



FIȘA DISCIPLINEI

1. Date despre program

| | |
|---------------------------------------|--|
| 1.1 Instituția de învățământ superior | Universitatea “Alexandru Ioan Cuza” din Iași |
| 1.2 Facultatea | Facultatea de Informatică |
| 1.3 Departamentul | Informatică |
| 1.4 Domeniul de studii | Inteligență artificială |
| 1.5 Ciclul de studii | Licență |
| 1.6 Programul de studii / Calificarea | |

2. Date despre disciplină

| | | | | | | | |
|--|---|--------------|---|---------------------|---|--------------------------|----|
| 2.1 Denumirea disciplinei | Ingineria programării | | | | | | |
| 2.2 Titularul activităților de curs | Iftene Adrian, Moruz Alex | | | | | | |
| 2.3 Titularul activităților de seminar | Iftene Adrian, Moruz Alex, Olariu Florin, Cristian Simionescu, Mihai Costandache, Cosmin Irimia | | | | | | |
| 2.4 An de studiu | 2 | 2.5 Semestru | 2 | 2.6 Tip de evaluare | M | 2.7 Regimul disciplinei* | DF |

* OB – Obligatoriu / OP – Opțional

3. Timpul total estimat (ore pe semestru și activități didactice)

| | | | | | |
|--|----|--------------------|----|---------------|-----|
| 3.1 Număr de ore pe săptămână | 4 | din care: 3.2 curs | 2 | 3.3 laborator | 2 |
| 3.4 Total ore din planul de învățământ | 56 | din care: 3.5 curs | 28 | 3.6 laborator | 28 |
| Distribuția fondului de timp | | | | | ore |
| Studiu după manual, suport de curs, bibliografie și altele | | | | | 24 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren | | | | | 24 |
| Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri | | | | | 23 |
| Tutoriat | | | | | |
| Examinări | | | | | 12 |
| Alte activități | | | | | 1 |
| 3.7 Total ore studiu individual | | | | | 56 |
| 3.8 Total ore pe semestru | | | | | 125 |
| 3.9 Număr de credite | | | | | 5 |

4. Precondiții (dacă este cazul)

| | |
|-------------------|---|
| 4.1 De curriculum | Algoritmi și Programare. Programare orientată obiect |
| 4.2 De competențe | Programarea în limbaje de nivel înalt. Dezvoltarea și întreținerea aplicațiilor informatice. Proiectarea și gestiunea bazelor de date |

5. Condiții (dacă este cazul)

| | |
|------------------------------------|---|
| 5.1 De desfășurare a cursului | Sala de curs trebuie să dispună de video-proiector, conexiune la Internet și tablă pentru exemplificări |
| 5.2 De desfășurare a laboratorului | Sala de laborator trebuie să dispună de conexiune la Internet și tablă pentru exemplificări. Studenții au nevoie de calculatoare pe care să fie instalat un limbaj ce le permite să programeze orientat-obiect (Java, C#, Python, PhP, C++, etc.) |



6. Competențe specifice acumulate [[definițiile conceptelor de mai jos se găsesc la adresa http://docis.acpart.ro/uploads/Fisiere/Metodologie%20CNCIS.pdf](http://docis.acpart.ro/uploads/Fisiere/Metodologie%20CNCIS.pdf)]

| | |
|--------------------------------|---|
| Competențe profesionale | C1. Programarea în limbaje de nivel înalt. C2. Dezvoltarea și întreținerea aplicațiilor informatice. C3. Proiectarea și gestiunea bazelor de date. |
| Competențe transversale | CT1. Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatice de comunicare inter-personala, de relaționare și colaborare cu grupuri diverse |

