

**CURSURI OPȚIONALE DE
INFORMATICĂ**

PROPUSE

**PENTRU ANUL
UNIVERSITAR 2020-2021**

**DOMENIUL DE LICENȚĂ:
INFORMATICĂ**

**SPECIALIZAREA:
INFORMATICĂ**

CERERE PENTRU CURSURI OPȚIONALE

INFORMATICĂ

LICENȚĂ ANUL III (2020 - 2021)

NUME STUDENT:

GRUPA CURENTĂ:

<i>Nr.</i>	<i>Denumire curs</i>	
1	Algoritmi combinatoriali pe secvențe finite și infinite	<i>Semestrul I</i>
2	Aplicații web în Python și Django (*)	
3	Cloud computing	
4	Concepte geometrice în grafica pe calculator	
5	Concepte și aplicații în vederea artificială	
6	Front-end avansat (React și Angular) (*)	
7	Introducere în Reinforcement Learning	
8	Introducere în Robotic Process Automation (RPA)	
9	JavaScript server-side: Node.js + GraphQL (*)	
10	Procesarea semnalelor	
11	Programare web cu PHP și MySQL	
12	Proiectarea jocurilor pe calculator	
13	Rutare și comutare în rețele de calculatoare	
14	Blockchain – concepte, tehnologii și aplicații	<i>Semestrul II</i>
15	Inițiere în bioinformatică și cercetare	
16	Introducere în prelucrarea limbajului natural	
17	Învățare automată în vedere artificială (*)	
18	Învățarea rețelelor neurale adânci (*)	
19	Prelucrarea limbajului natural în python cu NLTK și SPACY	
20	Programare dispozitive Android (*)	
21	Programare dispozitive iOS (*)	
22	Protocoale criptografice	
23	Securizarea și automatizarea rețelelor	
24	Tehnici de compilare	
25	Tehnici de programare a aplicațiilor grafice	
26	Testarea sistemelor software	

NOTĂ: Cursurile opționale vor fi numerotate de către studenți în ordinea preferințelor, de la 1 la 26, indiferent de semestru. Studenții vor fi alocați la cursuri în ordinea descrescătoare a punctajului de clasificare după primii 2 ani de studii și în limita numărului de locuri la fiecare curs, astfel încât să urmeze câte un curs pe fiecare semestru, dar nu mai mult de un curs cu (*) în total.

DATA:

SEMNĂTURA:

FIȘA CURSULUI

Titlu: **Algoritmi combinatoriali pe secvențe finite și infinite (*)**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Procesarea algoritmică a secvențelor finite și infinite de simboluri este foarte utilă atât din punct de vedere teoretic dar și practic. Secvențierea ADN-ului este doar un exemplu care motivează studiul algoritmic al secvențelor, dar accentul acestui curs este pus pe algoritmi și concepte care nu sunt specifice bioinformaticii. Cursul discută diverse abordări algoritmice ale unor probleme referitoare la secvențe liniare finite sau infinite de simboluri ce apar în diferite domenii.

În acest curs se vor prezenta atât algoritmi clasici, precum și algoritmi recent prezentați în literatură, algoritmi exacti sau aproximativi, algoritmi avansați pentru analiza bazelor de date cu secvențe mari.

PROGRAMĂ:

- Probleme algoritmice pe L-sisteme.
- Repetitive în cuvinte. Cuvinte primitive. Separarea cuvintelor cu automate. Periodicitate. Relații de ordine pe cuvinte. Complexitatea cuvintelor prin subcuvinte.
- Secvențe infinite: Siruri Morse, Thue, Kolakowski, "self-reading", "self-adding", Langford, Sturmieni.
- Căutare exactă și aproximativă de cuvinte în dicționare. Noțiunea de pattern. Căutare cu ajutorul automatelor. Căutări multiple în dicționare fixe și dinamice. Compresie de date.
- Algoritmi combinatoriali pe secvențe ADN. Multiple sequence alignment. Determinarea unor distanțe între genomuri (numărul minim de mutații cromozomiale ce transformă un genom în altul). Distanța prin inversiuni. Distanța prin translocatii.

BIBLIOGRAFIE:

1. D. GUSFIELD, Algorithm on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge Press, 1997.
2. V. MITRANA, Bioinformatica, Ed. L& S Infomat, București, 1997.
3. E. OHLEBUSCH, Bioinformatics Algorithms: Sequence Analysis, Genome Rearrangements, and Phylogenetic Reconstruction. Oldenbusch Verlag, 2013.
4. G. NAVARRO, M. RAFFINOT, Flexible Pattern Matching in Strings. Practical On-Line Search Algorithms for Texts and Biological Sequences, Cambridge University Press, 2014.
5. V. MAKINEN, D. BELAZZOUGUI, F. CUNIAL, A. TOMESCU, Genome-Scale Algorithm Design, Cambridge University Press, 2015.

(*) Cursul va fi susținut de către Prof. Dr. Victor Mitrana.

