

## FIȘA DISCIPLINEI

### 1. DATE DE IDENTIFICARE

Titlul Disciplinei: Structuri de date si algoritmi

Titular/i de disciplină: Conf. Dr. Ing. Gheorghe Petrescu

Tipul: pregătire generală

Număr ore curs: 42 ore

Număr ore aplicații: 28 ore

Numărul de puncte de credit: 5

Semestrul: 2

Pachetul: aria curriculara comună

Precondiții: parcurgerea și/sau promovarea următoarelor discipline:

Informatică, Analiză matematică, Limbaje de Programare C/C++ (din liceu și facultate).

### 2. OBIECTIVELE DISCIPLINEI

- pentru curs

Obiectul disciplinei SDA îl constituie studiul unor structuri de date simple și analiza unor categorii de algoritmi. Formarea unor deprinderi de reprezentare a obiectelor statice și de prelucrare a obiectelor dinamice precum și capacitatea de control a performanțelor programului relativ la raportul memorie consumată / viteza de execuție.

- pentru aplicații

Tematica orelor de laborator și aplicațiile implementate au ca obiectiv însușirea cunoștințelor de la curs și urmăresc realizarea de programe în concordanță cu structura cursului. Formarea unor aptitudini legate de proiectarea unor structuri de date care sa permită scrierea unor programe performante.

**3. COMPETENȚE SPECIFICE (din spectrul de competențe al programului de studii)** Studenții se vor familiariza cu noțiunile elementare privind sistemul de calcul. De asemenea, vor învăța noțiuni specifice privind tehnica analizei și sintezei sistemelor informatice precum și cu tehnica programării în limbaje de nivel înalt (C/C++).

### 4 CONTINUTUL TEMATIC (SYLLABUS)

- a. Curs:

Capitolul	Conținutul	Nr. Ore
1	Structuri elementare de date Liste (simplu și dublu înlănțuite, deschise și circulare) Cozi și Stive.	2
2	Arbori (binari, binari de căutare, arbori multicăi). Implementarea arborilor de Interclasare. Parcurgerea unui arbore în lățime. Parcurgeri în adâncime pentru arbore binar. Arbori de căutare.	3
3	Construire arbore binar de căutare. Inserare în arbore binar de căutare.	2
4	Algoritmi de sortare clasici. Sortarea prin selecție, sortarea prin inserare, căutarea binară, Heapsort, Mergesort, Quicksort Interclasarea optimă a șirurilor ordonate.	3
5	Metode de rezolvare a problemelor. Backtracking (căutare cu revenire). Greedy (optim local). Cele mai scurte drumuri care pleacă din același punct.	3
6	Implementarea algoritmului lui Dijkstra Algoritmi divide et impera. Căutare binară.	3
7	Sortare prin interclasare Turnurile din Hanoi, Înmulțire matrice, Înmulțire numere întregi mari.	3
8	Programare dinamică. Determinarea celor mai scurte drumuri într-un graf. