

**CURSURI OPȚIONALE DE
INFORMATICĂ**

PROPUSE

**PENTRU ANUL
UNIVERSITAR 2019-2020**

**DOMENIUL DE LICENȚĂ:
INFORMATICĂ**

**SPECIALIZAREA:
INFORMATICĂ**

FIȘA CURSULUI

Titlu: **Aplicații web în Python și 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/>

FIȘA CURSULUI

Titlu: **Cloud Computing**

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:

Familiarizarea studenților cu paradigmele și principiile uneia dintre primele cinci tehnologii emergente cu un impact major în următorii 10 ani asupra economiei, științei și societății. Arhitecturi software pentru orchestrarea sistemelor de servicii disponibilizate prin cloud computing cum ar fi IaaS, PaaS, SaaS, DBaaS, DWaaS, Hadoop-as-a-Service etc. Studenții își vor dezvolta capacitatea de a orchestra servicii în cloud, de a integra resurse de calcul din cloud cu cele disponibilizate on-premises, de a identifica și implementa modele de stocare scalabile în funcție de cerințele de business, dar în același timp vor învăța să dezvolte aplicații în cloud cu DevOps.

PROGRAMĂ:

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

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 contururilor, 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.

FIȘA CURSULUI

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

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:

Oamenii pot interpreta fără efort ceea ce văd în lumea înconjurătoare. Scopul Vederii Artificiale este de a înzestra un sistem computerizat cu caracteristici asemănătoare, care "vede" și "înțelege" ceea ce se întâmplă în imagini. Pentru realizarea acestui deziderat este necesară utilizarea unor algoritmi eficienți care să extragă informații din pixeli pentru interpretarea imaginilor.

Cursul are ca obiectiv atât studierea conceptelor de bază cât și prezentarea unor tehnici de actualitate din domeniul Vederii Artificiale. În ultimii ani, tehnicile și metodele folosite în Vederea Artificială și-au dovedit aplicabilitatea din ce în ce mai răspândită pentru rezolvarea unor probleme importante cum ar fi recunoașterea facială, detectarea și recunoașterea obiectelor în imagini, detectarea și urmărirea obiectelor în video, recunoașterea caracterelor, recunoașterea semnelor de circulație, căutarea imaginilor după conținut și multe altele. Din acest motiv, cursul va pune un foarte mare accent pe partea de aplicabilitate a tehnicilor prezentate. Mai precis, fiecare curs va conține exemple practice și aplicații. La finalul cursului, studenții vor putea înțelege și aplica tehnicile de ultimă oră din domeniu pentru rezolvarea unor probleme interesante precum recunoașterea obiectelor, clasificarea imaginilor, detectare facială, etc.

FORMA DE EXAMINARE:

Dat fiind caracterul aplicativ al cursului, studenții vor fi evaluați pe baza temelor realizate acasă și în cadrul laboratorului. 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 vederea artificială. Concepte de bază.
2. Filtre pentru imagini. Muchii și gradienti. Textură și culoare.
3. Puncte de interes. Trăsături locale pentru imagini. Descriptorii SIFT și HOG.
4. Metode de învățare. Concepte de bază. Modelul Nearest Neighbors.
5. Metode kernel. Funcții de similaritate și clasificatori liniari: SVM, KRR, KDA.
6. Modele de învățare deep. Rețele neuronale convoluționale pentru recunoașterea obiectelor.

7. Recunoașterea obiectelor folosind modelul bag of visual words.
8. Detectarea facială folosind metoda ferestrei glisante.
9. Metode pentru recunoașterea gesturilor din video.

BIBLIOGRAFIE:

1. R. Szeliski. Computer Vision: Algorithms and Applications. Springer, 2010.
2. J. Shawe-Taylor, N. Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, 2004.
3. Simon Prince: Computer Vision: Models, Learning and Inference, Cambridge University Press, 2012. Draft disponibil online.
4. David Forsyth, Jean Ponce: Computer Vision: A modern approach (second edition), Prentice Hall, 2011.
5. Radu Ionescu, Marius Popescu: Knowledge Transfer between Computer Vision and Text Mining, Springer, 2016.
6. <http://www.vlfeat.org>
7. <http://www.vlfeat.org/matconvnet>

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 notiuni 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 notiunile 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, Redux și Redux. Realizare de aplicații simple care să demonstreze notiunile învățate. Arhitectura proiectelor bazate pe ReactJS în practică.

