

# CURSURI OPȚIONALE DE INFORMATICĂ

## PROPUSE

PENTRU ANUL UNIVERSITAR 2020-2021

DOMENIUL DE LICENȚĂ: MATEMATICĂ

SPECIALIZAREA:  
MATEMATICĂ- INFORMATICĂ

<i>Nr.</i>	<i>Denumire curs</i>	
1	Blockchain – concepte, tehnologii și aplicații	<i>Semestrul II</i>
2	Inițiere în bioinformatică și cercetare	
3	Introducere în prelucrarea limbajului natural	
4	Invatare automata in vedere artificiala(*)	
5	Învățarea rețelelor neurale adânci (*)	
6	Prelucrarea limbajului natural in python cu NLTK si SPACY	
7	Programare dispozitive Android (*)	
8	Programare dispozitive iOS (*)	
9	Protocoale criptografice	
10	Securizarea și automatizarea rețelelor	
11	Tehnici de compilare	
12	Tehnici de programare a aplicațiilor grafice	
13	Testarea sistemelor software	

**NOTĂ:** Cursurile opționale vor fi numerotate de către studenți în ordinea preferințelor, de la 1 la 13.

## FIȘA CURSULUI

**Titlu: Blockchain–concepte,tehnologiișiaplicații**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: examen

Credite: 5

### **OBIECTIVE:**

1. Gartner include Blockchain printre primele 10 tehnologii strategice din 2020! De aceea este importantă familiarizarea studenților în informatică (și nu numai) cu această tehnologie de stocare și management al datelor descentralizate și imutabile, care valorifică ideea de *consens* peste un *ledger* distribuit.
2. Familiarizarea studenților cu conceptele Blockchain ca structură de date ce implică existența unei familii de blocuri înlanțuite, confirmate prin intermediul unui mecanism de consens PoW, care găzduiesc tranzacții între nodurile rețelei P2P.
3. Tehnologia în varianta ei inițială propusă de Satoshi Nakamoto ca un "public ledger" global pentru stocarea istoricului tranzacțiilor într-o criptomonedă (Bitcoin) coincidea cu Blockchain 1.0, iar aceasta a evoluat la Blockchain 2.0, care revenea contractelor "smart", iar acum tehnologia Blockchain 3.0, este un model de organizare de spectru larg pornind de la genomică și ajungând la managementul sistemelor sociale de asistență pentru sănătate publică la nivel global, care și-a câștigat o poziție centrală în tabloul tehnologic actual.
4. Cursul propus se structurează pe blockchain-ul ETHEREUM (a doua generație de blockchain) ca platformă pentru dezvoltarea de aplicații descentralizate în limbajele Solidity sau Go.
5. Pe lângă prezentarea conceptelor aferente sistemului de tehnologii Blockchain, cursul va pune accentul și pe unele aspecte practice, prezentând framework-uri pentru dezvoltarea de aplicații descentralizate încercând să pună în evidență potențialul tehnologic al acestor platforme.

### **PROGRAMĂ:**

1. Arhitectură pentru calcul descentralizat. IPFS (o combinație de Kademia + BitTorrent + Git), NeoFS și FileCoin ca stocare asociată blockchain-ului.
2. Introducere în tehnologia Blockchain. Storage descentralizat. Blockchain public vs. Blockchain privat (Bitcoin, Ethereum, Corda, Hyperledger).
3. Consensul în Blockchain-PoW (Ethereum), PoS (Ethereum 2.0), PBFT (Hyperledger), PoET
4. Platforme Blockchain: BigChainDB, Hyperledger Fabric, ChainCore, IBM Bluemix Blockchain
5. Contracte și proprietăți "smart" peste blockchain, implementate prin intermediul platformelor (Ex. Truffle și Remix implementează un contract generic)
6. Limbaj pentru implementarea contractelor "smart" pe diferite platforme cu Solidity și Go
7. Platforme de execuție a contractelor smart ETHEREUM-EVM (Mașina virtuală Ethereum).
8. Scalabilitatea Blockchain: SEGWIT vs Gas. Testare și verificare automată a smart contracts în Solidity.
9. Arhitectura aplicațiilor descentralizate.
10. Elementele de securitate în blockchain.

### **BIBLIOGRAFIE:**

1. Dannen, Cris. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners.*: APress, 2017.

2. Antonopoulos, Andreas M. Mastering Bitcoin. Programming the Open Blockchain. O'Reilly, 2017.
3. Asharaf, S. and Adarsh, S. Decentralized Computing Using Blockchain Technologies and Smart Contracts: Emerging Research and Opportunities. 2017.
4. Lewis, Christopher. Blockchain: your comprehensive guide to understanding the decentralized future. 2016.
5. Raval. Decentralized Applications Harnessing Bitcoin's Blockchain Technology. s.l.: O'Reilly, 2016.

Cursul va fi susținut de către Conf.dr. Cristian Kevorchian și Lect.dr. Iulia Banu-Demergian

## FIȘA CURSULUI

Titlu: **Inițiere în bioinformatică și cercetare**

Domeniul de licență: informatică (matematică)

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5 (3)

### **OBIECTIVE:**

Cursul se adresează studenților interesați în bioinformatică cu accent pe noțiuni de cercetare. Studenții vor fi expuși atât la elemente de bază din bioinformatică cât și la rezultate recente din bioinformatică și biologia sistemelor (virusi, bacterii e.g. COVID-19). Studenții vor dobândi la acest curs și experiență în cercetare: cum se citește o lucrare, structurarea lucrărilor științifice, cum se scrie o lucrare științifică, modelul de recenzie al lucrărilor științifice: peer-review, clasificarea revistelor și a conferințelor.

### **PROGRAMĂ:**

1. Noțiuni elementare de biologie celulară, genetică și biochimie.
2. Virusul COVID-19 descriere și detalii
3. Experimentul lui Adleman pentru rezolvarea unei probleme NP completa folosind ADN și unelte biochimice.
4. Experimentul lui Lipton pentru rezolvarea SAT, alte rezultate din calculabilitatea bazată pe ADN.
5. Autoasamblare: experimentele lui Eric Winfree, Ned Seeman, Paul Rothemund, etc.
6. Alinierea secvențelor (sequence alignment): Algoritmii optimali de aliniere pentru secvențe ADN sau proteine, pentru aliniere globală sau locală: Smith-Waterman, Needleman- Wunsch. Algoritmii heuristici pentru aliniere: Blast și variante, Fasta, PatternHunter, PatternHunter2.
7. Biologie evoluționară: Aliniere afină, matrici de substituție: BLOSUM50, BLOSUM62, PAM150, aliniere multiplă.
8. Modele abstracte de calculabilitate bazate pe ADN și celule: Sisteme H, Sisteme P.
9. Introducere în cercetare: revista, articol, peer-review, referințe, cuvinte cheie, factor de impact pentru revista, LaTeX, ordinea autorilor, structurarea articolelor, elemente introductive despre redactarea tehnică.

### **BIBLIOGRAFIE:**

1. Xu Z, Shi L, Wang Y, Zhang J, Huang L, Zhang C, Liu S, Zhao P, Liu H, Zhu L, Tai Y. Pathological findings of COVID-19 associated with acute respiratory distress syndrome. *The Lancet Respiratory Medicine*. 2020 Feb 18.
2. Leonard M. Adleman, Molecular computation of solutions to combinatorial problems, *Science*, Vol. 266, Iss. 5187, 1994, pp. 1021-1024.
3. Richard J. Lipton, DNA Solution of Hard Computational Problems, *Science*, Vol. 268, Iss. 5210, 1995, pp. 542-545.
4. Erik Winfree, Furong Liu, Lisa A. Wenzler, Nadrian C. Seeman, Design and self-assembly of two-dimensional DNA crystals, *Nature*, Vol. 394, 1998, pp. 539-544
5. Paul W. K. Rothemund, Folding DNA to create nanoscale shapes and patterns, *Nature*, Vol. 440, 2006, pp. 297-302
6. Arthur M. Lesk, *Introduction to Bioinformatics*, Oxford University Press, 2002, 290 pp.

7. Neil C. Jones, Pavel A. Pevzner, An Introduction to Bioinformatics Algorithms (Computational Molecular Biology), The MIT Press, 2004, 456 pp., ISBN: 0262101068.
8. Gheorghe Paun, Grzegorz Rozenberg, Arto Salomaa, The Oxford Handbook of Membrane Computing, Oxford University Press, 2010, 696 pp., ISBN: 0199556679.
9. Temple F Smith, Michael S Waterman, Comparison of biosequences, Advances in Applied Mathematics, Vol 2, Iss. 4, 1981, pp. 482-489.
10. Saul B. Needleman, Christian D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology, Vol. 48, Iss. 3, 1970, pp. 443-453.
11. 10: Steven Henikoff, Jorja G. Henikoff, Amino acid substitution matrices from protein blocks, PNAS, Vol. 89, 1992, pp. 10915-10919.
12. Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, David J. Lipman, Basic local alignment search tool, Journal of Molecular Biology, Vol. 215, Iss. 3, 1990, pp. 403-410.
13. Bin Ma, John Tromp, Ming Li, PatternHunter: faster and more sensitive homology search, Bioinformatics, Vol. 18, Iss. 3, 2002, pp. 440-445.

Cursul va fi susținut de către Prof.univ.dr. Andrei Paun.

## FIȘA CURSULUI

**Titlu: Introducere în prelucrarea limbajului natural**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

### **OBIECTIVE:**

Cursul își propune să ofere studenților o perspectivă modernă și la zi asupra domeniului. Fiecare temă abordată va fi însoțită de aplicații concrete care vor viza diverse aspecte ale limbajului natural, cu o atenție specială acordată limbii române. Vor fi abordate aspecte computaționale, cantitative și formale ale limbajului natural, făcând distincția clară între analiza textelor și cea a limbilor naturale în general. Studenții vor lua contact cu temele prioritare de cercetare în domeniu, în scopul deschiderii acestora spre doctorat. Vor fi prezentate cerințe punctuale ale diverselor firme interesate de aplicații ale procesării limbajului natural în chestiuni curente de producție.

