

## FIȘA DISCIPLINEI

## 1. Date despre program

|                                   |   |
|-----------------------------------|---|
| Instituția de învățământ superior | Universitatea „Ștefan cel Mare” din Suceava                               |
| Facultatea                        | Facultatea de Inginerie Electrică și Știința Calculatoarelor              |
| Departamentul                     | Departamentul de Calculatoare   |
| Domeniul de studii                | Calculatoare și tehnologia informației                                    |
| Ciclul de studii                  | Program de conversie profesională   |
| Programul de studii               | Program de conversie profesională – Tehnologia informației și comunicării |

## 2. Date despre disciplină

|                                    |   |           |   |                   |        |
|------------------------------------|---|-----------|---|-------------------|--------|
| Denumirea disciplinei              | <b>PROIECTAREA ALGORITMILOR</b>   |           |   |                   |        |
| Titularul activităților de curs    | Prof. univ. dr. ing. Radu-Daniel VATAVU   |           |   |                   |        |
| Titularul activităților aplicative | Asistent univ. drd. ing. Alexandru-Ionuț ȘIEAN  |           |   |                   |        |
| Anul de studiu                     | 2   | Semestrul | 4 | Tipul de evaluare | Examen |
| Regimul disciplinei                | Categorია formativă a disciplinei<br>DF - fundamentală, DD - în domeniu, DS - de specialitate, DC – complementară |           |   |                   | DD     |
|                                    | Categorია de opționalitate a disciplinei:<br>DI - impusă, DO - opțională, DF – facultativă                        |           |   |                   | DI     |

## 3. Timpul total estimat (ore alocate activităților didactice)

|  |    |      |    |         |   |                            |    |         |   |
|--|----|------|----|---------|---|----------------------------|----|---------|---|
| I a) Număr de ore pe săptămână                           | 4  | Curs | 2  | Seminar | - | Laborator/lucrări practice | 2  | Proiect | - |
| I b) Totalul de ore pe semestru din planul de învățământ | 56 | Curs | 28 | Seminar | - | Laborator/lucrări practice | 28 | Proiect | - |

|  |     |
|--|-----|
| II Distribuția fondului de timp pe semestru:   | ore |
| II a) Studiul după manual, suport de curs, bibliografie și notițe                                    | 36  |
| II b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren | 24  |
| II c) Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri                          | 56  |
| II d) Tutoriat   | -   |
| III Examinări  | 3   |
| IV Alte activități (precizați):  | -   |

|  |     |
|--|-----|
| Total ore studiu individual II (a+b+c+d) | 116 |
| Total ore pe semestru (Ib+II+III+IV)     | 175 |
| Numărul de credite                       | 7   |

## 4. Precondiții (acolo unde este cazul)

|            |   |
|------------|---|
| Curriculum | - |
| Competențe | - |

## 5. Condiții (acolo unde este cazul)

|                        |   |   |
|------------------------|---|---|
| Desfășurare a cursului | Laptop/PC, slide-uri suport pentru curs, exemple de programe funcționale pe calculator, conexiune la Internet |   |
| Desfășurare aplicații  | Seminar   | -   |
|                        | Laborator/lucrări practice  | Unități laptop/PC având instalat mediul de dezvoltare Visual Studio (Community, Code, etc.), ghid de lucrări practice în format electronic, conexiune la Internet |
|                        | Proiect   | -   |

## 6. Competențe specifice acumulate

|                         |   |
|-------------------------|---|
| Competențe profesionale | CP1. Operarea cu fundamente științifice, ingineresti și ale informaticii<br>CP2. Proiectarea componentelor hardware, software și de comunicații<br>CP3. Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații<br>CP6. Proiectarea sistemelor inteligente |
| Competențe transversale | -   |

