# FIŞA DISCIPLINEI

*Algoritmi şi Programare*

Anul universitar 2025-2026

**1. Date despre program**

| | |
|---|---|
| 1.1. Instituţia de învăţământ superior | Universitatea Babeş-Bolyai |
| 1.2. Facultatea | Facultatea de Matematică şi Informatică |
| 1.3. Departamentul | Departamentul de Informatică |
| 1.4. Domeniul de studii | Matematică |
| 1.5. Ciclul de studii | Licenţă |
| 1.6. Programul de studii / Calificarea | Matematică – limba de studiu română |
| 1.7. Forma de învăţământ | Cu frecventa |

**2. Date despre disciplină**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.1. Denumirea disciplinei | **Algoritmi şi Programare** | | | | | Codul disciplinei | **MLR5115** |
| 2.2. Titularul activităţilor de curs | Lect. Dr. Găceanu Radu Dan | | | | | | |
| 2.3. Titularul activităţilor de seminar | Lect. Dr. Găceanu Radu Dan | | | | | | |
| 2.4. Anul de studiu | 1 | 2.5. Semestrul | 1 | 2.6. Tipul de evaluare | C | 2.7. Regimul disciplinei | Obligatoriu |

**3. Timpul total estimat** (ore pe semestru al activităţilor didactice)

| | | | | | |
|---|---|---|---|---|---|
| 3.1. Număr de ore pe săptămână | **6** | din care: 3.2. curs | **2** | 3.3. seminar/ laborator/proiect | **2 sem 2 lab** |
| 3.4. Total ore din planul de învăţământ | 84 | din care: 3.5. curs | 28 | 3.6 seminar/laborator/proiect | **56** |
| **Distribuţia fondului de timp pentru studiul individual (SI) şi activităţi de autoinstruire (AI)** | | | | | **ore** |
| Studiul după manual, suport de curs, bibliografie şi notiţe (AI) | | | | | 14 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren | | | | | 12 |
| Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii şi eseuri | | | | | 14 |
| Tutoriat (consiliere profesională) | | | | | 8 |
| Examinări | | | | | 18 |
| Alte activităţi | | | | | |
| **3.7. Total ore studiu individual (SI) şi activităţi de autoinstruire (AI)** | | | | **66** | |
| **3.8. Total ore pe semestru** | | | | **150** | |
| **3.9. Numărul de credite** | | | | **6** | |

**4. Precondiţii** (acolo unde este cazul)

| | |
|---|---|
| 4.1. de curriculum | |
| 4.2. de competenţe | |

**5. Condiţii** (acolo unde este cazul)

| | |
|---|---|
| 5.1. de desfăşurare a cursului | Sală cu proiector |
| 5.2. de desfăşurare a seminarului/ laboratorului | Laboratoare echipate cu medii de dezvoltare Python; sala cu proiector (seminar) |

**6.1. Competenţele specifice acumulate**[1]

| | |
|---|---|
| **Competenţe profesionale/esenţiale** | • elaborarea şi analiza unor algoritmi pentru rezolvarea problemelor |
| **Competenţe transversale** | • aplicarea regulilor de muncă riguroasă şi eficientă, manifestarea unor atitudini responsabile faţă de domeniul didactico-ştiinţific, pentru valorificarea optimă şi creativă a propriului potenţial în situaţii specifice, cu respectarea principiilor şi a normelor de etică profesională<br>• desfăşurarea eficientă şi eficace a activităţilor organizate în echipă<br>• utilizarea eficientă a surselor informaţionale şi a resurselor de comunicare şi formare profesională asistată, atât în limba română, cât şi într-o limbă de circulaţie internaţională |

**6.2. Rezultatele învăţării**

| | |
|---|---|
| **Cunoştinţe** | Absolventul/a este familiarizat/ă cu metodele de prelucrare a datelor şi cu instrumentele de vizualizare a rezultatelor obţinute. |
| **Aptitudini** | Absolventul/a este capabil/ă să analizeze literatura de specialitate şi să utilizeze instrumente de sprijinire a cercetării. |
| **Responsabilităţi şi autonomie** | Absolventul/a are capacitatea de a interpreta articole sau cărţi din literatura de specialitate şi de a încorpora idei şi rezultate din literatura de specialitate în prezentările lor scrise şi orale. |

**7. Obiectivele disciplinei** (reieşind din grila competenţelor acumulate)

| | |
|---|---|
| **7.1 Obiectivul general al disciplinei** | • Să cunoască concepele de baza ale ingineriei software (proiectare, implementare şi intreţinere) şi să invețe limbajul de programare Python. |

[1] Se poate opta pentru competenţe sau pentru rezultatele învăţării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va şterge tabelul aferent celeilalte opţiuni, iar opţiunea păstrată va fi numerotată cu 6.