### **PROGRAMĂ:**

- [1] Probleme curente (practice, teoretice și de cercetare) în lingvistica matematică și computațională și în procesarea limbajului natural.
- [2] Aspecte cantitative ale limbajului natural.
- [3] Probleme de similaritate lingvistică.
- [4] Abordări și rezolvări computaționale eficiente ale unor probleme de morfologie, fonologie și semantica.
- [5] Analiza computațională a amprentei stilistice.
- [6] Analiza computațională a documentelor.
- [7] Analiza de corpus. Detectarea de colocatii, detectare automată și analiza ortografică a cuvintelor înrudite.
- [8] Traducere automată.
- [9] Detectarea automată din texte a opiniilor, analiza computațională a polarității sentimentelor. Aplicații în predicția opțiunii politice din analiza discursurilor.

### **BIBLIOGRAFIE:**

- [1] D Jurafsky, JH Martin. Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition 2nd ed., 2015, Prentice Hall
- [2] G. Altmann (ed). Handbook of Quantitative Linguistics, 2003
- [3] R. Dale (ed) The Handbook of Natural Language Processing, Marcel Dekker, New York, 2000.
- [4] Liviu P. Dinu. Rank distance and applications. Ed. Universității București, 2011
- [5] Alina Maria Ciobanu, Liviu P. Dinu, 2014. An Etymological Approach, to CrossLanguage Orthographic Similarity. Application on Romanian. In Proc. EMNLP 2014, p1047–1058.
- [6] Manning, C., H. Schütze. Foundations of statistical natural language processing, MIT Press, 1999
- [7] Mosteller, Frederick and David L. Wallace. Inference and Disputed Authorship. Distributed for the Center for the Study of Language and Information. 320 p. 6 x 9 Series: (CSLI-DHS) Center for the Study of Language and Information - The David Hume Series, 2007
- [8] Arhivele revistelor: Computational Linguistics, Literary and Linguistic Computing, Quantitative Linguistics

[9] Volumele conferintelor sustinute de ACL (ACL, EMNLP, EACL, NAACL, COLING, CICLING, RANLP). Disponibile on-line la <http://aclweb.org/anthology-new/>

Cursul va fi susținut de către Prof.dr. Liviu Dinu.

## FIȘA CURSULUI

Titlu: **Invatare automata in vedere artificiala**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 1; laborator = 2)

Semestrul: 6 / anul III de studiu

Forma de examinare: Verificare

Credite: 5

### **OBIECTIVE:**

Acest curs își propune prezentarea celor mai noi tehnologii din domeniul Învățării Automate în contextul prelucrării de imagini.

Accentul cursului va fi pus pe latura practică, axată cu predilecție pe seturi de date compuse din imagini.

Vor fi prezentate arhitecturi esențiale de rețele neuronale convoluționale, folosite cu succes în obținerea unor rezultate remarcabile pentru diverse probleme de Vedere Artificială: recunoașterea obiectelor dintr-o imagine, auto-tagging și altele. Studenții vor lucra cu framework-uri software moderne utilizate în prezent în Învățare Automată: PyTorch

Folosind cunoștințele acumulate pe parcurs, studenții vor realiza până la finalul cursului un proiect pe o tema de vedere artificiala, continuând un pipeline intreg de invatare automata, cu tot lifecycle-ul: antrenare, evaluare, optimizare, discutarea rezultatelor.

### **PROGRAMĂ:**

1. Introducere. Concepte de bază în Învățare automată.
2. Clasificarea imaginilor. Data-Driven Approach. Rețele neuronale convoluționale.
3. Antrenarea rețelelor. Backpropagation. Metode de optimizare.
4. Software pentru deep learning: Pytorch
5. Transfer learning. De la clasificarea imaginilor la generare de imagini.
6. Aplicații ale rețelelor convoluționale: detectarea obiectelor, segmentarea imaginilor, rețele generative si aplicatiile lor.

### **RECOMANDARI:**

1. cunostinte algebra liniara, structuri de date
2. cunostinte de baza de Python
3. cunostinte de baza numpy

### **BIBLIOGRAFIE:**

1. Christopher M. Bishop, Pattern Recognition and Machine Learning. Springer, 2010.
2. [Yoshua Bengio, Ian Goodfellow, Aaron Courville, Deep Learning. The MIT Press, 2016](#)
3. Imagenet: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
4. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd, Pearson, 2009
5. CS231n: Convolutional Neural Networks for Visual Recognition [Stanford]



(\*) Cursul va fi susținut de către specialiști de la firma Arnia Software.

## FIȘA CURSULUI

**Titlu: Învățarea rețelelor neurale adânci (\*)**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

### **OBIECTIVE:**

Cursul este orientat spre practică. De-a lungul acestuia, studenții vor învăța să recunoască tipurile de probleme care se pretează rezolvării cu rețele neurale și vor putea să aleagă cu ușurință abordarea potrivită conform celor mai bune soluții de la momentul actual. Până la final, studenții vor putea să extindă aceste soluții cu idei proprii și vor lucra confortabil în PyTorch. Aplicațiile practice sunt în zona de imagini și de limbaj natural.