FIȘA CURSULUI

Titlu: **Aplicații web în Python si Django (*)**

Domeniul de licență: informatică

Specializarea: informatică

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Cursul își propune să ofere studenților o introducere în programarea web folosind limbajul de programare Python și platforma Django. În prima parte a cursului, studenții sunt familiarizați cu limbajul, cu sintaxa și construcțiile specifice programării de nivel înalt. În cea de-a doua parte, se pune accentul pe dezvoltarea unei aplicații web, pornind de la partea de proiectare și parcurgând cu atenție toate etapele de dezvoltare, până la finalizarea aplicației. Cursul este interactiv, în timpul prezentării studenții fiind încurajați să participe cu sugestii și întrebări. Partea practică de tipul hands-on se face în paralel cu prezentarea, anumite părți de cod fiind puse la dispoziție pe proiector sau online pe pagina de resurse a cursului. La finalul cursului, absolvenții vor avea atât noțiunile necesare înțelegerii programării web, cât și abilitatea de a dezvolta o aplicație web de la zero.

PROGRAMĂ:

1. Curs introductiv: Modalitate de evaluare; Mediu de dezvoltare
2. Introducere și sintaxă: Lucrul cu interpretorul; Instrucțiuni condiționale, bucle; Variabile, tipuri de date; Funcții, argumente, apelare; Operații de bază și noțiuni specifice Python; Funcții built-in, I/O, Fișiere; Librării standard: re, csv, json etc.; List/dictionary/set comprehension; Map/Lambda/Filter
3. OOP: Clase și obiecte; Instanțiere; Metode și atribute; Moștenire; Web basics; Client – server; HTTP protocol (text and methods); URL; How to send data: query param/body; Python requests - GET/POST
4. MVC, introducere în Django: Modele; View-uri; Template-uri; Formulare; Interfața de administrare
5. Lucrul cu baza de date: Modele; Interogarea bazei de date; Comenzi de management
6. View-uri și formulare: View-uri, formulare; Template-uri; Formulare bazate pe modele
7. Testare

BIBLIOGRAFIE:

1. Documentația online PurePython: <http://purepython.eaudeweb.ro/wiki/Home.html>
2. Documentația online Python: <https://docs.python.org/3/>
3. Documentația online Django : <https://docs.djangoproject.com/en/2.1/>

(*) Cursul va fi susținut de către M.Sc. Diana Boianțiu de la Eau de Web

FIȘA CURSULUI

Titlu: **Cloud Computing**

Domeniul de licență: **Informatică**

Specializarea: informatică (matematica-informatica),

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare Credite: 5

OBIECTIVE:

- Familiarizarea studenților cu una dintre paradigmele centrale ale ”transformării digitale” care se constituie în una dintre tehnologiile convergente ale viitorului (**cloud computing**, IoT și ML) cu impact major în dezvoltarea economică, socială și tehnologică la nivel global.
- Dezvoltarea de aptitudini în domeniul proiectării și implementării de aplicații bazate pe cloud computing, care vizează orchestrarea sistemelor de servicii disponibilizate în varianta ITaaS incluzând aici IaaS, PaaS, SaaS, DBaaS, DWaaS, Hadoop-as-a-Service, MLaaS, care integrează cunoștințele tehnologice acumulate în primii ani de studiu și care crează noi abilități tehnologice solicitate de o piață a muncii extrem de dinamică.
- Dezvoltarea de abilități de proiectare a unor aplicații bazate pe arhitecturi orientate pe servicii(SOA) în care să se integreze resurse de calcul disponibilizate în cloud cu cele expuse on-premises, de o manieră scalabilă în funcție de cerințele de business ale utilizatorului, în condițiile alinierii la metodele de dezvoltare software bazate pe **DevOps**.

PROGRAMĂ:

1. Sisteme distribuite tolerante-la-erori (engl.fault-tolerant systems)
2. Masini virtuale vs. Containere. Orchestrare- Kubernetes, DC/OS în cloud
3. Arhitecturi bazate pe microservicii și multi-containere în cloud
4. Servicii Distribuite în Cloud Computing
5. Servicii de stocare și caching. Interfețe programabile.
6. Servicii platformă pentru **baze de date, data warehouse** și data lake utilizate în context cloud computing.
7. Servicii Big Data în context cloud computing(HDInsight-BaaS)
8. Big Data bazat pe memory-computing(Apache Spark)
9. Servicii pentru Machine Learning în Cloud Computing
10. Servicii pentru analitice în Big Data(Azure Data Bricks)
11. Analitice în timp-real bazate pe Data-Lake-as-a-Service în cloud
12. Arhitecturi Lambda (AWS Lambda, Azure Function, Google Function)
13. Cloud computing și dezvoltarea de aplicații cognitive
14. Medii de test și deployment în DEVOPS
15. Platforme Blockchain în cloud

BIBLIOGRAFIE:

1. Barnes, Jeff. 2015. Azure Machine Learning. Microsoft Press, 2015.
2. Buyya, Rajkumar, Broberg, James and Goscinski, Andrej. 2011. Cloud Computing. Principles and Paradigms., Wiley, 2011.
3. Chandrasekaran, K. 2015. Essentials of CLOUD COMPUTING. s.l. : CRC Press, 2015.