| | |
|---|---|
| **7.2 Obiectivele specifice** | • Să cunoască conceptele de baza ale programarii<br>• Să cunoască conceptele de baza ale ingineriei software<br>• Să folosească instrumente de baza pentru construirea programelor<br>• Să învețe limbajul Python si instrumente de dezvoltare pentru programarea, executia si depanarea programelor Python.<br>• Să-și însușească un stil de programare conform celor mai bune recomandări practice. |

**8. Conţinuturi**

| 8.1 Curs | Metode de predare | Observaţii |
|---|---|---|
| 1. Introducere în procese de dezvoltare software<br>Ce este programarea: algoritm, program, elemente de bază Python, interpretor Python, roluri în ingineria software; Cum scriem programe: enunț problemă, cerințe, proces de dezvoltare dirijat de funcționalități; Exemple: calculator | Expunere interactivă; Explicaţie; Conversaţie; Exemple; Demonstraţie didactică | |
| 2. Programare procedurală<br>Tipuri structurate: liste, tuple, dictionare; Functii: cazuri de testare, definire, variabile, apel, transmiterea parametrilor, functii anonime; Cum scriem functii:programare dirijata de teste, refactorizari | Expunere interactivă; Explicaţie; Conversaţie; Exemple; Demonstraţie didactică | |
| 3. Programare modulara<br>Ce este un modul: modul Python, domeniul variabilelor, pachete, module standard, distribuire module; Cum organizam codul sursa: responsabilitati, single responsibility principle, separation of concerns, dependency, coupling, cohesion; Eclipse+PyDev | Expunere interactivă; Explicaţie; Conversaţie; Exemple; Demonstraţie didactică | |
| 4. Tipuri definite de utilizator<br>Cum definim tipuri noi; Incapsulare, ascunderea informatiei, tipuri abstracte de date; Introducere programarea orientata obiect –clasa implementarea unui TAD | Expunere interactivă; Explicaţie; Conversaţie; Exemple; Demonstraţie didactică | |
| 5-6. Programarea orientata obiect<br>Mecanismele programarii orientate obiect; Dezideratele proiectarii obiectuale: Cuplare redusa; Coeziune ridicata; Abstractizare corespunzatoare; Complexitate gestionabila; Principii si sabloane de proiectare: SOLID; DDD: entity, value object, repository, service; IoC, app coordinator, dependency injection; GOF: singleton, strategy;<br>Operatii CRUD;<br>Arhitectura stratificata. | Expunere interactivă; Explicaţie; Conversaţie; Exemple; Demonstraţie didactică | |
| 7. Proiectarea programelor<br>Diagrame UML; Sabloane Grasp; Exceptii | Expunere interactivă; Explicaţie; Conversaţie; Exemple; Demonstraţie didactică | |
| 8. Persistenta şi procesarea datelor<br>DTO, Filter, lambda | Expunere interactivă; Explicaţie; Conversaţie; Exemple; Demonstraţie didactică | |
| 9. Testarea si inspectarea programelor<br>Black box testing, white box testing; Unit testing, integration testing; Program inspection: coding style, refactoring | Expunere interactivă; Explicaţie; Conversaţie; Exemple; Demonstraţie didactică | |
| 10-11. Complexitatea Algoritmilor, recursivitate, căutare, sortare | Expunere interactivă; Explicaţie; Conversaţie; Exemple; | |