Punem accent însă și pe componenta teoretică. În afara scurtelor pastile teoretice din cursuri, în cadrul temelor există bonusuri cu caracter teoretic. Vom ghida studenții interesați către proiecte specifice.

### **EVALUARE:**

Teme 35%, proiect 40%, examen 25%. În cadrul laboratoarelor, studenții vor lucra la tema în curs și vor discuta neclaritățile cu laborantului. Temele și proiectul se rezolvă în PyTorch. Proiectul constă în reproducerea arhitecturii, a configurației și a rezultatelor pentru un articol care oferă o soluție pentru o problemă de vedere computațională, limbaj natural sau învățare prin recompensă (dintr-un subset selectat de articole).

### **PROGRAMĂ:**

1. Introducere în învățarea automată. Concepte de bază (preprocesare, bias, varianță, selectare model, overfit, validare încrucișată, evaluare)
2. Antrenarea rețelelor neurale (forward și back-propagation, gradient, softmax, regularizare).
3. Optimizare convexă (geometria funcției de cost, metode de ordin 1 și 2, serie Taylor, saddle points).
4. Rețele convoluționale (arhitecturi clasice)
5. Aplicații în vederea computațională (cele mai bune arhitecturi și configurații pentru clasificare, detecție, segmentare)
6. Rețele recurente (RNN, LSTM, dispariția și explozia gradientilor)
7. Aplicații în limbaj natural (tokenizare, vocabular, embedding, rețele sequence-2-sequence, modulul de atenție)
8. Învățare prin recompensă (ecuațiile lui Bellman, explorare on/off policy, policy gradient, Q-learning, SARSA, MCTS)

### **BIBLIOGRAFIE:**

- Christopher M. Bishop. Pattern Recognition And Machine Learning
- Yurii Nesterov. Introductory Lectures on Convex Optimization
- CS231n: Convolutional Neural Networks for Visual Recognition [Stanford]
- CS224d: Deep Learning for Natural Language Processing [Stanford]
- Richard S. Sutton, Andrew G. Barto. Reinforcement Learning: An Introduction, 2017

(\*) Cursul va fi susținut de către specialiști de la compania Bitdefender

## FIȘA CURSULUI

Titlu: **Prelucrarea limbajului natural in python cu NLTK si SPACY**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

### OBIECTIVE:

**Procesarea limbajului natural** (*Natural Language Processing – NLP*) este un subdomeniu al inteligenței artificiale care face trimitere la o **tehnologie** (adică un ansamblu de procese, metode, operații) ce creează și implementează modalități de a executa diferite sarcini referitoare la limbajul natural (cum ar fi construcția unor interfețe – bazate pe limbaj natural – cu baze de date, traducerea automată s.a.) Procesarea limbajului natural reprezintă și astăzi o problemă dificilă și în mare parte nerezolvată. Găsirea unei tehnologii adecvate este extrem de grea datorită naturii multidisciplinare a problemei. Principalele aplicații ale procesării limbajului natural se înscriu în categoria celor bazate pe text și a celor bazate pe dialog. Din prima categorie amintim: clasificarea documentelor (și respectiv găsirea documentelor legate de un anumit subiect), regăsirea informației (cautarea unor cuvinte-cheie și/sau a unor concepte), extragerea informației (legate de un anumit subiect, deci de un anumit cuvânt-cheie), **înțelegerea textelor** (care face obiectul acestui curs și care presupune o analiză profundă a structurii acestora), traducerea automată dintr-o limbă în alta, alcătuirea de sinteze, achiziția de cunoștințe, analiza sentimentelor s.a. Aplicațiile bazate pe dialog, care implică comunicarea între om și mașină, sunt aplicații cum ar fi: sistemele de învățare, sistemele de interogare și răspuns la întrebări, rezolvarea problemelor, controlul (bazat pe limba vorbită) al unui calculator s.a. Numeroase companii care activează în numeroase industrii din întreaga lume adoptă soluții NLP pentru depășirea decalajului de comunicare om-mașină.

Cursul de față este unul introductiv, care se adresează începătorilor în domeniu. Cursul porneste de la noțiunile și conceptele de bază ale domeniului, cu care programatorul trebuie să fie familiarizat și construiește gradual o teorie care să conducă spre una dintre cele mai moderne aplicații ale domeniului, aceea a înțelegerii textelor (dezambiguizarea globală a sensului unui text).

Cursul își propune să realizeze acest lucru utilizând cele mai moderne instrumente puse la dispoziția programatorului și suportate de limbajul Python: biblioteca **NLTK** și **spaCy**. NLTK (Natural Language Toolkit) este astăzi principala platformă pentru construcția programelor în Python care lucrează cu date aparținând limbajului natural și este considerată „the mother of all NLP libraries”. spaCy adresează prelucrarea limbajului natural la nivel industrial și, din această cauză, este folosit de numeroase companii. spaCy suportă diverse limbi (germană, spaniolă, portugheză, franceză, italiană, olandeză s.a.) și înregistrează deja (în 2017) o acuratețe de 92,6%.