## 7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

|                                   |   |
|-----------------------------------|---|
| Obiectivul general al disciplinei | Introducere în proiectarea și analiza algoritmilor prin prezentarea, exemplificarea, analiza și discutarea diverselor metode de proiectare a algoritmilor, evaluare a complexității teoretice și implementare de soluții algoritmice pentru probleme specifice.   |
|                                   | <ul style="list-style-type: none"> <li>- Însușirea de metode de proiectare a algoritmilor (Divide et Impera, programarea dinamică) ca fundamente științifice, ingineresti și ale informaticii.</li> <li>- Însușirea metodelor de proiectare a algoritmilor pentru prelucrarea datelor în diverse reprezentări</li> <li>- Însușirea metodelor de analiză a complexității algoritmilor în vederea evaluării și îmbunătățirii performanțelor sistemelor software.</li> </ul> |

## 8. Conținuturi

| Curs  | Nr. ore | Metode de predare  | Observații |
|---|---------|--|------------|
| <b>1. Modele de calcul și analiză a complexității algoritmilor</b><br>1.1. Instrumente de măsurare a complexității unui algoritm. Complexitatea temporală. Complexitatea spațială. Ordine de complexitate. Notățiile asimptotice $O$ , $\Theta$ și $\Omega$ . Exemple.<br>1.2. Demonstrarea modului de analiză a complexității algoritmilor pentru probleme specifice.  | 3       | Expunerea, prelegerea, conversația, exemplificarea, demonstrația |            |
| <b>2. Metoda Divide et Impera: introducere</b><br>2.1. Prezentarea principiilor metodei Divide et Impera.<br>2.2. Introducerea metodei Divide et Impera prin discutarea și analiza unor probleme exemplu (determinarea valorii extreme dintr-o mulțime de numere reale, algoritmul căutării binare) și realizarea de comparații cu alte soluții algoritmice pentru rezolvarea acestor probleme.<br>2.3. Analiza complexității algoritmilor Divide et Impera. Teorema Master. Exemple de aplicare a teoremei Master pentru diverse probleme.   | 3       |  |            |
| <b>3. Metoda Divide et Impera: aplicații</b><br>3.1. Algoritmul sortării prin interclasare.<br>3.2. Algoritmul sortării rapide.<br>3.3. Analiza comparativă a performanțelor diverselor metode de sortare a datelor. Realizarea de conexiuni cu alți algoritmi studiați în cadrul unor discipline anterioare. Identificarea algoritmilor de sortare implementați în cadrul diverselor platforme de dezvoltare a aplicațiilor software.<br>3.4. Demonstrarea aplicării metodei Divide et Impera pentru diverse probleme specifice, <i>e.g.</i> , problema determinării numărului de inversiuni ale unei permutări, problema tăieturilor, probleme de partiționare a figurilor geometrice, etc. | 3       |  |            |
| <b>4. Metoda programării dinamice: introducere</b><br>4.1. Prezentarea principiilor metodei programării dinamice. Rezolvarea problemelor algoritmice folosind principiile programării dinamice.<br>4.2. Exemplificarea metodei prin analiza unor probleme specifice, <i>e.g.</i> , șirul lui Fibonacci, calculul combinărilor $C_{n,k}$ . Realizarea de comparații cu alte abordări pentru rezolvarea acestor probleme și analiza complexității diverselor abordări.  | 3       |  |            |
| <b>5. Metoda programării dinamice: aplicații (1)</b><br>5.1. Problema subsecvenței de sumă maximă. Discutarea diverselor soluții algoritmice și analiza complexității acestora.<br>5.2. Rezolvarea eficientă a problemelor de natură algoritmică folosind structuri de tip Lookup Table (LUT) sau Summed Area Table (SAT).  | 3       |  |            |