BIBLIOGRAFIE:

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

FIȘA CURSULUI

Titlu: **Introducere în învățare automată utilizand Python (*)**

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 prezinte studenților concepte de bază și tehnici de actualitate din domeniul învățării automate. Acesta va fi de tipul “hands-on” și, fără a ocoli noțiunile matematice, va pune accent pe înțelegerea intuitivă a conceptelor.

Cursul va avea un laborator asociat, care va pune studenții în situația de a trata probleme reale de învățare automată, folosind Python.

La finalul cursului, studenții vor fi familiari cu “bunele practici” ale învățării automate, fiind capabili să trateze riguros și corect o problemă, implicând analiza datelor, alegerea algoritmului și a parametrilor și evaluarea soluției.

Cursul este dedicat celor care sunt pasionați de acest domeniu sau care vor să urmeze ulterior o carieră în Machine Learning. Dificultatea este apropiată de cea a unui curs de Master și presupune un efort semnificativ pe parcursul semestrului. Studenții vor primi, însă, ajutorul necesar pentru a înțelege conceptele discutate, atât la curs cat și la laborator.

PROGRAMĂ:

- Noțiuni de bază de învățare automată.
- Evaluarea algoritmilor de învățare automată.
- Tehnici de învățare supervizată:
 - Clasificare: Regresie logistică, Arbori de decizie, KNN, SVM.
 - Regresie: Regresie liniară, Regresie Ridge, SVR.
- Rețele neurale:
 - Multilayer Perceptron
 - Rețele convoluționale
 - LSTM
- Tehnici de învățare nesupervizată:
 - Clustering: K-means, DBSCAN
 - Reducerea dimensionalității: PCA, T-sne

BIBLIOGRAFIE:

- Hands-On Machine Learning with Scikit-Learn and TensorFlow, A. Géron, O'Reilly Media, 2017
- Machine Learning, Tom Mitchell, McGraw Hill, 1997.
- Scikit-learn Documentation: <http://scikit-learn.org/stable/documentation.html>

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; Unelte 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 (detalii: Tennați; Autentificare; Utilizatori; Roluri; Roboți; Mediul; Procesuri; Coșuri și tranzacții; Joburi; 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

FIȘA CURSULUI

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

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 o introducere în mediul de dezvoltare JavaScript pentru server-side cu o aprofundare mai detaliată a platformei Node.js, nou apărută în aceasta direcție.

În cadrul cursului va fi introdus, în primă fază, limbajul JavaScript, concepte de bază și concepte avansate necesare dezvoltării de aplicații web atât client-side cât și server-side. A doua parte a cursului se va concentra pe aprofundarea platformei Node.js urmând ca la final, accentul să fie pus pe un framework cum ar fi ExpressJS.

La finalul cursului, studenții vor avea abilitățile necesare pentru a putea proiecta și dezvolta o aplicație full-stack JavaScript.

PROGRAMĂ:

1. JavaScript ca limbaj de programare: fundamente, concepte, pattern-uri
2. Arhitectura aplicațiilor web: concepte, n-tier, DDD (Domain Driven Design),
3. Javascript advanced, ES6, nvm
4. NodeJs (npm, Express, Postman)
5. RESTful APIs
6. Websockets
7. DB (mongoDB), redis
8. Testing
9. Build Systems (gulp)
10. Angular
11. Async/await

BIBLIOGRAFIE:

1. Documentatia online: <https://nodejs.org/api/>
2. Documentatia online: <https://github.com/vndmtrx/awesome-nodejs>

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., & Schafer, 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. Stănășilă, O., *Undine. Teorie și aplicații.*, Academia Română, 2010

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 proaspăt angajați, folosesc PHP la locul de muncă

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: verificare

Credite: 5

OBIECTIVE:

Blockchain este tehnologia despre care s-a vorbit cel mai mult în 2018! De aceea este importanta familiarizarea studentilor nostri cu tehnologia Blockchain de stocare și management al datelor descentralizate și imutabile, care valorifică ideea de consens peste un Ledger distribuit. Descentralizarea, procesului de distribuire a consensului bazat pe o abordare bazată pe distribuirea registrului public al tranzacțiilor implică auditarea tranzacționării prin intermediul secvențelor hash asociate și face din Blockchain o tehnologie de colaborare în medii lipsite de încredere (byzantine). Tehnologia inițial propusă de Satoshi Nakamoto ca registru public 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. Cursul acoperă domeniul Blockchain 1.0 și Blockchain 2.0.

