**fişa disciplinei**

**1. Date despre program**

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

**2. Date despre disciplină**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2.1 Denumirea disciplinei | | | **Programare logică** | | | | | | |
| 2.2 Titularul activităţilor de curs | | | | | **Conf. dr. Săcărea Christian** | | | | |
| 2.3 Titularul activităţilor de seminar | | | | | **Asist. dr. Florin Albisoru** | | | | |
| 2.4 Anul de studiu | **2** | 2.5 Semestrul | | **3** | | 2.6. Tipul de evaluare | **C** | 2.7 Regimul disciplinei | **Obligatorie** |
| 2.8. Cod disciplină | MLG5143 | | | | | | | | |

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3.1 Număr de ore pe săptămână | 4 | | Din care: 3.2 curs | | 2 | 3.3 seminar/laborator | 1+1 |
| 3.4 Total ore din planul de învăţământ | 56 | | Din care: 3.5 curs | | 28 | 3.6 seminar/laborator | 28 |
| Distribuţia fondului de timp: | | | | | | | ore |
| Studiul după manual, suport de curs, bibliografie şi notiţe | | | | | | | 25 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren | | | | | | | 20 |
| Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri | | | | | | | 30 |
| Tutoriat | | | | | | | 10 |
| Examinări | | | | | | | 14 |
| Alte activităţi: .................. | | | | | | | - |
| 3.7 Total ore studiu individual | | 94 | |
| 3.8 Total ore pe semestru | | 150 | |
| 3.9 Numărul de credite | | 6 | |

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

|  |  |
| --- | --- |
| 4.1 de curriculum | Fundamentele programării, Logica computationala, Structuri de date si algoritmi |
| 4.2 de competenţe |  |

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

|  |  |
| --- | --- |
| 5.1 De desfăşurare a cursului | * Sală de curs cu videoproiector |
| 5.2 De desfăşurare a seminarului/laboratorului | * Sală de laborator cu calculatoare dotate cu limbajele de programare GCLisp, CLisp şi TurboProlog |

**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 in mod adecvat cunoştinţele de bază  C1.3 Elaborarea codurilor sursă adecvate şi testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificaţii de proiectare date  C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente | |
| **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 profesională  **CT3** 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 deprindă studentul cu paradigma programării declarative (programarea funcţională şi programarea logică). |
| 7.2 Obiectivele specifice | * Să introducă câte un limbaj de programare pentru fiecare din aceste paradigme (Common Lisp şi Prolog). * Să inducă ideea utilizării acestor paradigme în funcţie de necesităţile aplicaţiilor. * Să asigure baza necesară urmării unor cursuri avansate de programare declarativă. |