4. Gendron, Michael S. 2014. Business Intelligence and the Cloud. s.l. : Wiley, 2014.
5. J. Hurwitz, M. Kaufman, A. Bowles. Cognitive Computing and Big Data Analytics.. Wiley, 2015.
6. Marc, Farley. 2013. Rethinking enterprise storage-A hybrid cloud model. s.l. : Microsoft Press, 2013.
7. N. Tanasseri, R., Rai. 2017. Microservices with Azure. Birmingham : Packt Publishing Ltd., 2017.
8. Torre, Cesar de la. 2017. Containerized Docker Application Lifecycle with Microsoft Platform and Tools. Microsoft Press, 2017.
9. Microsoft Press, 2017.
10. Varga, Stacia, Cherry, Denny and D'Antoni, Joseph. 2016. Introducing Microsoft SQL Server 2016 Mission-Critical Applications, Deeper Insights, Hyperscale Cloud. s.l. : Microsoft Press, 2016.

Cursul va fi susținut de către conf.univ.dr. Cristian Kevorchian.

FIȘA CURSULUI

Titlu: **Concepte geometrice in grafica pe calculator**

Domeniul de licență: informatică

Specializarea: informatică

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Cursul are două componente complementare. Prima dintre ele are o natură teoretică: în cadrul cursului vor fi prezentate noțiuni și rezultate geometrice esențiale pentru dezvoltarea aplicațiilor grafice, precum și câțiva algoritmi de bază în grafica pe calculator. Cea de-a doua componentă are o natură aplicativă și include o introducere în utilizarea versiunii moderne a OpenGL – bazată pe unități programabile (shaders) și dezvoltarea unor aplicații interactive de grafică pe calculator. Proiectele de laborator, concretizate prin proiecte, vor reprezenta componenta principală a verificării de la sfârșitul semestrului.

PROGRAMĂ:

1. Generalități despre OpenGL. Principii de lucru, biblioteci utilizate.
2. Primitive grafice. Atribute ale primitivelor grafice.
3. Transformări geometrice. Modelarea deplasării unui obiect.
4. Utilizarea texturilor.
5. Reprezentarea imaginilor 2D.
6. Transformări de vizualizare pentru scene 3D.
7. Proiecții paralele și proiecții centrale.
8. Modelarea și reprezentarea obiectelor în grafica 3D.
9. Modele de iluminare. Efecte vizuale (netezirea conturilor, ceață, umbre).
10. Aplicații grafice interactive. Animații pe calculator.

BIBLIOGRAFIE:

1. G. Albeanu, *Grafica pe calculator. Algoritmi fundamentali*, Editura Universității din București, 2001.
2. R. Baci, *Programarea aplicațiilor grafice 3D cu OpenGL*, Editura Albastră, 2005.
3. W. Boehm, H. Prautzsch, *Geometric Concepts for Geometric Design*, AK Peters, Wellesley, 1994.
4. 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.
5. D. Hearn, M. Baker, W. Carithers, *Computer Graphics with OpenGL*, 4th ed, Prentice Hall, 2010.
6. 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.
7. D. Shreiner, G. Sellers, J. Kessenich, B. Licea-Kane, *OpenGL Programming Guide, Eighth Edition*, Addison-Wesley, 2013.

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

FIȘA CURSULUI

Titlu: **Concepte și aplicații în vederea artificială**

Domeniul de licență: informatică

Specializarea: informatică

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Oamenii interpretează fără efort ceea ce văd. Domeniul Vederii Artificiale își propune să construiască o mașină care, asemenea oamenilor, "vede" și "înțelege" ceea ce se întâmplă în imagini. Pentru realizarea acestui deziderat este nevoie de algoritmi eficienți care să extragă informații din pixeli pentru interpretarea imaginilor.

Cursul își propune să prezinte noțiunile fundamentale în domeniul Vederii Artificiale însoțite de aplicații implementate la laborator în Python și OpenCV. Cele mai interesante aplicații vor fi transformate în teme. De-a lungul semestrului vor fi cinci teme din care studenții își pot obține nota finală. Pentru cei care nu rezolvă teme, evaluarea va consta într-o lucrare de laborator în ultima săptămână.

La sfârșitul cursului, studenții vor căpăta competențe în proiectarea sistemelor inteligente ce procesează imagini pentru rezolvarea unor probleme practice precum: realizarea de mozaicuri tematice, redimensionarea imaginilor cu păstrarea conținutului, sinteza și transferul texturii, detectare facială, colorizarea unei imagini, lucrul cu rețele neuronale convoluționale, etc.

PROGRAMĂ:

[1] Introducere în vederea artificială. Concepte de bază.