7. Obiectivele disciplinei (din grila competențelor specifice acumulate) [la fel, detalii în documentul http://docis.acpart.ro/uploads/Fisiere/Metodologie%20CNCIS.pdf](http://docis.acpart.ro/uploads/Fisiere/Metodologie%20CNCIS.pdf)

| | |
|----------------------------------|--|
| 7.1 Obiectivul general | Construirea unei viziuni profesionale asupra procesului de dezvoltare a programelor cu ajutorul inteligenței artificiale (IA). Studenții vor învăța metode și tehnici avansate pentru realizarea unor programe complexe de o calitate superioară cu ajutorul IA în contextul respectării cerințelor clienților privind funcționalitatea, costurile și termenul limită. |
| 7.2 Obiectivele specifice | La finalizarea cu succes a acestei discipline, studenții vor fi capabili să: <ul style="list-style-type: none">▪ Explice cum pot fi folosite principiile programării orientate obiect și IA în dezvoltarea proiectelor de dimensiuni mari▪ Descrie pașii necesari implementării cu succes a unui proiect cu ajutorul IA pentru un client și vor putea descrie pașii necesari testării produsului construit cu ajutorul IA▪ Utilizeze tehnologiile Java și IA în implementarea componentelor funcționale și de testare ale unui proiect dat▪ Analizeze cerințele clientului cu IA, analizeze erorile software care apar pe parcursul dezvoltării proiectului cu IA, analizeze testele manuale pe care le execută asupra unui sistem▪ Calculeze necesarul de timp, necesarul de personal și necesarul de bani pentru dezvoltarea proiectului cerut de client cu IA |

8. Conținut

| 8.1 | Curs | Metode de predare | Observații (ore și referințe bibliografice) |
|------------|---|---|---|
| 1. | Conținutul cursului. Bibliografie. Motivație, Definiții, Erori celebre, Statistici. Modele de dezvoltare (Cascadă, Spirală) | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore, [1], [2], [3], [4], [5], [6] |
| 2. | Modele de dezvoltare (RUP, V-Model, XP, Agile, Lean, Scrum, AMDD, TDD). Ingineria cerințelor (Actor, Use Case) | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore, [1], [7], [8] |
| 3. | Modelare cu IA, Limbaje de Modelare, UML, Diagrame Use Case, Diagrame de Clase | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. Desenarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [1], [9] |



| | | | |
|-----|--|--|-------------------|
| 4. | Diagrame UML (Secvența, Colaborare, Stări, Activitate, Deployment, Pachete). Metodologia SCRUM | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. Problematizare. | 2 ore, [4] |
| 5. | Reverse Engineering. Prezentarea uneltelor necesare dezvoltării de proiecte: Github, Trello, GoogleDocs. | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore, [4], [10] |
| 6. | GRASP. Principii SOLID | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. Problematizare și lucrul pe exemple. | 2 ore, [2], [4] |
| 7. | Design Patterns cu IA: Definiții, Elemente, Clasificare, Paternuri creaționale | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore, [3], [5] |
| 8. | Recapitularea noțiunilor studiate: ingineria cerințelor, modelare, design patterns | Problematizare. Desenarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [3], [5] |
| 9. | Design Patterns cu IA: Paternuri creaționale, Paternuri structurale | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore, [3], [5] |
| 10. | Design Patterns cu IA: Paternuri comportamentale | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore, [3], [5] |
| 11. | Design Patterns cu IA: Concurență, Testare, Distribuite. Testare software | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore, [3], [5] |
| 12. | Testare software cu IA: Introducere, metode, procese Testare manuală vs Testare automată | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore, [11], [12] |
| 13. | Calitatea Programelor, Metrice, Copyright | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore |
| 14. | Conținutul cursului de la master: Swebok, SOA, QoS, BPMN, AOP, Refactorizare | Prezentare de slide-uri. Note de curs și tutoriale disponibile în versiune electronică. | 2 ore |

Bibliografie**Referințe principale:**

- [1] Ian Sommerville: Software Engineering, Addison Wesley, 2001
- [2] Craig Larman: Applying UML and Patterns, Addison Wesley, 2002
- [3] Erich Gamma, Richard Helm, Ralph Johnson, John Vissides: Design Patterns, Elements of Reusable Object-Oriented Software, Addison Wesley, 1998

Referințe suplimentare:

- [4] ArgoUML: <http://argouml-downloads.tigris.org/>
- [5] Preview of Patterns in Java Volume 2: http://www.mindspring.com/~mgrand/pattern_synopses2.htm
- [6] Failure rate: http://www.it-cortex.com/Stat_Failure_Rate.htm
- [7] RUP in the dialogue with Scrum: <http://www.ibm.com/developerworks/rational/library/feb05/krebs>
- [8] Agile Manifesto: <http://agilemanifesto.org/>
- [9] Requirements analysis process: <http://www.outsource2india.com/software/RequirementAnalysis.asp>
- [10] Chikofsky, E.J. and Cross, J.: Reverse Engineering and Design Discovery: A Taxonomy, January 1990
- [11] JUnit Test Example: <http://www.cs.unc.edu/~weiss/COMP401/s08-27-JUnitTestExample.doc>
- [12] Bug Life Cycle: <http://www.buzzle.com/editorials/4-6-2005-68177.asp>, <http://qastation.wordpress.com/2008/06/13/process-for-bug-life-cycle/>

| 8.2 | Laborator | Metode de predare | Observații (ore și referințe bibliografice) |
|-----|---|---|--|
| 1. | Recapitularea noțiunilor de programare orientată-obiect (moștenire, | Problematizare. Discuții. Schițarea la tablă a soluțiilor | 2 ore, [13] |



| | | | |
|-----|--|--|----------------------------|
| | polimorfism, incapsulare) | pentru o problemă dată. | |
| 2. | Fișa Cerințelor, UML - Diagrame de Clase | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [1], [2], [4], [9] |
| 3. | Fișa Cerințelor, UML - Diagrame Use Case, Diagrame de Clase | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [4], [9] |
| 4. | Folosirea metodologiei SCRUM în rezolvarea de probleme | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [4] |
| 5. | Reverse Engineering. Folosire uneltelor Github, Trello, GoogleDocs. | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [3], [4], [5] |
| 6. | Design Patterns – Paternuri creaționale, JUnit Testing | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [2], [4], [5], [11] |
| 7. | Lucrul la Proiect: Studiu de Risc, Ingineria Cerințelor | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [4], [9] |
| 8. | Recapitularea noțiunilor studiate | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [3], [5] |
| 9. | Lucrul la Proiect: Ingineria cerințelor, Diagrame UML | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [4], [9] |
| 10. | Lucrul la Proiect: Iterații SCRUM (Implementare, testare, integrare) | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [3], [4], [5] |
| 11. | Lucrul la Proiect: Iterații SCRUM (Implementare, testare, integrare) | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore, [3], [4], [5] |
| 12. | Lucrul la Proiect: Iterații SCRUM (Implementare, testare, integrare) | Problematizare. Discuții. Schițarea la tablă a soluțiilor pentru o problemă dată. | 2 ore |
| 13. | Lucrul la Proiect: Iterații SCRUM (Implementare, testare, integrare) | Prezentare folosind video-proiectorul. Problematizare. Discuții. | 2 ore, [11] |
| 14. | Lucrul la Proiect: Evaluare | Prezentare folosind video-proiectorul. Problematizare. Discuții. | 2 ore, [11] |

Bibliografie:**Referințe principale:**

[1] Ian Sommerville: Software Engineering, Addison Wesley, 2001

[2] Craig Larman: Applying UML and Patterns, Addison Wesley, 2002

[3] Erich Gamma, Richard Helm, Ralph Johnson, John Vissides: Design Patterns, Elements of Reusable Object-Oriented Software, Addison Wesley, 1998

Referințe suplimentare:

[4] ArgoUML: <http://argouml-downloads.tigris.org/>

[5] Preview of Patterns in Java Volume 2: http://www.mindspring.com/~mgrand/pattern_synopses2.htm

[6] Failure rate: http://www.it-cortex.com/Stat_Failure_Rate.htm

[7] RUP in the dialogue with Scrum: <http://www.ibm.com/developerworks/rational/library/feb05/krebs>

[8] Agile Manifesto: <http://agilemanifesto.org/>

[9] Requirements analysis process: <http://www.outsource2india.com/software/RequirementAnalysis.asp>

