**Fişa disciplinei**

**1. Date despre program**

|  |  |
| --- | --- |
| 1.1 Instituţia de învăţământ superior | **Universitatea Babeş-Bolyai Cluj-Napoca** |
| 1.2 Facultatea | Facultatea de Matematică şi Infrormatică |
| 1.3 Departamentul | Informatică |
| 1.4 Domeniul de studii | Informatică |
| 1.5 Ciclul de studii | Licență |
| 1.6 Programul de studiu / Calificarea | Informatică (în limba germană) |

**2. Date despre disciplină**

|  |  |
| --- | --- |
| 2.1 Denumirea disciplinei | Fundamentele programării |
| 2.2 Titularul activităţilor de curs | Lect. dr. Cătălin Rusu |
| 2.3 Titularul activităţilor de seminar | Lect. dr. Cătălin Rusu |
| 2.4 Anul de studiu | 1 | 2.5 Semestrul | 1 | 2.6. Tipul de evaluare | E | 2.7 Regimul disciplinei | Obligatorie |

**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 | 4 |
| 3.4 Total ore din planul de învăţământ | 84 | Din care: 3.5 curs | 28 | 3.6 seminar/laborator | 56 |
| Distribuţia fondului de timp: | Std. |
| Studiul după manual, suport de curs, bibliografie şi notiţe | 14 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren | 12 |
| Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri | 14 |
| Tutoriat | 8 |
| Examinări | 18 |
| Alte activităţi: .................. | - |
| 3.7 Total ore studiu individual | 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, plus proiector
 |
| 5.2 De desfăşurare a seminarului/laboratorului | * Laboratoare echipate cu Python
 |

**6. Competenţele specifice acumulate**

|  |  |
| --- | --- |
| **Competenţe profesionale** | C1.1 Descrierea adecvată a paradigmelor de programare şi a mecanismelor de limbaj specifice, precum şi identificarea diferenţei dintre aspectele de ordin semantic şi sintactic.C1.2 Explicarea unor aplicaţii soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoştinţele de bazăC1.3 Elaborarea codurilor sursă adecvate şi testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificaţii de proiectare date |
| **Competenţe transversale** | CT1 Aplicarea regulilor de muncă organizată şi eficientă, a unor atitudini responsabile faţă de domeniul didactic-ştiinţific, pentru valorificarea creativă a propriului potenţial, cu respectarea principiilor şi a normelor de etică profesionalăCT3 Utilizarea unor metode şi tehnici eficiente de învăţare, informare, cercetare şi dezvoltare a capacităţilor de valorificare a cunoştinţelor, de adaptare la cerinţele unei societăţi dinamice și de comunicare în limba română și într-o limbă de circulație internațională |

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

|  |  |
| --- | --- |
| 7.1 Obiectivul general al disciplinei | * Să cunoască concepele de bază ale ingineriei software (proiectare, implementare si intreținere) și să invețe limbajul de programare Python.
 |
| 7.2 Obiectivele specifice | * Să cunoască conceptele de bază ale programării
* Să cunoască conceptele de bază ale ingineriei software
* Să folosească instrumente de bază pentru construirea programelor
* Să invețe limbajul Python și instrumente de dezvoltare pentru programarea, execuția și depanarea programelor Python
 |