La finele acestui curs studentul va fi familiarizat cu principalele concepte de baza ale domeniului NLP, va fi vazut una dintre cele mai importante aplicatii ale domeniului si va fi pregatit pentru a aborda alte tipuri de aplicatii in viitor. De asemenea, va fi in posesia principalelor si totodata celor mai moderne instrumente puse la dispozitia programatorilor in domeniu.

### **PROGRAMĂ:**

1. Preprocesarea datelor apartinand limbajului natural: operatii si instrumente de baza
2. Analiza limbajului natural la nivel morfologic; programe de tip POS-tagger (Stanford POS-tagger, POS-tagger propriu NLTK)
3. Analiza limbajului natural la nivel sintactic (analiza sintactica bazata pe constituinti, analiza sintactica de dependenta)
4. Analiza limbajului natural la nivel semantic (dezambiguizare locala a sensului unui cuvânt polisemantic; dezambiguizare globala a sensului unui text)

### **CERINTE MINIMALE:**

Cunoasterea limbajului de programare PYTHON – nivel mediu

### **BIBLIOGRAFIE:**

1. Steven Bird, Ewan Klein, Edward Loper, “Natural Language Processing with Python”. Editura O’Reilly, 2009: [nltk.org/book](http://nltk.org/book)
2. Florentina Hristea, Introducere in procesarea limbajului natural. Editura Universitatii din Bucuresti, 2010
3. Tutorial Python: <https://www.learnpython.org/>

### **PROFESOR:**

Cursul si laboratorul aferent vor fi ținute de către Prof. dr. Florentina Hristea

## FIȘA CURSULUI

Titlu: **Programare dispozitive Android (\*)**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

### **OBIECTIVE:**

Cursul oferă studenților oportunitatea de a învăța noțiunile fundamentale necesare dezvoltării de aplicații mobile pentru platforma Android. În timpul laboratorului studenții vor dezvolta o aplicație mobilă nativă de la zero, aplicând cunoștințele dobândite la curs, în mediul de dezvoltare Android Studio.

### **PROGRAMĂ:**

1. Dezvoltare de aplicații mobile în Android Studio: configurarea unui proiect nou, Gradle, instrumente de debuging: Android Manager, Logcat
2. AndroidManifest
3. Interfața unei aplicații: Layouts, ListViews și Adapters; fișierele xml de tip layout
4. Fire de execuție și request-uri de tip HTTP; preluarea datelor de la un API; parsarea unui răspuns de tip json
5. Permișunile unei aplicații Android
6. Intents: deschiderea unei alte aplicații dintr-o aplicație curentă; crearea claselor de tip Activity și rolul lor
7. Content Providers; crearea unei baze de date locale SQLite; crearea de Loaders
8. Principii fundamentale de design în Android; crearea unor aplicații care suportă mai multe dimensiuni de ecran; crearea de layout-uri pentru tablete; Fragmente; crearea de view-uri personalizate;
9. Servicii de fundal și programarea sarcinilor; crearea de notificări
10. Rularea unei aplicații pe un emulator sau un dispozitiv fizic

### **BIBLIOGRAFIE:**

1. Documentația Android: <http://developer.android.com>
2. Cursul de pe platforma Udacity: <https://www.udacity.com/course/developing-android-apps--ud853>
3. Android in Action, Third Edition, W. Frank Ableson, Robi Sen, Chris King, C. Enrique Ortiz, Manning, November 2011
4. Cursul The New Boston: <https://www.thenewboston.com/videos.php?cat=278>

(\*) Cursul va fi susținut de specialiști din cadrul companiei Softbinator

## FIȘA CURSULUI

Titlu: **Programare dispozitive iOS (\*)**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

### **OBIECTIVE:**

Cursul prezintă principiile de bază ale dezvoltării aplicațiilor mobile utilizând limbajul de programare Swift în sistemul de operare iOS. În cadrul cursului vor fi prezentate atât particularitățile limbajului **Swift**, cât și cele mai bune practici de dezvoltare de aplicații mobile. În cadrul laboratorului va fi prezentat mediul de dezvoltare Xcode.

### **PROGRAMĂ:**

1. Dezvoltarea de aplicații mobile în general: diferența între dezvoltarea de aplicații Desktop și aplicații mobile, limitările și capacitățile dispozitivelor mobile, generalități despre mediile de dezvoltare de aplicații mobile (Android, iOS).
2. Introducere în Swift: tehnologia iOS, conceptul de design MVC, mecanismele de bază ale limbajului Swift, framework-ul Foundation, protocoale.
3. View: obiectul UIView, subclase custom ale lui UIView, metode de desenare, suport pentru auto-rotație, tehnici de recunoaștere a gesturilor.
4. Controller: ciclul de viață al unui UINavigationController, controller-e de navigație în aplicație (UINavigationController și UITabBarController), tranziții între controller-e.
5. Controale UI standard (framework-ul UIKit): UITableView, UIImageView, UIScrollView, UIWebView, etc.
6. Folosirea capacităților telefonului: serviciul de localizare GPS, accelerometrul, busola, integrarea hărților, etc.
7. Persistența datelor: liste de proprietăți, Realm, CoreData, Firebase, etc.
8. Topici avansate: librării externe, transmiterea de notificări între obiecte, blocuri, debugging, măsurarea performanței aplicației.