[10] Chikofsky, E.J. and Cross, J.: Reverse Engineering and Design Discovery: A Taxonomy, January 1990

[11] Junit Test Example: <http://www.cs.unc.edu/~weiss/COMP401/s08-27-JUnitTestExample.doc>



[12] Bug Life Cycle: <http://www.buzzle.com/editorials/4-6-2005-68177.asp>,
<http://qastation.wordpress.com/2008/06/13/process-for-bug-life-cycle/>

[13] Lucanu D.: Principiile programării orientate-obiect,
<http://thor.info.uaic.ro/~dlucanu/cursuri/poo/resurse/principiiPOO.pps>

9. Coroborarea conținutului disciplinei cu așteptările reprezentanților comunității, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

Companiile de IT din Iași și din țară folosesc în dezvoltarea de proiecte de dimensiuni mari noțiuni de inginerie software. Etapele necesare dezvoltării aplicațiilor de dimensiuni mari (ingineria cerințelor, modelare, implementare, integrare, testare, deployment) sunt discutate pe larg pe parcursul cursului, iar studenții învață noțiunile de bază și parcurg efectiv aceste etape. Pe parcursul verii studenții participă la sesiuni de stagii în cadrul firmelor, fiind implicați în proiecte reale, unde aplică practic pe proiecte reale noțiunile teoretice învățate la acest curs. În urma discuțiilor cu principalii angajatori, acest curs se actualizează de la an la an, adaptându-se la cerințele actuale ale pieței de IT și nu numai.

10. Evaluare

| Tip activitate | 10.1 Criterii de evaluare | 10.2 Metode de evaluare | 10.3 Pondere în nota finală (%) |
|---|---|--|---|
| 10.4 Curs | Punctaj minim examen: 40% din punctajul maxim ce poate fi obținut la examen | Test scris + Bonus pentru activitatea de la curs | 50 % |
| 10.5 Laborator | Punctaj minim laborator: 50% din punctajul maxim ce poate fi obținut la laborator (minim 50 % din punctajele temelor din primele 6 laboratoare și minim 50 % din punctajul proiectului) | Prezență + Prezentare de soluții la temele propuse săptămânal în primele 6 săptămâni + Lucrul la proiect + Bonus pentru activități suplimentare ce au legătură cu cursul de IP | 50 % (din care 50 % e lucrul la proiect) |
| 10.6 Standard minim de performanță [raportate la competențele definite la punctul 7. Obiectivele disciplinei] | | | |
| Studenții vor fi capabili să identifice principalele cerințe în urma discuției cu clientul și vor fi capabili să modeleze soluția pe care acesta o așteaptă cu ajutorul IA. Implementarea se va reduce la construirea claselor principale și a atributelor acestora cu ajutorul IA, fără implementarea metodelor de bază. Testarea proiectului o vor putea face din punct de vedere al testării automate cu IA și manuale. Punctaj minim laborator: 50% din punctajul maxim ce poate fi obținut la laborator fără bonusuri (minim 50 % din temele de laborator și minim 50 % proiect) Punctaj minim examen: 40% din punctajul maxim ce poate fi obținut la examen Pentru studenții care îndeplinesc criteriile de promovare nota finală se stabilește împărțind punctajul obținut la maximul punctajelor fără bonus, iar rezultatul este înmulțit cu 10. Studentul care participă la examen, va primi o notă, altfel va fi considerat absent. Dacă unul din criteriile de promovare nu este îndeplinit, studentul va obține o nota mai mică sau egală cu 4. | | | |

Data completării
29.09.2024

Titulari de curs
Prof. Dr. Iftene Adrian
Lector Dr. Moruz Alex

Titulari de seminar
Prof. Dr. Iftene Adrian
Lector Dr. Moruz Alex
Drd. Olariu Florin
Drd. Costandache Mihai
Drd. Irimia Cosmin
Drd. Simionescu Cristian

Data avizării în departament

Director de departament
Prof. Dr. Lucanu Dorel