Pe langa prezentarea conceptelor Blockchain, cursul va pune accentul si pe aspecte practice, prezentand framworkuri dar si studii concrete de caz, acestea fiind prezentate de invitati din industrie.

PROGRAMĂ:

1. Arhitecturi pentru calcul descentralizat. IPFS (o combinatie Kademia + BitTorrent + Git) și NeoFS.
2. Introducere în tehnologia Blockchain. Storage descentralizat. Protocoale (Bitcoin, Ethereum, Corda, Hyperledger).
3. Consensul în Blockchain-PoW (Ethereum), PoS (Ethereum), 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. BigChainDB implementează un contract generic)
6. Limbaje pentru implementarea contractelor "smart" pe diferite platforme: Solidity și NEO
7. Mediul de execuție al contractelor smart pe Ethereum EVM (Mașina virtuală Ethereum). "Gas"
8. Mediul de execuție al contractelor smart (NeoContracts) pentru monedele criptografice NEO și Gas pe NeoVM.
9. Elemente de securitate în blockchain. Testare și verificare automată de smart contracts în Solidity.

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.

FIȘA CURSULUI

Titlu: **Fundamentele rutării**

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:

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.

Se lucrează pe platforma CISCO versiunea 7.0.1 din cadrul Facultății de Matematică și Informatică - Universitatea din București.

PROGRAMĂ:

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:

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

FIȘA CURSULUI

Titlu: **Inițiere în cercetare și bioinformatică**

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 se adresează studenților interesați în cercetare în bioinformatică. Studenții vor fi expuși atât la elemente de bază din bioinformatică cât și la rezultate recente din bioinformatică și biologia sistemelor. 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 a lucrărilor științifice: peer-review, clasificarea revistelor și a conferințelor.

PROGRAMĂ:

1. Noțiuni elementare de biologie celulară, genetică și biochimie.
2. Experimentul lui Adleman pentru rezolvarea unei probleme NP.
3. Experimentul lui Lipton pentru rezolvarea SAT, alte rezultate din calculabilitatea bazată pe ADN.
4. Autoasamblare: experimentele lui Eric Winfree, Ned Seeman, Paul Rothemund, etc.
5. 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.
6. Biologie evoluționară: Aliniere afină, matrici de substituție: BLOSUM50, BLOSUM62, PAM150, etc.
7. Modele abstracte de calculabilitate bazată pe ADN și celule: Sisteme H, Sisteme P.
8. 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. L. M. Adleman, Molecular computation of solutions to combinatorial problems, Science, Vol. 266, Iss. 5187, 1994, pp. 1021-1024.
2. R. Lipton, DNA Solution of Hard Computational Problems, Science, Vol. 268, Iss. 5210, 1995, pp. 542-545.
3. E. Winfree, F. Liu, Lisa A. Wenzler, N. C. Seeman, Design and self-assembly of two-dimensional DNA crystals, Nature, Vol. 394, 1998, pp. 539-544
4. P. K. Rothemund, Folding DNA to create nanoscale shapes and patterns, Nature, 440, 2006, 297-302
5. N. C. Jones, P. A. Pevzner, An Introduction to Bioinformatics Algorithms (Computational Molecular Biology), The MIT Press, 2004, 456 pp., ISBN: 0262101068.
6. T. Smith, M. Waterman, Comparison of biosequences, Adv. in App. Mathematics, 2(4) 1981, 482-489.
7. S. 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.
8. B. Ma, J. Tromp, M. Li, PatternHunter: faster and more sensitive homology search, Bioinformatics, Vol. 18, Iss. 3, 2002, pp. 440-445.

FIȘA CURSULUI

Titlu: **Introducere in 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 isi propune sa ofere studentilor o perspectiva moderna si la zi asupra domeniului. Fiecare tema abordata va fi insotita de aplicatii concrete care vor viza diverse aspecte ale limbajului natural, cu o atentie speciala acordata limbii romane. Vor fi abordate aspecte computationale, cantitative si formale ale limbajului natural, facand distinctia clara intre analiza textelor si cea a limbilor naturale in general.. Studentii vor lua contact cu temele prioritare de cercetare in domeniu, in scopul deschiderii acestora spre doctorat. Vor fi prezentate cerinte punctuale ale diverselor firme interesate de aplicatii ale procesarii limbajului natural in chestiuni curente de productie.