[2] Filtre pentru imagini. Muchii și gradienti. Textură și culoare.

[3] Reprezentarea conținutului vizual al unei imagini folosind descriptori. Descriptori bazați pe histograme de gradienti orientați. Detectarea facială folosind metoda ferestrei glisante și histograme de gradienti orientați.

[4] Rețele neuronale convoluționale. Aplicații la recunoașterea claselor de obiecte.

BIBLIOGRAFIE:

1. Richard Szeliski: *Computer Vision: Algorithms and Applications*, Springer, 2010. Draft available online at <http://szeliski.org/Book/>

2. David A. Forsyth, Jean Ponce: *Computer Vision: A Modern Approach*, Pearson, 2nd Edition, 2011, Draft available online.

3. Christopher M. Bishop: *Pattern Recognition and Machine Learning*, Springer, 2006.

4. Kristen Grauman, Bastian Leibe: *Visual Object Recognition*, Morgan&Claypool publishers, 2011.

5. Ian Goodfellow, Yoshua Bengio, Aaron Courville: *Deep Learning*, MIT Press, 2016.

6. Tutorial Python: <https://www.learnpython.org/>

Cursul va fi ținut de către Prof. Radu Ionescu și Conf. Bogdan Alexe, laboratorul va ținut de către drd. Iuliana Georgescu.

FIȘA CURSULUI

Titlu: **Front-end avansat (React si Angular) (*)**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Cursul prezintă principiile de bază ale dezvoltării de aplicații WEB folosind HTML5, CSS3 și două dintre cele mai folosite framework-uri pentru Front-end – AngularJS și ReactJS. În cadrul cursului vor fi prezentate atât noțiuni avansate de Javascript, cât și cele mai bune practici de implementare.

PROGRAMĂ:

1. Introducere în HTML5, CSS3 și Bootstrap/Semantic. Responsive design/Mobile first. Aplicații de tip Single Page (SPA). Javascript și JQuery.
2. Javascript avansat - Scope, Hoisting, Closures, Prototipuri, Callbacks, Promises și Obiecte (JSON).
3. Introducere în AngularJS – prezentare framework (componente, module și servicii). comunicarea cu servicii. ES6 și Typescript.
4. AngularJS avansat - Template-uri, formulare și routing. Realizare de aplicații simple care să demonstreze noțiunile învățate. Arhitectura proiectelor bazate pe AngularJS în practică.
5. Introducere în ReactJS – prezentare framework – JSX, React-dom.js, Nodes. Componente – props și state.
6. ReactJS avansat – Flux, Reflux și Redux. Realizare de aplicații simple care să demonstreze noțiunile învățate. Arhitectura proiectelor bazate pe ReactJS în practică.

BIBLIOGRAFIE:

1. Documentația oficială React (<https://reactjs.org/>) și Angular (<https://angularjs.org/>).

Cursul va fi susținut de către specialiști de la compania Softbinator

FIȘA CURSULUI

Titlu: **Introducere in Reinforcement Learning**

Domeniul de licență: informatică

Specializarea: informatică

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Acest curs acoperă principiile și tehnicile fundamentale ale metodelor de Reinforcement Learning. Subiectele tratate includ metode precum: Procese de decizie Markov, metode Monte Carlo, Policy Gradients, Inverse Reinforcement Learning. Sunt tratate de asemenea și toate notiunile teoretice necesare înțelegerii acestui curs: rețele neuronale convoluționale și recurente, algoritmi de backpropagation, tehnici de regularizare și optimizare pentru antrenarea unor astfel de rețele.

Atât cursul cât și laboratorul vor fi orientate către partea aplicativă. În cadrul laboratorului vom urmări și implementa metode de Reinforcement Learning din domenii ca: self driving cars, jocuri video, sisteme de recomandări, etc. Limbajele acceptate pentru implementări și proiecte sunt Python și C++. O introducere aplicată în Python este de asemenea realizată în cadrul laboratorului. Notarea se va face pe baza unui proiect individual ales de fiecare student. Proiectele realizate în cadrul cursului pot constitui de altfel și baza proiectului de licență.

PROGRAMĂ:

1. Procese de decizie Markov
2. Metode Monte Carlo. Q-Learning.
3. Introducere în structuri neuronale avansate. Rețele convoluționale și recurente.
4. Algoritmi de backpropagation, tehnici de regularizare și optimizare.
5. Deep Q-Learning și variante.
6. Introducere și motivație în Policy Gradients. Diferența temporală. Metoda Actor-Critic.
7. Natural Policy Gradients. Metodele TRPO, PPO.
8. Model-based Reinforcement Learning. Imitation Learning.
9. Inverse Reinforcement Learning.
10. Generative Adversarial Reinforcement Learning.

BIBLIOGRAFIE:

1. Richard S. Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, Second Edition. The MIT Press, 2018.
2. Yoshua Bengio, Ian Goodfellow, Aaron Courville, Deep Learning. The MIT Press, 2016
3. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd ed.. Pearson, 2009

(*) Cursul va fi susținut de către Lect. Dr. Paduraru Ciprian.