### **BIBLIOGRAFIE:**

1. Documentația Apple: XCode, Swift Basics .

(\*) Cursul va fi susținut de către specialiști din cadrul companiei Softbinator

## FIȘA CURSULUI

**Titlu: Protocoale criptografice**

Domeniul de licență: informatică (matematica)

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; seminar = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare sau examen

Credite: 5 (3)

### **OBIECTIVE:**

Cursul își propune să ofere studenților o introducere în protocoalele criptografice. Protocoalele criptografice reprezintă o preocupare fascinantă din cadrul criptografiei moderne. Studentii vor avea prima dată contact cu scheme complexe de partajare a informației, în care partenerii cu interese contradictorii și care nu au relații de încredere ajung în situația de a colabora pentru gestionarea corectă sau păstrarea unui secret. În același timp secretul lucrează și pune în mișcare resorturile care îi corespund (dacă este vorba despre un password, de exemplu) fără a fi devalat celor care îl folosesc.

### **PROGRAMĂ:**

1. Criptare asimetrică (recapitulare RSA, Elgamal, schimb de cheie)
2. Funcții trap-door, funcții hash, commitment și bit-commitment, semnături digitale
3. Passwords, provocare și răspuns, semnatura Elgamal, Shamir's No-Key protocol, Joc de noroc criptat, Semnatura secretă
4. Protocoale Zero-Knowledge: isomorfism de grafuri, Fiat-Shamir, NP-probleme, Witness Hiding și indiscernibilitate, versiuni paralele de Zero-Knowledge
5. Comunicare între multiple părți care nu au încredere reciprocă: secret sharing, comparații sigure fără comunicarea valorii absolute, evaluare sigură a unui circuit
6. Anonimitate
7. GSM: autentificare și comunicare
8. Protocoalele Needham – Rees, Needham – Schroeder, TMN
9. Oblivious transfer

### **BIBLIOGRAFIE:**

1. Albrecht Beutelspacher, Jörg Schwenk, Klaus-Dieter Wolfenstetter: Moderne Verfahren der Kryptographie
2. Uwe Schöning: Kriptologie-Kompendium
3. Nigel Smart: Cryptography (An introduction)
4. J.Katz, Y.Lindell: Introduction to Modern Cryptography, Chapman & Hall/CRC Press, 2008
5. A.J.Menezes, P.C.van Oorschot, S.A.Vanstone: - Handbook of Applied Cryptography, CRC Press, 1996

(\*) Cursul va fi susținut de către Conf. Dr. Mihai Prunescu.



## FIȘA CURSULUI

**Titlu: Securizarea și automatizarea rețelelor**

Domeniul de licență: informatică (matematica)

Specializarea: informatică, matematică-informatică, tehnologia informației

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; seminar = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5 (3)

### **OBIECTIVE:**

Cursul prezintă arhitecturile și considerațiile legate de proiectarea, securizarea, operarea și depanarea rețelelor de întreprinderi - inclusiv tehnologii de rețea largă (WAN) și mecanisme de calitate a serviciului (QoS) pentru accesul la distanță sigur, împreună cu rețelele definite de software, concepte de virtualizare și automatizare care susțin digitalizarea rețelei

Securitatea cibernetică se referă la oameni, produse și procese care protejează datele electronice de o cuceră intenționează. Cursul introduce, aprofundează și fixează noțiuni fundamentale de securizare a sistemelor informatice în general și a rețelelor de tip LAN în special pentru viitorii specialiști din domeniul IT. Studenții capătă abilități pentru a configura și depana rețelele de întreprinderi și învață să identifice și să protejeze împotriva amenințărilor de securitate cibernetică. Acestea sunt introduse în instrumentele de gestionare a rețelei și învață conceptele cheie ale rețelelor definite de software, inclusiv arhitecturile bazate pe controlere și modul în care interfețele de programare a aplicațiilor (API) permit automatizarea rețelei, iar cursul este destinat pentru a sublinia necesitatea globală pentru securitatea cibernetică și progresele în această industrie, precum și o primă etapă de înțelegere a conceptelor fundamentale din securitatea IT pentru a putea participa la Masterul "SECURITATE & LOGICĂ APLICATĂ".

### **PROGRAMĂ:**

Capitolul 1: Conceptele OSPFv2 cu o singură arie

Capitolul 2: Configurare OSPFv2 cu o singură arie

Capitolul 3: Conceptele WAN

Capitolul 4: Conceptele de securitate a rețelei

Capitolul 5: Conceptele ACL

Capitolul 6: ACL-uri și configurarea lor pentru IPv4

Capitolul 7: NAT pentru IPv4

Capitolul 8: Administrare rețea

Capitolul 9: Proiectare și depanarea rețelei

Capitolul 10: Virtualizarea și automatizarea rețelei

### **BIBLIOGRAFIE:**

[1] B. Sosinsky, CLOUD COMPUTING BIBLE, Wiley Publishing Inc., Indianapolis, Indiana, 2011.