| | | |
|---|---|---|
| Recursivitate directa si indirecta; Căutare secvențială; Căutare binară  Complexitatea algoritmilor; BubbleSort; SelectionSort; InsertionSort; QuickSort; MergeSort; Cmplexitatea algoritmilor | Demonstrație didactică | |
| 12. Sisteme de versionare a codului<br>Introducere in Git | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 13. Metode de rezolvare a problemelor<br>Metoda divizarii; Backtracking; Greedy; Programare dinamică | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 14. Colocviu - Evaluare | | |
| Bibliografie | | |
| 1. Kent Beck. Test Driven Development: By Example. Addison-Wesley Longman, 2002. See also Testdriven development. http://en.wikipedia.org/wiki/Test-driven_development<br>2. Martin Fowler. Refactoring. Improving the Design of Existing Code. Addison-Wesley, 1999. See also http://refactoring.com/catalog/index.html<br>3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms 3rd ed., 2009.<br>4. Craig Larman. Applying UML and Patterns. An Introduction to Object Oriented Analysis and Design, 2004.<br>5. The Python language reference. http://docs.python.org/py3k/reference/index.html<br>6. The Python standard library. http://docs.python.org/py3k/library/index.html | | |
| 8.2 Seminar / laborator | Metode de predare | Observații |
| 1. Introducere în procese de dezvoltare software<br>Ce este programarea: algoritm, program, elemente de bază Python, interpretor Python, roluri în ingineria software; Cum scriem programe: enunț problemă, cerințe, proces de dezvoltare dirijat de funcționalități; Exemple: calculator | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 2. Programare procedurală<br>Tipuri structurate: liste, tuple, dictionare; Functii: cazuri de testare, definire, variabile, apel, transmiterea parametrilor, functii anonime; Cum scriem functii:programare dirijata de teste, refactorizari | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 3. Programare modulara<br>Ce este un modul: modul Python, domeniul variabilelor, pachete, module standard, distribuire module; Cum organizam codul sursa: responsabilitati, single responsibility principle, separation of concerns, dependency, coupling, cohesion; Eclipse+PyDev | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 4. Tipuri definite de utilizator<br>Cum definim tipuri noi; Incapsulare, ascunderea informatiei, tipuri abstracte de date; Introducere programarea orientata obiect –clasa implementarea unui TAD | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 5-6. Programarea orientata obiect<br>Mecanismele programarii orientate obiect; Dezideratele proiectarii obiectuale: Cuplare redusa; Coeziune ridicata; Abstractizare corespunzatoare; Complexitate gestionabila; Principii si sabloane de proiectare: SOLID; DDD: entity, value object, repository, service; IoC, app coordinator, dependency injection; GOF: singleton, strategy;<br>Operatii CRUD;<br>Arhitectura stratificata. | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 7. Proiectarea programelor<br>Diagrame UML; Sabloane Grasp; Exceptii | Expunere interactivă; Explicație; Conversație; Exemple; | |

| | | |
|---|---|---|
| | Demonstrație didactică | |
| 8. Persistenta și procesarea datelor<br>DTO, Filter, lambda | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 9. Testarea si inspectarea programelor<br>Black box testing, white box testing; Unit testing, integration testing; Program inspection: coding style, refactoring | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 10-11. Complexitatea Algoritmilor, recursivitate, căutare, sortare<br>Recursivitate directa si indirecta; Căutare secvențială; Căutare binară  Complexitatea algoritmilor; BubbleSort; SelectionSort; InsertionSort; QuickSort; MergeSort; Cmplexitatea algoritmilor | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 12. Sisteme de versionare a codului<br>Introducere in Git | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| 13-14. Pregatire examene. | Expunere interactivă; Explicație; Conversație; Exemple; Demonstrație didactică | |
| | | |
| | | |
| | | |

Bibliografie


1. Kent Beck. Test Driven Development: By Example. Addison-Wesley Longman, 2002. See also Testdriven development. http://en.wikipedia.org/wiki/Test-driven_development
2. Martin Fowler. Refactoring. Improving the Design of Existing Code. Addison-Wesley, 1999. See also http://refactoring.com/catalog/index.html
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms 3rd ed., 2009.
4. Craig Larman. Applying UML and Patterns. An Introduction to Object Oriented Analysis and Design, 2004.
5. The Python language reference. http://docs.python.org/py3k/reference/index.html
6. The Python standard library. http://docs.python.org/py3k/library/index.html


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

- Cursul respecta recomandările IEEE și ACM legate de Curiculla pentru specializarea Informatică.
- Cursul face parte din programul de studiu de la majoritatea universităților importante din România și din străinătate.
- Conținutul cursului este considerat de companiile soft ca fiind important pentru un nivel mediu de cunoștințe în programare.

**10. Evaluare**

| Tip activitate | 10.1 Criterii de evaluare | 10.2 Metode de evaluare | 10.3 Pondere din nota finală |
|---|---|---|---|
| 10.4 Curs | Corectitudinea și completitudinea cunoștințelor acumulate. Capacitatea de a proiecta și implementa programe scrise in limbajul Python | Examen scris | 20% |

| | | | |
|---|---|---|---|
| 10.5 Seminar/laborator | Abilitatea de a scrie şi depana un program Python | Examen practic | 70% |
| | Programele scrise în timpul semestrului | Documentaţie; teme | 10% |
| 10.6 Standard minim de performanţă | | | |
| • Nota finala minim 5. | | | |

**11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)[2]**

*Nu se aplică.*

Data completării:
15.04.2025

Semnătura titularului de curs

Lect. Dr. Găceanu Radu Dan

Semnătura titularului de seminar

Lect. Dr. Găceanu Radu Dan

Data avizării în departament:
...

Semnătura directorului de departament

Conf.dr. Adrian STERCA

[2] Păstraţi doar etichetele care, în conformitate cu *Procedura de aplicare a etichetelor ODD în procesul academic*, se potrivesc disciplinei şi ştergeţi-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ştergeţi-le pe toate şi scrieţi "*Nu se aplică.*".