FIȘA CURSULUI

Titlu: **Introducere in Robotic Process Automation (RPA)**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Robotic Process Automation este tehnologia care permite oricui astăzi să configureze un "robot" pentru a emula și a integra acțiunile unei persoane care interacționează cu sistemele digitale în executarea unui proces. Robotii RPA utilizează interfața cu utilizatorul pentru a capta date și a manipula aplicații la fel ca oamenii. Aceștia interpretează, declanșează răspunsuri și comunică cu alte sisteme pentru a putea realiza o gamă largă de sarcini repetitive, în mod substanțial mai bine: un robot software RPA nu doarme, nu face greșeli și costă mult mai puțin decât un angajat.

În cadrul acestui curs, veți învăța să construiți roboți RPA ce sunt capabili să se conecteze la aplicații, să opereze fișiere și foldere, să copieze și să insereze date, să completeze formulare, să extragă date structurate și semi-structurate din documente, să facă scrape în browsere, să citească și să insereze date într-o bază de date, să trimită și să deschidă email-uri și atasamente, să facă calcule și multe altele. De asemenea, soluțiile RPA create pot fi îmbunătățite prin machine learning și inteligența artificială (AI), care pot fi inserate în fluxurile de lucru RPA pentru a efectua sarcini de percepție a mașinii, cum ar fi image recognition.

UiPath este o firmă de software românească care a devenit în ultimii ani **lider mondial în domeniul RPA, fiind primul „unicorn” din România** (este momentan evaluată la peste 7 miliarde de dolari). UiPath a dezvoltat un framework RPA foarte apreciat în industrie.

Scopul acestui curs este de a oferi studenților o introducere în această nouă tehnologie și într-unul din frameworkurile cele mai bune și folosite din industrie – UiPath. **Deoarece piața RPA este în creștere foarte rapidă, specialiștii RPA vor fi foarte căutați pe piața dezvoltatorilor software** (deja există multe joburi de RPA developer), fiind un domeniu de nișă care necesită atât cunoștințe de programare, dar și elemente de AI și un nou mod de a implementa soluții software într-o companie.

Welcome to the future!

PROGRAMĂ:

1. Introducere în RPA:
 - Ce este RPA; Unele RPA și standarde de industrie; Tehnologii conexe (Microsoft .NET Workflow Foundation)
2. Implementare RPA în UiPath:
 - Componente și interacțiunea dintre ele; UiPath Studio; UiPath Activities; UiPath Robot; UiPath Orchestrator (detaliere: Tennaants; Authentication; Users; Roles; Robots; Environments; Processes; Queues and transactions; Jobs; Schedules; Logging; Alerts)
3. Elemente avansate RPA:

- Types and Control Flows; Data Manipulation; Recording (web, desktop); Advanced UI interaction (input/output methods, screen scraping, data scraping); Selectors; Image and Text Automation; Excel & Data Tables automation; PDF automation; E-mail automation; Custom activities using C# Debugging & Exception Handling Project Organization; Testing; Project organization; AI for RPA.

BIBLIOGRAFIE:

- <https://www.uipath.com/rpa/academy>
- <https://go.uipath.com/>
- <https://www.uipath.com/automate/robotic-process-automation>
- Alok Mani Tripathi. Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool – UiPath. Packt Publ.; 1st ed. 2018
- Steve Kaelble, RPA for Dummies, 2018
https://www.nice.com/websites/rpa/assets/robotic_process_automation_for_dummies.pdf

Cursul va fi susținut de către un cadru didactic UB împreună cu experți de la compania UiPath și alți experți din industrie

FIȘA CURSULUI

Titlu: **JavaScript server-side: Node.js + GraphQL (*)**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Cursul își propune să ofere studenților oportunitatea de a lucra cu un stack Javascript modern pe server-side. Prima parte a cursului se va concentra pe modul de lucru cu Javascript, sintaxă, toolchain, build process. Cea de-a doua parte integrează toate layere-ele de la baza de date la serverul web folosind GraphQL și tool-urile aferente.

Obiectivul final este un proiect, particularizat pentru fiecare în parte, pe o temă aleasă, care să demonstreze cunostiintele acumulate în domeniul Javascript-ului pe Backend.

Arnia Software este un grup care activează în zona de custom software development, dar deține și TimesNewRoman.ro, dezvoltă mai multe produse open-source și are expertiză în domenii variate (deep learning, computer vision, image processing).