PROGRAMĂ:

- [1] Probleme curente (practice, teoretice si de cercetare) in lingvistica matematica si computationala si in procesarea limbajului natural.
- [2] Aspecte cantitative ale limbajului natural.
- [3] Probleme de similaritate lingvistica .
- [4] Abordari si rezolvari computationale eficiente ale unor probleme de morfologie, fonologie si semantica.
- [5] Analiza computationala a amprentei stilistice.
- [6] Analiza computationala a documentelor.
- [7] Analiza de corpus. Detectarea de colocatii, detectare automata si analiza ortografica a cuvintelor inrudite.
- [8] Traducere automata.
- [9] Detectarea automata din texte a opiniilor, analiza computationala a polaritatii sentimentelor. Aplicatii in predictia optiunii 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. Universitatii Bucuresti, 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. Schutze. 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/>

FIȘA CURSULUI

Titlu: Introducere în **proiectarea jocurilor electronice**

Domeniul de licență: informatică

Specializarea: informatică

Statutul: opțional

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

Semestrul: 6 / anul III de studiu

Forma de examinare: verificare

Credite: 5

OBIECTIVE:

Obiectivul cursului este unul dublu:

În primul rând, în cazul prelegerilor de curs, studentul va aprofunda cunoștințe din zona geometriei 2D și structuri de date pentru a implementa în mod eficient taskurile de proiect. De asemenea pe partea de multiplayer și persistență 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.

Laboratoarele sunt dedicate strict implementării proiectelor de echipa. Întâlnirile de laborator țin locul ședințelor de brainstorming, unde studenții din toate echipele discută la un loc, prezintă dificultățile întâmpinate, posibile soluții la acestea și minimizarea codului de tip legacy care va trebui eventual modificat pentru viitoarele task-uri.

Laboratoarele sunt dedicate în parte ședințelor de tip SCRUM pe fiecare echipa 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 leadership.

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

FORMA DE EXAMINARE:

Echipele vor fi notate atât pe parcursul semestrului cât și pe baza unei prezentări a aplicației în variantă finală. Se va ține cont și de activitatea fiecărui student din echipă: cât de apreciat a fost în rândul colegilor, cât de bine s-a desfășurat workflow-ul pe taskurile pe taskurile la care acesta a fost team leader.

PROGRAMĂ:

1. Curs introductiv; prezentarea cerințelor de proiect și a metodei de evaluare
2. Game engine & Game arhitectură (MVC arhitectură, sprite animation, etc)
3. Elemente de grafică și geometrie vectorială: rotații, traiectorii, accelerare și coliziuni între obiecte
4. Persistența datelor și fișiere de tip JSON
5. Multithreading (X)
6. Trecerea de la o aplicație peer to peer la una cu server dedicat
7. Protocoale de rețea și securitate
8. Tool-uri pentru lucrul în rețea și monitorizarea traficului.
9. Optimizare de bandwidth
10. Virtualizare și containere
11. Portarea aplicației pe un third-party server
12. Amazon Web Services și Docker

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. Tanenbaum A. S., Wetherall D. J., Computer Networks, 5th ed, 2010.

FIȘA CURSULUI

Titlu: Invatare automata in arta vizuala (*)

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:

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 cu caracter artistic. Se vor studia algoritmi specifici unui astfel de set de date constrâns, precum recunoșterea stilului artistic a unei imagini sau transferul stilului artistic între două imagini.

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

Folosind cunoștințele acumulate pe parcurs, studenții vor realiza până la finalul cursului un sistem de recomandare de imagini cu conținut artistic.

PROGRAMĂ:

1. Introducere. Concepte de bază în Învățare automată.
2. Clasificarea imaginilor I. Abordarea clasică. Hand-coded features. Modele liniare.
3. Clasificarea imaginilor II. Data-Driven Approach. Rețele neuronale convoluționale.
4. Antrenarea rețelelor. Backpropagation. Metode de optimizare.
5. Software pentru deep learning: Caffe, Torch, TensorFlow
6. Transfer learning. De la clasificarea imaginilor la recunoșterea stilului artistic al imaginilor.
7. Transferarea stilului artistic. Studiu de caz - Prisma App.
8. Alte aplicații ale rețelelor convoluționale: detectarea obiectelor, segmentarea imaginilor

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. <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
4. http://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf
5. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd ed.. Pearson, 2009

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

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. Permișunile 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>

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 UIViewController, 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 .

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 grafică 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 contururilor, 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 grafică 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.

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 cauză-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 metaheuristice:** metode de căutare locale și metode globale; folosirea metodelor metaheuristice 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 acceptabile, 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