|  |   |  |  |
|--|---|--|--|
| <p><b>6. Metoda programării dinamice: aplicații (2)</b><br/> 6.1. Problema celei mai lungi subsecvențe crescătoare (LIS/LAS) și aplicații privind rezolvarea de probleme asemănătoare întâlnite în practică.<br/> 6.2. Problema celei mai lungi subsecvențe de elemente comune (LCS) și aplicații privind rezolvarea de probleme asemănătoare întâlnite în practică.<br/> Analiza complexității algoritmice a soluțiilor problemelor prezentate.</p>   | 3 | Expunerea, prelegerea, conversația, exemplificarea, demonstrația |  |
| <p><b>7. Metoda programării dinamice: aplicații (3)</b><br/> 7.1. Algoritmi eficienți pentru determinarea drumurilor optime în probleme de parcurgere a unui labirint.<br/> 7.2. Rezolvarea problemelor de drumuri în labirint și prezentarea de aplicații privind probleme asemănătoare întâlnite în practică. Analiza complexității algoritmice a soluțiilor prezentate.</p>   | 3 |  |  |
| <p><b>8. Algoritmi în grafuri: introducere</b><br/> 8.1. Modalități de reprezentare a grafurilor: matricea de adiacență și liste de adiacență. Exemple.<br/> 8.2. Parcurgerea informației dintr-un graf: algoritmul parcurgerii în lățime (BFS) și algoritmul parcurgerii în adâncime (DFS).</p>   | 3 |  |  |
| <p><b>9. Algoritmi în grafuri: componente conexe</b><br/> 9.1. Probleme practice rezolvate folosind arborele parcurgerii în adâncime.<br/> 9.2. Determinarea componentelor conexe ale unui graf neorientat. Analiza complexității algoritmice. Rezolvarea de probleme exemplu.</p>   | 2 |  |  |
| <p><b>10. Algoritmi în grafuri: drumuri</b><br/> 10.1. Drumuri în graf. Drumuri de cost minim.<br/> 10.2. Algoritmul Roy-Floyd-Warshall pentru determinarea tuturor drumurilor de cost minim dintre nodurile unui graf.<br/> 10.3. Algoritmul lui Dijkstra.<br/> 10.4. Aplicații ale determinării drumurilor într-un graf pentru rezolvarea de probleme specifice.</p>   | 2 |  |  |
| <b>Bibliografie</b>  |   |  |  |
| <ol style="list-style-type: none"> <li>1. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, Clifford Stein. 2022. Introduction to Algorithms, 4th Ed. MIT Press (sau traducerea în limba română: Introducere în algoritmi, Ed. Agora)</li> <li>2. Donald E. Knuth. 2022 Ed. The Art of Computer Programming, vol. 1, Addison-Wesley Professional (sau traducerea în limba română: Arta programării calculatoarelor, vol. I – Algoritmi Fundamentali, Ed. Teora)</li> <li>3. Donald E. Knuth. 2022 Ed. The Art of Computer Programming, vol. 2, Addison-Wesley Professional (sau traducerea în limba română: Arta programării calculatoarelor, vol. II – Algoritmi Seminumerici, Ed. Teora)</li> <li>4. Donald E. Knuth. 2022 Ed. The Art of Computer Programming, vol. 3, Addison-Wesley Professional (sau traducerea în limba română: Arta programării calculatoarelor, vol. III – Sortare și căutare, Ed. Teora)</li> <li>5. Aditya Y. Bhargava. 2024. Grokking Algorithms, 2nd Ed. Manning</li> <li>6. Jay Wengrow. 2020. A Common-Sense Guide to Data Structures and Algorithms, Second Edition: Level Up Your Core Programming Skills, 2nd Ed. Pragmatic Bookshelf</li> <li>7. Panos Louridas. 2020. Algorithms, The MIT Press Essential Knowledge series. The MIT Press</li> <li>8. Tim Roughgarden. 2022. Algorithms Illuminated: Omnibus Edition, 1st Ed.</li> <li>9. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi. 2018. How to Design Programs: An Introduction to Programming and Computing, 2nd Ed., MIT Press</li> <li>10. Robert Sedgewick, Kevin Wayne. 2011. Algorithms, 4th Ed. Addison-Wesley Professional</li> <li>11. Gayle Laakmann McDowell. 2015. Cracking the Coding Interview: 189 Programming Questions and Solutions, 6th Ed., CareerCup</li> <li>12. Cristian A. Giumale. 2004. Introducere în analiza algoritmilor. Ed. Polirom</li> <li>13. Cristian A. Giumale. 2000. Un atelier de programare. Ed. Agora</li> <li>14. Radu-Daniel Vatavu. 2013. Ghid practic pentru analiza și proiectarea algoritmilor. Aplicații în C#. MatrixRom</li> <li>15. Ștefan-Gheorghe Pentiu, Radu-Daniel Vatavu. 2009. Algoritmi și metode de programare în Java. Ed. Universității Ștefan cel Mare din Suceava</li> <li>16. Doina Logofătu. 2007. Algoritmi fundamentali în C++: Aplicații. Ed. Polirom</li> <li>17. Mihai Oltean. 1999. Proiectarea și implementarea algoritmilor. Ed. Agora</li> <li>18. Radu-Daniel Vatavu, Lisa Anthony, Jacob O. Wobbrock. 2018. \$Q: A super-quick, articulation-invariant stroke-</li> </ol> |   |  |  |

gesture recognizer for low-resource devices. In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18). ACM, New York, NY, USA