PROGRAMĂ:

- Javascript I: Fundamente, mediu de dezvoltare, sintaxă
- Javascript II: Async/await, Promises, modules, classes
- NodeJS: Fundamente, prezentare platformă, npm, npm scripts
- GraphQL I: Fundamente, concepte, set-up în proiect
- DB: SQL vs. NoSQL, ORM, concepte, setup in proiect, queries
- DB II: Relații între Modele, relații polimorfice
- GraphQL II: Middleware, autentificare, mutations
- Testing: Fundamente, toolset, exemple, setup în proiect
- Production Systems: Deployment, build process

BIBLIOGRAFIE:

1. <http://es6-features.org>
2. <https://nodejs.org/dist/latest-v8.x/docs/api/>
3. <https://www.apollographql.com/docs/apollo-server/>
4. <https://vincit.github.io/objection.js/>
5. <http://knexjs.org/>
6. <https://jestjs.io/>

Cursul va fi susținut de către Adrian Mihai și Virgil Bugnariu, specialiști din cadrul firmei Arnia Software.

FIȘA CURSULUI

Titlu: **Procesarea semnalelor**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Dacă informația este reprezentată binar pe mașina de calcul modernă, cum sunt atunci traduse sunetele, imaginile și alte semnale continue din viața reală, într-un domeniu discret atât de limitat? Cât din ce aud și din ce văd trebuie înregistrat pentru a reproduce întâmplarea întocmai în viitor? Cursul răspunde la aceste întrebări și oferă o viziune de ansamblu a domeniului de procesare a semnalelor.

Abordarea subiectelor este pragmatică, în doi pași. La primul pas sunt prezentate intuitiv aspectele teoretice, acestea fiind tratate drept simple unelte. Pasul doi constă în studii de caz ce analizează implementări concrete, de succes, din domeniu.

La final, studenții vor avea baza necesară prelucrării și reprezentării oricărui tip de semnal, împreună cu experiența aplicării acestora în domeniile audio, video, de compresie a datelor și de imagistică medicală.

FORMA DE EXAMINARE:

Dat fiind caracterul aplicativ al cursului, studenții vor fi evaluați pe baza temelor realizate acasă și în cadrul laboratorului aferente studiilor de caz. În lipsa rezolvării acestor teme, studenții vor fi evaluați printr-o lucrare de laborator, în ultima săptămână.

Studenții au șansa de a propune proiecte proprii, cele mai interesante putându-se transforma în teme pentru lucrarea de licență.

PROGRAMĂ:

1. Introducere în procesarea semnalelor. Concepte de bază
2. Domeniul timpului și al spațiului
3. Domeniul Fourier
4. Tipuri de filtre. Filtrare în timp și frecvență
5. Eșantionare. Decimare. Interpolare
6. Studiu de caz: formatul audio MP3
7. Domeniul Wavelet
8. Studiu de caz: formatul imaginilor JPEG2000
9. Cuantizarea. Compresia datelor
10. Studiu de caz: coduri Huffman și formatul ZIP
11. Reprezentări rare. Compressed sensing
12. Studiu de caz: imagistică medicală prin tehnici RMN și CT

BIBLIOGRAFIE:

1. Oppenheim, A. V., & Schaffer, R. W., *Digital Signal Processing*, Prentice-Hall, Inc., 1975.
2. Lyons, R. G., *Understanding Digital Signal Processing*, 3/E Pearson Education, 2004.
3. Mallat, S., *A Wavelet Tour of Signal Processing*. Elsevier, 1999.
4. Walker, J.S. *A Primer on Wavelets and Their Scientific Applications*. CRC press, 2002.
5. M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal Processing*, Springer, 2010.
6. Eldar, Y.C.; Kutyniok, G., *Compressed sensing: theory and applications*. Cambridge University Press, 2012.
7. Dumitrescu, B., *Prelucrarea semnalelor: breviar teoretic, probleme rezolvate, ghid Matlab*, suport de curs, 2006
8. Stănișilă, O., *Undine. Teorie și aplicații.*, Academia Română, 2010

Cursul va fi susținut de către Lect.univ.dr. Paul Irofti

FIȘA CURSULUI

Titlu: **Programare web cu PHP și MySQL**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Cursul prezintă noțiuni fundamentale, tehnici și tehnologii care stau la baza paginilor web ce utilizează baze de date. Studenții care vor urma acest curs vor dobândi experiență în utilizarea PHP și MySQL, aplicate în crearea/prelucrarea paginilor web, în folosirea unor librării care permit realizarea unor aplicații complexe etc. Dintre obiectivele cursului menționăm:

- însușirea celor mai importante tehnologii și dobândirea aptitudinilor necesare dezvoltării și gestiunii paginilor web dinamice;
- prezentarea și utilizarea limbajului PHP, precum și a tehnicilor de stocare și regăsire a datelor utilizând sistemul de gestiune a bazelor de date MySQL;
- aplicarea noțiunilor prezentate la curs pentru rezolvarea unor probleme specifice întâlnite în dezvoltarea aplicațiilor web;
- dobândirea de noi cunoștințe cu privire la tendințele actuale în programarea aplicațiilor web.