[2] R. Jennings, CLOUD COMPUTING with Windows Azure Platform, Wiley Publishing Inc., Indianapolis, Indiana, 2009

[3] R. Buyya, J. Broberg, A. Goscinski CLOUD COMPUTING. Principles and Paradigms, Wiley, Inc., Hoboken, New Jersey, 2011

[4] T.Velte, J. Velte, R. Elsenpeter CLOUD COMPUTING: A Practical Approach, MacGrow Hill, 2010

(\*) Cursul va fi susținut de către Asist.dr. Mihaita Dragan.

## FIȘA CURSULUI

### **Titlu: Tehnici de compilare**

Domeniul de licență: informatică

Specializarea: informatică

Statutul: opțional

Nr. ore/săptămână: 4 (curs = 2; laborator = 1; seminar = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

### **OBIECTIVE:**

Cursul oferă studenților o introducere în construirea compilatoarelor. Fiecare soft care rulează pe un calculator a fost scris într-un anumit limbaj de programare. Compilatoarele translatează un program scris într-un limbaj de programare într-o formă care poate fi executată de calculator. În partea introductivă este prezentată structura unui compilator. Se prezintă în detaliu analiza lexicală și modalități de implementare a unui analizor lexical. Pentru partea de analiză sintactică (parsing) sunt introduse metodele generale top down și bottom up, care se particularizează apoi pentru gramaticile de tip LL și de tip LR care constituie baza majorității compilatoarelor. Sunt prezentați în detaliu algoritmi de analiză sintactică pentru gramaticile LL(k) tari, LR(1), SLR(1), LALR(1) care vor fi implementați la laborator. Se face o introducere și pentru partea de semantică și de generare a codului.

**EVALUARE:** 50% laborator, 50% examen

### **PROGRAMĂ:**

1. Motivație, scurt istoric. Structura unui compilator. Exemple.
2. Analiza lexicală. Descrieri lexicale cu ajutorul expresiilor regulate. Implementarea analizorului lexical.
3. Metode generale de analiză sintactică. Analiza sintactică top down, algoritmul top-down general.
4. Gramatici și limbaje LL(k). Gramatici și limbaje LL(k) tari. Mulțimile FIRST, FOLLOW. Recursivitatea la stânga. Factorizarea stângă.
5. Proprietăți ale gramaticilor LL(k). Echivalența dintre gramaticile LL(1) tari și LL(1). Parserul recursiv descendent – algoritm.
6. Parser predictiv pentru gramatici LL(k) tari – algoritm. Demonstrarea validității algoritmului pentru gramatici LL(k) tari.
7. Algoritmul Earley. Analiza sintactică bottom up - metoda generală. Gramatici și limbaje LR(k), definiții, proprietăți.
8. Parser de tip deplasare-reducere pentru gramatici LR(1) – algoritm. Demonstrarea validității algoritmului pentru gramatici LR(1).
9. Parser SLR(1) – algoritm. Parser LALR(1) – algoritm. Revenirea din eroare în parsere de tip LR.
10. Analiza semantică. Gramatici cu atribute, atribute sintetizate și atribute moștenite. Exemple.

### **BIBLIOGRAFIE:**

- A. Aho, M. Lam, R. Sethi, J. Ullman, *Compilers: Principles, Techniques & Tools*, 2007, Addison Wesley
- A. Aaby, *Compiler Construction using Flex and Bison*, 2004,
- Bruno Preiss, *Lexical Analysis and Parsing using C++*, 2004

Cursul va fi susținut de lect. Dr. Gianina Georgescu

## FIȘA CURSULUI

Titlu: **Tehnici de programare a aplicațiilor grafice**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

### **OBIECTIVE:**

Grafica pe calculator este omniprezentă, având aplicații în domenii variate, cum ar fi medicina, ingineria asistată de calculator, dezvoltarea jocurilor, realizarea filmelor, etc. Pentru a obține efecte cât mai realiste, este utilizată o gamă extrem de variată de tehnici și metode. Obiectivul cursului este de a prezenta câteva tehnici fundamentale de programare grafică, atât la nivelul graficii bidimensionale (având în vedere ambele formate, de grafică vectorială, respectiv de tip raster) cât și la nivelul modelării geometrice 3D. Expunerea fundamentelor teoretice ale acestor metode de lucru este însoțită de prezentarea și utilizarea unor aplicații software dedicate. Cursul va fi completat de un laborator, unde vor fi realizate proiecte care să ilustreze conceptele și rezultatele expuse și care vor reprezenta componenta principală a verificării semestriale.