19. Radu-Daniel Vatavu. 2017. Improving Gesture Recognition Accuracy on Touch Screens for Users with Low Vision. In Proceedings of the CHI Conference on Human Factors in Computing Systems. ACM, USA, 4667-4679

20. Yosra Rekik, Radu-Daniel Vatavu, Laurent Grisoni. 2014. Match-Up & Conquer: A Two-Step Technique for Recognizing Unconstrained Bimanual and Multi-Finger Touch Input. In Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14). ACM, New York, NY, USA, 201-208

21. Radu-Daniel Vatavu, Lisa Anthony, Jacob O. Wobbrock. 2012. Gestures as point clouds: A SP recognizer for user interface prototypes. Proceedings of the ACM International Conference on Multimodal Interaction. ACM, 273-280

#### **Bibliografie minimală**

1. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, Clifford Stein. 2022. Introduction to Algorithms, 4th Ed. MIT Press (sau traducerea în limba română: Introducere în algoritmi, Ed. Agora)
2. Gayle Laakmann McDowell. 2015. Cracking the Coding Interview: 189 Programming Questions and Solutions, 6th Ed., CareerCup
3. Radu-Daniel Vatavu. 2013. Ghid practic pentru analiza si proiectarea algoritmilor. Aplicații în C#, MatrixRom

| <b>Aplicații (Seminar/laborator/lucrări practice/proiect)</b>                                 | <b>Nr. ore</b> | <b>Metode de predare</b>                                    | <b>Observații</b>  |
|---|----------------|---|--|
| 1. Analiza complexității algoritmilor (1)   | 2              | Lucrări practice, experimentul, conversația, studiul de caz |  |
| 2. Analiza complexității algoritmilor (2)   | 2              |   |  |
| 3. Metoda Divide et Impera: Introducere   | 2              |   |  |
| 4. Metoda Divide et Impera: Algoritmul sortării prin interclasare. Algoritmul sortării rapide | 2              |   |  |
| 5. Test practic și evaluare   | 2              | Probă practică  | Rezolvarea unei probleme algoritmice prin implementarea corectă într-un limbaj de programare |
| 6. Metoda programării dinamice: Introducere. Subsecvența crescătoare de lungime maximă        | 2              | Lucrări practice, experimentul, conversația, studiul de caz |  |
| 7. Metoda programării dinamice: Subsecvența comună de lungime maximă                          | 2              |   |  |
| 8. Metoda programării dinamice: Parcurgerea labirintului                                      | 2              |   |  |
| 9. Test practic și evaluare   | 2              | Probă practică  | Rezolvarea unei probleme algoritmice prin implementarea corectă într-un limbaj de programare |
| 10. Grafuri: Introducere. Probleme de reprezentare  | 2              | Lucrări practice, experimentul, conversația, studiul de caz |  |
| 11. Grafuri: Parcurgere și componente conexe  | 2              |   |  |
| 12. Grafuri: Drumuri de cost minim  | 2              |   |  |
| 13. Test practic și evaluare  | 2              | Probă practică  | Rezolvarea unei probleme algoritmice prin implementarea corectă într-un limbaj de programare |
| 14. Rezolvarea de probleme algoritmice specifice  | 2              | Lucrări practice, test practic, experimentul                |  |

Prezența la activitățile obligatorii este reglementată de “Regulamentul cadru privind evaluarea studenților”. Recuperarea activităților aplicative obligatorii programate pe parcursul semestrului se face în conformitate cu precizările acestui regulament. De asemenea, în conformitate cu prevederile articolului 5.2.3 ale regulamentului sus-menționat, activitățile aplicative programate pe parcursul semestrului pot fi echivalate cu realizarea și susținerea de proiecte de complexitate ridicată din tematica disciplinei, cu acordul cadrului didactic titular.

1. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, Clifford Stein. 2022. Introduction to Algorithms, 4th Ed. MIT Press (sau traducerea în limba română: Introducere în algoritmi, Ed. Agora)
2. Donald E. Knuth. 2022 Ed. The Art of Computer Programming, vol. 1, Addison-Wesley Professional (sau traducerea în limba română: Arta programării calculatoarelor, vol. I – Algoritmi Fundamentali, Ed. Teora)
3. Donald E. Knuth. 2022 Ed. The Art of Computer Programming, vol. 2, Addison-Wesley Professional (sau traducerea în limba română: Arta programării calculatoarelor, vol. II – Algoritmi Seminumerici, Ed. Teora)
4. Donald E. Knuth. 2022 Ed. The Art of Computer Programming, vol. 3, Addison-Wesley Professional (sau traducerea în limba română: Arta programării calculatoarelor, vol. III – Sortare și căutare, Ed. Teora)
5. Aditya Y. Bhargava. 2024. Grokking Algorithms, 2nd Ed. Manning

6. Jay Wengrow. 2020. A Common-Sense Guide to Data Structures and Algorithms, Second Edition: Level Up Your Core Programming Skills, 2nd Ed. Pragmatic Bookshelf
7. Panos Louridas. 2020. Algorithms, The MIT Press Essential Knowledge series. The MIT Press
8. Tim Roughgarden. 2022. Algorithms Illuminated: Omnibus Edition, 1st Ed.
9. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi. 2018. How to Design Programs: An Introduction to Programming and Computing, 2nd Ed., MIT Press
10. Robert Sedgewick, Kevin Wayne. 2011. Algorithms, 4th Ed. Addison-Wesley Professional
11. Gayle Laakmann McDowell. 2015. Cracking the Coding Interview: 189 Programming Questions and Solutions, 6th Ed., CareerCup
12. Cristian A. Giumale. 2004. Introducere în analiza algoritmilor. Ed. Polirom
13. Cristian A. Giumale. 2000. Un atelier de programare. Ed. Agora
14. Radu-Daniel Vatavu. 2013. Ghid practic pentru analiza și proiectarea algoritmilor. Aplicații în C#. MatrixRom
15. Ștefan-Gheorghe Pentiuc, Radu-Daniel Vatavu. 2009. Algoritmi și metode de programare în Java. Ed. Universității Ștefan cel Mare din Suceava
16. Doina Logofătu. 2007. Algoritmi fundamentali în C++: Aplicații. Ed. Polirom
17. Mihai Oltean. 1999. Proiectarea și implementarea algoritmilor. Ed. Agora
18. Radu-Daniel Vatavu, Lisa Anthony, Jacob O. Wobbrock. 2018. \$Q: A super-quick, articulation-invariant stroke-gesture recognizer for low-resource devices. In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18). ACM, New York, NY, USA
19. Radu-Daniel Vatavu. 2017. Improving Gesture Recognition Accuracy on Touch Screens for Users with Low Vision. In Proceedings of the CHI Conference on Human Factors in Computing Systems. ACM, USA, 4667-4679
20. Yosra Rekik, Radu-Daniel Vatavu, Laurent Grisoni. 2014. Match-Up & Conquer: A Two-Step Technique for Recognizing Unconstrained Bimanual and Multi-Finger Touch Input. In Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14). ACM, New York, NY, USA, 201-208
21. Radu-Daniel Vatavu, Lisa Anthony, Jacob O. Wobbrock. 2012. Gestures as point clouds: A \$P recognizer for user interface prototypes. Proceedings of the ACM International Conference on Multimodal Interaction. ACM, 273-280

#### **Bibliografie minimală**

1. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, Clifford Stein. 2022. Introduction to Algorithms, 4th Ed. MIT Press (sau traducerea în limba română: Introducere în algoritmi, Ed. Agora)
2. Gayle Laakmann McDowell. 2015. Cracking the Coding Interview: 189 Programming Questions and Solutions, 6th Ed., CareerCup
3. Radu-Daniel Vatavu. 2013. Ghid practic pentru analiza și proiectarea algoritmilor. Aplicații în C#, MatrixRom