PROGRAMĂ:

- [1] Introducere în PHP: HTML vs. PHP, aplicații web dinamice, procesarea la nivel de client versus procesarea la nivel de server, avantaje și funcționalități ale PHP, configurări de bază.
- [2] Utilizarea server-ului web Apache: instalare, configurare, host-uri virtuale, proprietăți etc.
- [3] Elemente de programare PHP: operatori, funcții, tablouri etc.
- [4] Prelucrarea fișierelor de date în PHP: căutare, editare, securitate, gestiunea directoarelor etc.
- [5] Configurarea și utilizarea server-ului de baze de date MySQL: instalare și configurare, conectare, funcții specifice, apeluri SQL, utilitare pentru administrarea bazei de date.
- [6] Lucrul cu baze de date și formulare: configurarea PHP pentru interacțiunea cu baza de date, interfețe de programare specifice, lucrul cu formulare, utilizarea cookie-urilor etc.
- [7] Generarea imaginilor, documentelor PDF și Excel utilizând PHP.
- [8] Programare orientată pe obiecte în PHP.
- [9] Prezentarea celor mai cunoscute/utilizate librării din PHP."

BIBLIOGRAFIE:

- [1] Lockhart J., Modern PHP: New Features and Good Practices, O'Reilly Media, 2015
- [2] Ullman L., PHP and MySQL for Dynamic Web Sites, Peachpit Press, 2012
- [3] Welling L., Thomson L., PHP and MySQL Web Development, Addison Wesley, 2009
- [4] Nixon R., Learning PHP, MySQL & JavaScript With jQuery, CSS & HTML5, O'Reilly, 2015
- [5] DuBois P., MySQL (5th Edition), Addison Wesley, 2013

Nota: Dintr-un studiu publicat în luna octombrie 2015 reiese că cele mai căutate limbaje de programare în România sunt: Java, PHP, Net/C#, C/C++. Un procent semnificativ al lucrărilor de licență au aplicații ce folosesc PHP, iar numeroși studenți proaspat angajați, folosesc PHP la locul de muncă

Cursul va fi susținut de către Lect.dr. Silviu Laurentiu Vasile.

FIȘA CURSULUI

Titlu: Introducere în Programarea Jocurilor pe Calculator

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Obiectivul cursului este unul dublu:

În primul rând, în cazul prelegilor de curs, studentul va dobândi cunoștințe despre elemente teoretice din zona matematicii vectoriale, geometrie 2D și structuri de date pentru a implementa în mod eficient taskurile de proiect. De asemenea pe partea de multiplayer și persistența a datelor se vor trata noțiuni de protocoale de rețea, lucrul cu fișiere JSON, multithreading, minimizarea cantității de informații transmise pe rețea, prezentarea tool-urilor specifice pentru monitorizarea și testarea aplicației.

Seminariile și laboratoarele sunt dedicate strict implementării proiectelor de echipă.

Seminariile țin locul sedintelor de brainstorming, unde studenții din toate echipele discută la un loc, prezintă dificultățile întâmpinate, posibilele soluții la acestea și minimizarea codului de tip legacy care va trebui eventual modificat pentru viitoarele task-uri.

Laboratoarele sunt dedicate în parte sedintelor de tip SCRUM pe fiecare echipă și prezentarea realizărilor din parcursul etapei de dezvoltare. La fiecare etapă de sprint echipa trebuie să aleagă un membru MVP – cel care a contribuit cel mai mult la îndeplinirea task-urilor sprintului precedent. La sfârșitul laboratorului sunt prezentate task-urile din etapa următoare și sunt numiți noii team leaders.

La finalul semestrului, echipele de studenți vor prezenta un joc funcțional ce să întrunească toate elementele cerute la curs.

FORMA DE EXAMINARE:

Principala criteriu de evaluare va fi prezentarea proiectului de echipă. În acest pas, fiecare echipă va fi notată ca un întreg. Diferențele individuale dintre studenții din cadrul aceleiași echipe sunt făcute pe baza activității fiecărui student în cadrul proiectului: de câte ori a fost ales MVP de către colegii lui, felul în care au fost realizate task-urile din etapă în care studentul a fost team leader, etc.

PROGRAMĂ:

1. Curs introductiv; prezentarea cerințelor de proiect și a metodei de evaluare/ Prezentare introductivă a Engine-ului grafic
2. Game engine & Game architecture (MVC architecture, sprite animation, etc)
3. Input management, Rigidbody, forțe și coliziuni
4. Elemente de grafică și geometrie vectorială: rotații, traiectorii, accelerare
5. Design patterns, prefabs, lucrul cu asset store.
6. Game manager, menu flows, scene usage.
7. Persistența datelor și fișiere de tip JSON
8. Multithreading
9. Protocoale de rețea și securitate
10. Prezentare de proiecte.

