in vederea dezvoltarii aplicatiei si analizei programului dezvoltat la seminar	Se recomandă studentilor parcurgerea suportului de curs si seminar/laborator din platforma a programului de licenta pentru a putea interacționa mai ușor cu profesorul.

Bibliografie

- Jonathan Knudsen, Patrick Niemeyer, Learning in Java, O'Reilly, 2005, Statele Unite ale Americii
- Bruce Eckel , Thinking in Java – 3rd Edition, Prentice Hall, 2003, Statele Unite ale Americii

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

Conținutul disciplinei este discutat cu specialiști și cu personalități din universități și companii de marcă ce activează în domeniul IT&C.

10. Evaluare

Activitatea	Criterii de evaluare	Metode de evaluare	Pondere în nota finală
10.1. S(S)	Evaluare teste si/sau studiu de caz individual	Testare pe platforma electronica de e-learning si PC-uri	30.00
10.2. Evaluare finală	Evaluarea cunoștințelor acumulate in EXAMEN final in sesiune.	Testare pe platforma electronica de e-learning si PC-uri Nota 5 din 10 este obligatorie pentru promovare.	70.00
10.3. Modalitatea de notare	Note întregi 1-10		
10.4. Standard minim de performanță	<p>Cumulativ, pentru intrare in examen si promovare trebuie indeplinite mai multe conditii obligatorii:</p> <ul style="list-style-type: none"> - Nota din examenul final <=> evaluarea finala trebuie sa fie minim 5 din 10. - Nota finala calculata ca podere 70% examen si 30% nota de la seminar sa fie mai mare decat 5 din 10. - Testul/Testele din cadrul seminarului/laboratorului trebuie sustinut(e) de student; nu p[oaate intra in examen studentul care nu are testul dat. - Proiectul trebuie sustinut in timpul semestrului si trimis cadrului didactic conform conditiilor afisate si conform termenelor limita stabilite - cadrul didactic de la seminar poate inlocui proiectul cu un test la calculator. - Fiecare student trebuie sa aiba o prezenta la seminar de minim 75% (11 seminarii din 14). - Cel putin jumătate din sarcinile tehnice primite la curs si seminar trebuie trimise conform conditiilor si termenelor limita stabilite. <p>Nota 5:</p> <p>Dezvoltarea de aplicatii si programe in Java, precum si compilare si rulare in linia de comanda, mediu IDE si depanare CLASS/JAR, plus dezvoltarea unor clase complete care au constructor, implementeaza metode din interfețele Cloneable, Comparable si suprascierea metodelor din clasa Object in pachete (package-uri) with the following details:</p> <ul style="list-style-type: none"> - Dezvoltarea de proiecte Java - stand-alone, librării (JAR) si module JDK9 in linie de comanda cu JDK (8 & 9) si IDE - ex. Eclipse - Dezvoltarea de clase si instante Java in concordanta cu cerintele si logica din biletul de examen - Contruktor fara parametrii si constructori cu parametrii - Suprascierea metodelor mostenite din clasa Object: clone, equals, hashCode si toString - Metode de acces: get/set - Masive in Java: uni-dimensionale si multidimensionale - Clase Java prin compozitie (relatia "has a") si mostenire (relatia "is a") plus polimorfism - Implementarea interfetelor si claselor abstracte + obiecte imutabile si interfata Comparable (metoda compareTo) + suprascierea metodei hashCode - Java I/O Streams - byte & d char oriented; - Mecanismul de serializare si deserializare - Tratarea exceptiilor: try {...} catch() {...} si "throws" - Java Generics & Reflection + Annotations (si Meta-annotations) - Java Collection Framework: containere, iteratori si algoritmi - java.util.* - Multi-threading pentru paralelism si concurenta (incl. java.util.concurrent - Executor/Services & Callable/Future plus Java 8 Fork-Join utilizat in programarea paralela - HPC = High Performance Computing) - Network Programming (TCP, UDP si TCP cu muti-threading) + serializare/deserializare in cadrul conexiunilor TCP/IP - Depanare la executie si intelegerea memoriei masinii virtuale Java (ex. deep-copy versus shallow-copy) <p>- Additional 1 punct:</p> <ul style="list-style-type: none"> - Simple Software Design Patterns: Singleton & Factory Methods - Programare Functionala si expresii Lambda in Java 8+ <p>- Additional 1 punct:</p> <ul style="list-style-type: none"> - Conlucrarea cu baze de date: JDBC (Java Data-Base Connectivity) folosind SQLite / MySQL / Oracle si NoSQL (ex. MongoDB) <p>- Additional 1 punct:</p> <ul style="list-style-type: none"> - XML Parsing (JAXB2), JSON encoding/decoding (third-party libraries cu JSON.org/Jackson) si RegEx <p>- Additional 1 punct:</p> <ul style="list-style-type: none"> - Java NIO - New Input Output 		

- Additional 1 punct:
- GUI - Java Swing / Java FX 8 + call-back si MVP (Model View Presenter) architectural design pattern (Java FX - FXML).

* Extra:
- Caracteristici avansate JVM / Gradle, Maven, ANT pentru building automation - optional / Profiling & Test Intro (JUnit) - optional
- Optimizarea memoriei si algoritmilor utilizati
- Implementari inovative ale cerintelor din bilet

* Nota: Baremul are sens pe aplicatii/programe ce trateaza logic problema enuntata in subiectul de lucrare si pe cod sursa care compileaza si ruleaza. Cadrul didactic de la evaluare poate alege rescrierea programelor din bilet, eventual cu modificari pentru asigurarea unei evaluari corecte si unitare.

Data listării,
05/03/2020

Titulari,

Data avizării în departament,

Director departament,