**8. Conţinuturi**

|  |  |  |
| --- | --- | --- |
| **8.1 Curs** | Metode de predare | Observaţii |
| *Programare Logică. Limbajul PROLOG* |  |  |
| 1. Programare şi limbaje de programare. Programare imperativa vs. programare declarativă. Introducere.  Recursivitate. Exemple. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 2. Elemente fundamentale ale limbajului Prolog. Fapte şi reguli Prolog. Intrebări. Strategia de control în Prolog. Variabile şi propoziţii compuse. Variabile anonime. Reguli de definire a potrivirilor. Model de flux. Secţiunile unui program Prolog. Exemple. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 3. Programul Prolog. Domenii predefinite. Întrebări interne şi externe. Predicate cu aritate multiplă. Simbolul IF (Prolog) şi instructiunea IF (alte limbaje). Directive de compilare. Expresii aritmetice şi comparaţii. Operaţii de intrare / ieşire. Şiruri de caractere. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 4. Backtracking. Controlarea backtracking-ului. Predicatele fail şi ! (cut). Utilizarea lui !. Tipuri de tăieturi. Predicatul “not”. Liste Prolog. Recursivitate. Exemple de tratare a backtracking-ului. Găsirea tuturor soluţiilor în acelaşi timp. Exemple de predicate Prolog. Predicate nedeterministe | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 5. Obiecte compuse si functori. Unificarea obiectelor compuse. Argumente de tipuri multiple; liste eterogene. Compararea obiectelor compuse. Backtracking cu ciclări. Exemple de proceduri recursive. Cadrul stivei. Optimizarea prin recursivitate de coadă. Utilizarea tăieturii pentru păstrarea recursivităţii de coadă. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 6. Structuri de date recursive. Arborii ca structuri de date. Construirea si traversarea unui arbore. Arbori de căutare. Baza de date internă a sistemului Prolog. Secţiunea database. Declararea bazei de date interne. Predicate relativ la operaţii cu baza de date internă.. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 7. Examen scris PROLOG | Lucrarea scrisă |  |
| *Programare Funcţională. Limbajul LISP* |  |  |
| 8. Importanţa programării funcţionale ca noua metodologie de programare. Istoric şi prezentare a limbajului LISP. Elemente de bază Lisp. Structuri dinamice de date. Reguli sintactice şi semantice. Clasificarea funcţiilor Lisp. Funcţii primitive în Lisp. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 9. Predicate de bază în Lisp. Predicate pentru liste; pentru numere. Funcţii logice şi aritmetice. Definirea funcţiilor utilizator. Ramificarea prelucrărilor. Metoda variabilei colectoare. Exemple. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 10. Gestiunea simbolurilor. Alte funcţii de acces la liste. OBLIST si ALIST. Funcţii cu caracter destructiv. Comparaţii. Alte funcţii interesante. Exemple. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 11. Mecanisme definiţionale evoluate Forma EVAL. Forme funcţionale; funcţiile FUNCALL si APPLY. Expresii LAMBDA. Expresii LABEL. Exemple | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 12. Generatori, argumente funcţionale. Funcţii MAP. Forme iterative. Exemple. |  |  |
| 13. Alte elemente ale limbajului Lisp. Structuri de date. Macrodefiniţii. Argumente opţionale. Exemple. | * Expunerea interactivă * Explicaţia * Conversaţia * Demonstraţia didactică |  |
| 14. Examen scris LISP | Lucrarea scrisă |  |
| **Bibliografie**  In limba germana:   1. GOOS, G., ZIMMERMANN, W., Vorlesungen uber Informatik, Band 1, Grundlagen und funktionales Programmieren, Springer, 2006. 2. LIPPE, W-M., Funktionale und Applikative Programmierung, Springer 2009. 3. Cordes/Kruse/Langendorfer/Rust: Prolog — Eine methodische Einfuhrung Vieweg, 1990, 2. Auflage, 245 Seiten 4. Hanus: Problemlosen mit Prolog Teubner, 1987, 2. Auflage, 224 Seiten 5. Kleine Buning/Schmitgen: Prolog, Teubner, 1988, 2. Auflage, 311 Seiten   **In alte limbi**   1. CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012 2. POP H.F., SERBAN G., Programare in Inteligenta Artificiala - Lisp si Prolog, Editura Albastra, Cluj-Napoca, 2003 3. http://www.lpa.co.uk, Logic Programming 4. FIELD A., Functional Programming, Addison Wesley, New York, 1988. 5. WINSTON P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984. | | |
| **8.2 Laborator si seminar** | Metode de predare | Observaţii |
|  |  | Laboratorul este structurat sub forma a 2 ore din 2 în 2 săptămâni. |
| Lab 1/Sem1: Recursivitate | * Lucrare de laborator * Explicaţia * Conversaţia * Modelarea | * Se predă tema primită la Lab 1 * Se primeşte tema pentru Lab 2 |
| Lab 2/Sem 2: Liste în Prolog | * Lucrare de laborator * Explicaţia * Conversaţia * Modelarea | * Se predă tema primită pentru Lab 2 * Se primeşte tema pentru Lab 3 |
| Lab 3/Sem3: Arbori în Prolog. Gestiunea listelor în Prolog. | * Lucrare de laborator * Explicaţia * Conversaţia * Modelarea | * Se predă tema primită pentru Lab 3 * Se primeşte tema pentru Lab 4 |
| Lab 4/Sem4: Backtracking în Prolog | * Lucrare de laborator * Explicaţia * Conversaţia * Modelarea | * 1 oră * Se predă tema primită pentru Lab 4 * Se primeşte tema pentru Lab 5 |
| Lab 4/Sem 5: Proba practică Prolog | Lucrare practică | 1 oră |
| Lab 5/Sem 5: Programare recursivă în Lisp | * Lucrare de laborator * Explicaţia * Conversaţia * Modelarea | * Se predă tema primită pentru Lab 5 * Se primeşte tema pentru Lab 6 |
| Lab 6/Sem 6: Folosirea funcţiilor MAP. | * Lucrare de laborator * Explicaţia * Conversaţia * Modelarea | * Se predă tema primită pentru Lab 6 * Se primeşte tema pentru Lab 7 |
| Lab 7/Sem 7: Programare iterativă în Lisp | * Lucrare de laborator * Explicaţia * Conversaţia * Modelarea | * 1 oră: Se predă tema primită pentru Lab 7 |
| Lab 7/Sem 7: Proba practică Lisp | Lucrare practică | ● 1 oră |
| **Bibliografie in limba germana**     1. Hölldobler S., Logik und Logikprogrammierung: Synchron Wissenschaftsverlag der Autoren, Heidelberg, 2001 2. Weisweber W., Logische Programmierung in der Praxis, Redline GmbH , 2000 3. LIPPE, W-M., Funktionale und Applikative Programmierung, Springer 2009.   Bibliografie generala:   1. CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012 2. Documentatia produselor: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp. 3. <http://www.swi-prolog.org/> | | |

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

|  |
| --- |
| * Conţinutul disciplinei este în concordanţă cu ceea ce se face în alte centre universitare din tară și din străinătate. * Conţinutul disciplinei asigură cunoștințele fundamentale necesare pentru programare ȋn Lisp și Prolog la eventualii angajatori. |

**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 cunoștinţelor asimilate. * Gradul de asimilare a limbajului de specialitate. | Evaluare scrisă (în timpul semestrului): examen parţial Prolog (curs 7) | 30% |
| Evaluare scrisă (în timpul semestrului): examen parţial Lisp (curs 14) | 30% |
| 10.5 Laborator | * Implementarea în Lisp şi Prolog a problemelor de laborator * Redactarea documentaţiei de laborator * Respectarea termenelor de predare. | Documentaţii şi programe | 10% |
| Proba practică Prolog (1 oră, lab. 4) | 15% |
| Proba practică Lisp (1 oră, lab. 7) | 15% |
| 10.6 Standard minim de performanţă | | | |
| * Cunoașterea elementelor fundamentale de teorie, rezolvarea unei aplicaţii simple într-un limbaj declarativ. Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoaştere şi înţelegere a domeniului, că este capabil să exprime cunoştinţele într-o formă coerentă, că are capacitatea de a stabili anumite conexiuni şi de a utiliza cunoştinţele în rezolvarea unor probleme. * Pentru promovare sunt necesare urmatoarele criterii minimale: nota minim 4 la fiecare din lucrările scrise, predarea a minim 5 lucrări de laborator şi nota finală minim 5. | | | |

Data completării Semnătura titularului de curs Semnătura titularului de seminar

12.04.2024 **Conf. dr. Săcărea Christian** **Asist. dr. Florin Albisoru**

Data avizării în departament Semnătura directorului de departament

Conf. dr. Adrian Sterca