### **PROGRAMĂ:**

1. Principii fundamentale ale graficii vectoriale. Generarea curbelor și suprafețelor Bézier. Utilizarea unui format specific pentru grafica vectorială (*SVG – Scalable Vector Graphics*).
2. Procesarea imaginilor. Utilizarea unui editor de grafică de tip raster (*GIMP – GNU Image Manipulation Program*).
3. Convoluție și aplicații: efecte vizuale, detectarea conturilor, compresia imaginilor.
4. Modelare 3D în grafică – principii generale.
5. Rețele poligonale (*polygon meshes*) și modelarea obiectelor în context 3D. Formate specifice. Utilizarea unui mediu de modelare 3D (*Blender*) pentru scene și animații tridimensionale.
6. Metodele *Ray Casting* și *Ray Tracing*.
7. Simularea unor modele fizice. Sisteme de particule în grafica pe calculator.

### **BIBLIOGRAFIE:**

1. W. Boehm, H. Prautzsch, *Geometric Concepts for Geometric Design*, AK Peters, Wellesley, 1994.
2. G. Farin, *Curves and Surfaces for CAGD, A practical Guide*, Academic Press, 2002.
3. J. Hughes, A. van Dam, M. McGuire, D. Sklar, J. Foley, S. Feiner, K. Akeley, *Computer Graphics: Principles and Practice* (3rd edition), Addison Wesley, 2013.
4. P. Schneider, D. Eberly, *Geometric Tools for Computer Graphics*, Morgan Kaufmann, 2003.
5. P. Shirley, M. Ashikhmin, M. Gleicher, S. Marschner, E. Reinhard, K. Sung, W. Thompson, P. Willemsen, *Fundamentals of Computer Graphics* (3rd edition), AK Peters, Wellesley, 2009.
6. A. Watt, M. Watt, *Advanced Animation and Rendering Techniques: Theory and Practice*, Addison-Wesley, 1992.

Cursul va fi susținut de către Conf.univ.dr. Sorin Stupariu

## FIȘA CURSULUI

**Titlu: Testarea sistemelor software**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

Nr. ore/săptămână: 3 (curs = 2; laborator = 1)

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

### **OBIECTIVE:**

Cursul prezintă principalele tehnici și metode de testare a sistemelor software. Sunt prezentate atât metodele ingineresti cu largă utilizare practică, cât și tehnici mai avansate, care fac obiectul cercetărilor recente. Evaluarea va fi făcută în urma unui scurt proiect individual (tematica fiind aceeași pentru toți studenții) și a unei prezentări în grupuri de 3-5 studenți (tema la alegere dintr-o listă dată). Proiectul individual va verifica noțiuni de bază, practice, despre testarea software, iar prezentarea va pune în evidență capacitatea studenților de a înțelege și discuta o temă mai avansată.

### **PROGRAMĂ:**

1. **Testarea sistemelor software:** problematica; testare vs verificare formală; tehnici de generare de date de test; unit testing; JUnit
2. **Metode de testare funcțională (black-box):** partitionarea în clase de echivalență, analiza valorilor de frontieră; metoda partitionării în categorii, testarea folosind analiza cauza-efect.
3. **Metode de testare structurală (white-box):** acoperiri la nivel de instrucțiuni, ramură, condiție/decizie, condiții multiple, etc.; complexitatea ciclomatică (McCabe), generarea de circuite liniar independente; strategii de generare de date de test la nivel de cale; utilizare de code coverage.
4. **Testarea bazată pe mutație (mutation testing):** weak mutation, strong mutation; operatori de mutație; utilizarea mutantilor pentru evaluarea seturilor de test; utilitățile PIT și MuJava.
5. **Generarea datelor de test folosind metode de căutare metaeuristice:** metode de căutare locale și metode globale; folosirea metodelor metaeuristice în testarea structurală și testarea funcțională.
6. **Metode de testare bazate pe formalisme cu stări:** metodele W, Wp(W partial), UIO (unique input/output), DS (distinguishing sequence) pentru mașini cu stări finite; testare bazată pe formalisme extinse cu stări (extended finite state machines).
7. **Testare și analiză pe baza modelului:** proprietăți de safety și liveness, invariante, stări acceptoare, stări moarte; explorarea scenariilor folosind compunere de automate; tehnici de reducere a modelului (pruning techniques); exemplificare folosind utilitarul NModel.

### **BIBLIOGRAFIE:**

1. PIT home page: <http://pitest.org/>
2. MuJava home page: <http://cs.gmu.edu/~offutt/mujava>
3. MuClipse home page <http://muclipse.sourceforge.net/>
4. NModel home page <http://nmodel.codeplex.com/>
5. M. Holcombe, F. Ipate: *Correct Systems: building business process solutions*, Springer Verlag, 1998.
6. F. Ipate: Testare funcțională; Testare structurală; Mutation testing; Search based testing – note de curs.

7. J. Jacky, M. Veanes, C. Campbell, W. Schulte. Model-based Software Testing and Analysis with C#. Cambridge University Press, 2008.
8. R. Lefticaru, F. Ipate: Automatic State-Based Test Generation Using Genetic Algorithms. SYNASC 2007, 188-195, 2007.
9. A. Mathur: *Foundations of Software Testing*, Addison-Wesley Professional, 2007.
10. M. Roper: *Software Testing*, McGraw-Hill, 1994

Cursul va fi susținut de către Lect.univ.dr. Sorina Predut