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

Conținutul disciplinei este în concordanță cu conținutul disciplinelor similare de la programele de studiu din cadrul altor universități din țară și străinătate, și anume “Introduction to Algorithms”, MIT, MA, USA (<https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020>); “Intro to Algorithms and their Limitations”, Harvard University, MA, USA (<https://salil.seas.harvard.edu/classes/intro-algorithms-and-their-limitations-fall21>); „Introduction to Algorithms”, Washington University, WA, USA (<https://courses.cs.washington.edu/courses/cse421>); „Proiectarea algoritmilor”, Universitatea Politehnica București, (<https://ocw.cs.pub.ro/courses/pa>). De asemenea, conținutul cursului este în acord cu recomandările ACM/IEEE privind Computer Engineering Curricula- Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering în ce privește “1. Fundamental algorithmic design principles, 2. Analysis of algorithmic behavior, including tradeoffs between algorithms, 3. Classic algorithms for such common tasks as searching and sorting, 4. Design and analysis of application-specific algorithms” (p. 71), <https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf>.

#### **10. Evaluare**

10.1. Standard minim de performanță evaluare la curs

10.2. Standard minim de performanță evaluare la activitatea aplicativă

| <b>Tip activitate</b> | <b>Criterii de evaluare</b>  | <b>Metode de evaluare</b>  | <b>Pondere din nota finală</b> |
|-----------------------|--|--|--------------------------------|
| Curs                  | Expunerea clară a conceptelor teoretice și corectitudinea operării cu noțiuni specifice proiectării algoritmilor ca fundamente științifice, ingineresti și ale informaticii. Rezolvarea corectă a problemelor de natură algoritmică pentru prelucrarea datelor în diverse reprezentări și pentru diverse aplicații. Analiza corectă a complexității teoretice a algoritmilor în vederea îmbunătățirii performanțelor | Probă mixtă, alcătuită din:<br>- lucrare scrisă și<br>- test practic pe calculator (rezolvarea unei probleme de natură algoritmică) cu verificarea orală a abordării folosite pentru rezolvarea testului practic | 50%                            |

|  |   |  |     |
|--|---|--|-----|
|  | sistemelor software.  |  |     |
| Seminar  | -   | -  | -   |
| Laborator/lucrări practice   | Corectitudinea rezolvării de probleme de natură algoritmică și a implementării acestora folosind un limbaj de programare și o platformă de dezvoltare a aplicațiilor software. Gradul de îndeplinire a activităților din lucrările practice de laborator și realizarea de teme suplimentare acestora. Participarea la diverse activități practice în legătură cu obiectivele specifice ale disciplinei. | Probe practice pe calculator, observația sistematică, verificarea temelor. | 50% |
| Proiect  | -   | -  | -   |
| <b>Standard minim de performanță</b>   |   |  |     |
| <ul style="list-style-type: none"> <li>- Însușirea principalelor noțiuni, idei, concepte din cadrul proiectării și analizei algoritmilor</li> <li>- Cunoașterea problemelor de bază din domeniul proiectării și analizei algoritmilor</li> <li>- Implementarea de soluții algoritmice pentru problemele de bază din domeniul proiectării algoritmilor</li> </ul> |   |  |     |

|                   |                               |   |
|-------------------|-------------------------------|---|
| Data completării  | Semnătura titularului de curs | Semnătura cadrului didactic coordonator |
| <b>23.09.2024</b> |                               |   |

|                   |                                      |
|-------------------|--------------------------------------|
| Data avizării     | Semnătura responsabilului de program |
| <b>24.09.2024</b> |                                      |

|                              |                                       |
|------------------------------|---------------------------------------|
| Data avizării în departament | Semnătura directorului de departament |
| <b>25.09.2024</b>            |                                       |

|  |                     |
|--|---------------------|
| Data aprobării în consiliul facultății | Semnătura decanului |
| <b>27.09.2024</b>                      |                     |