**8. Conţinuturi**

|  |  |  |
| --- | --- | --- |
| 8.1 Curs | Metode de predare | Observaţii |
| **1. Introducere in procese de dezvoltare software*** 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 (FDD)
* Exemple: calculator
 | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **2. Programare procedurală*** Tipuri structurate: liste, tuple, dicționare
* Funcții: cazuri de testare, definire, cariabile, apel
* Transmiterea parametrilor
* Funcții anonime
* Cum scriem funcții:programare dirijată de teste, refactorizări
 | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **3. Programare modulara*** Ce este un modul: modul Python, domeniul variabilelor, pachete, module standard, distribuire module
* Cum organizăm codul sursă: responsabilități, single responsibility principle, separation of concerns, dependency, coupling, cohesion
* Arhitecturi software stratificate
* Eclipse+PyDev/Pycharm
 | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 1. **Tipuri definite de utilizator**
* Cum definim tipuri noi
* Incapsulare, ascunderea informației, tipuri abstracte de date
 | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **5. Principii de proiectare si programare*** Problema: program cu operații CRUD pe entități de un tip dat
* Arhitectura stratificată: UI, Domeniu, Infrastructura
* Șabloane GRASP
* Șabloane DDD: entity, validator, repository, controller
* Principii: Information Expert, Low Coupling, High Cohesion, Protected Variation, Single responsibility, Dependency Injection
 | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **6. Programare orientată pe obiecte*** Obiecte și clase
* Diagrame UML
* Moștenire
* Exceptii
 | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **7. Proiectarea programelor*** Top down and bottom up strategies:
* Organizarea elementelor UI si relația cu alte straturi
 | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **8. Testarea și inspectarea programelor** | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **9. Recursivitate** | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **10. Complexitatea algoritmilor** | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **11. Metoda Backtracking**  | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **12. Algoritmi de căutare** | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **13 Algoritmi de sortare*** BubbleSort
* SelectionSort
* InsertionSort
* QuickSort
* MergeSort
* Cmplexitatea algoritmilor
 | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **14. Recapitulare** | * Expunere interactivă
* Conversație
 |  |
| Bibliografie1. *The Python language reference*.<http://docs.python.org/py3k/reference/index.html>
2. *The Python standard library*.<http://docs.python.org/py3k/library/index.html>
3. *The Python tutorial*.<http://docs.python.org/tutorial/index.html>

**În limba germană:**1. Allan B. Downey, Programmieren lernen mit Python, O'Reilly ISBN 978-3-86899-946-4
2. Manfred Baumgartner, Martin Klonk, Helmut Pichler, Richard Seidl, Siegfried Tanczos, Agile Testing, Hanser ISBN: 978-3-446-43194-2
3. Thomas Theis, Einstieg in Python: Ideal für Programmieranfänger
 |
| 8.2 Seminar / laborator | Metode de predare | Observaţii |
| 1. Programe Python | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 2. Programare procedurală | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 3. Programare modulară | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 4. Tipuri definite de utilizator | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 5. Principii de proiectare | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 6. POO | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 7. Proiectare | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 8. Testare și inspectare | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 9. Recursivitate | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 10. Complexitatea algoritmilor | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 11. Backtracking | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 12. Metoda înjumătățirii. Algoritmi de căutare | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 13. Pregătirea examenului practic | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| 14: Pregătirea examenului scris | * Expunere interactivă
* Explicație
* Conversație
* Exemple
* Demonstrație didactică
 |  |
| **Bibliografie**1. *The Python language reference*.<http://docs.python.org/py3k/reference/index.html>
2. *The Python standard library*.<http://docs.python.org/py3k/library/index.html>
3. *The Python tutorial*.<http://docs.python.org/tutorial/index.html>

**În limba germană:**1. *Robert Sedgewick:* Algorithmen (2. Auflage, Pearson Studium 2002)
2. Martin von Löwis, Nils Fischbeck , **Python 2 ,** Addison-Wesley-Longman ( 2000)
3. Tobias Himstedt and Klaus Mänzel , **Mit Python programmieren,** dpunkt.Verlag, 1999
 |

**9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului**

|  |
| --- |
| Cursul respecta recomandările IEEE şi ACM legate de Curiculla pentru specializarea Informatică. Cursul face parte din programul de studiu de la majoritatea universităților importante din Romănia 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 | Cunostintele acumulate | Examen scris | 30 % |
|  | Examen Intermediar | 20 % |
| 10.5 Seminar/laborator | Scrierea unui program | Examen practic | 20 % |
| Programele scrise in timpul semestrului | Documentatie | 30 % |
| 10.6 Standard minim de performanţă |  |  |  |
| Nota fiecărui exercițiu (laborator) trebuie să fie mai mare de 5. Pentru a promova examenul practic, nota minimă trebuie să fie 5. Pentru a promova examenul scris, nota minimă trebuie să fie 5. Pentru a promova examenul intermediar, nota minimă trebuie să fie 5. Doar nota finală se rotunjește la întreg.Participare obligatorie: Seminar: 75%; Laborator: 90%. |

Data completării Semnătura titularului de curs Semnătura titularului de seminar

15.04.2024 Lect. Dr. Cătălin Rusu Lect. Dr. Cătălin Rusu

Data avizării în departament Semnătura directorului de departament

 Conf. dr. Adrian Sterca