BIBLIOGRAFIE:

1. Adams E., Dormans J., Game Mechanics - Advanced Game Design, New Riders, 2012
2. Sellers M., Advanced Game Design: A Systems Approach, Addison-Wesley Professional, 2017
3. Paris Buttfield-Addison, Jon Manning, Tim Nugent, Unity Game Development Cookbook, O'Reilly Media, 2019

Cursul va fi susținut de către Lect.dr. Stefan Popescu

FIȘA CURSULUI

Titlu: **Rutare și comutare în rețele de calculatoare**

Domeniul de licență: informatică

Specializarea: informatică (matematica-informatica)

Statutul: opțional

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

Semestrul: 5 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Scopul cursului este de a prezenta viitorilor specialiști în domeniul IT elementele de bază de rutare și comutare în rețele de calculatoare, precum și tehnologii avansate, rețele de programe de studii conexe și cariere entry-level de rețea. Limbajul utilizat pentru a descrie concepte de networking este proiectat pentru a fi ușor de înțeles de către studenți prin intermediul activităților interactive integrate care contribuie la consolidarea înțelegerii. Cursul conduce la deprinderea unei gândiri critice, rezolvarea de probleme, colaborare, precum și aplicarea practică a competențelor.

Cursul utilizează o gamă variată de instrumente de învățare multimedia, inclusiv clipuri video, abordează diferite stiluri de învățare (laboratoare hands-on și activități de învățare cu ajutorul simulatorului Packet Tracer, GNS3), ceea ce ajută studenții să dezvolte gândirea analitică și abilitățile de rezolvare a problemelor complexe.

PROGRAMA:

Capitolul 1: Introducere în Rutare și Transmiterea Pachetelor

Capitolul 2: Configurarea Sistemului de operare de Rețea

Capitolul 3: VLANuri

Capitolul 4: Concepte de Rutare

Capitolul 5: Rutare Inter-VLAN

Capitolul 6: Rutarea Statică

Capitolul 7: Rutare Dinamică

Capitolul 8: OSPF Single-Area

Capitolul 9: Liste de Control ale Accesului

Capitolul 10: DHCP

Capitolul 11: Network Address Translation pentru IPv4

BIBLIOGRAFIE:

1. A.Tanenbaum – Rețele de calculatoare (ediția V), Byblos.
2. S.Buraga, G.Ciobanu – Atelier de programare în rețele de calculatoare, Polirom, Iași, 2001.
3. Cisco CCNA Routing and Switching 200-120 Official Cert Guide Library, Wendell Odom
4. CCNA Portable Command Guide, 3rd Edition, Scott Empson.
5. *CISCO.COM*

Se lucrează pe platforma CISCO versiunea 5 din cadrul Facultății de Matematică și Informatică - Universitatea din București

Cursul va fi susținut de către Asist.dr. Mihaita Dragan.

FIȘA CURSULUI

Titlu: **Blockchain – concepte, tehnologii si aplicații**

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 sanatate 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 limbaje ca 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. Arhitecturi pentru calcul descentralizat. IPFS (o combinatie Kademia + BitTorrent + Git), NeoFS și FileCoin ca storage asociat 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. Limbaje pentru implementarea contractelor "smart" pe diferite platforme cu Solidity și Go
7. Platforma de execuție a contractelor smart ETHEREUM-EVM (Mașina virtuală Ethereum).
8. Scalabilitatea Blockchain: SEGWIT vs Gas. Testare și verificare automata de smart contracts in Solidity.
9. Arhitectura aplicațiilor descentralizate.
10. Elemente de securitate in 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 experiența 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 completă 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 bazată 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 semantică.
- [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, p 1047–1058.
- [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, CACLING, 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 ș.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 ș.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 limbă vorbită) al unui calculator ș.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 ș.a.) și înregistrează deja (în 2017) o acuratețe de 92,6%.

La finele acestui curs studentul va fi familiarizat cu principalele concepte de bază ale domeniului NLP, va fi văzut una dintre cele mai importante aplicații ale domeniului și va fi pregătit pentru a aborda alte tipuri de aplicații în viitor. De asemenea, va fi în posesia principalelor și totodată celor mai moderne instrumente puse la dispoziția programatorilor în domeniu.

PROGRAMĂ:

1. Preprocesarea datelor aparținând limbajului natural: operații și instrumente de bază
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 sintactică bazată pe constituenți, analiză sintactică de dependență)
4. Analiza limbajului natural la nivel semantic (dezambiguizare locală a sensului unui cuvânt polisemantic; dezambiguizare globală a sensului unui text)

CERINTE MINIMALE:

Cunoașterea 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
2. Florentina Hristea, Introducere în procesarea limbajului natural. Editura Universității din București, 2010
3. Tutorial Python: <https://www.learnpython.org/>

PROFESOR:

Cursul și 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. Dezvoltarea de aplicații mobile în Android Studio: configurarea unui proiect nou, Gradle, instrumente de debugging: Android Manager, Logcat
2. Android Manifest
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. Permișiunile unei aplicații Android
6. Intents: deschiderea unei alte aplicații din aplicația 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 layouts 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-rotăție, 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 harților, 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 Schwenkt, 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 cei cu rea intenție. 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 instrucțiune, 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; utilitate 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://pittest.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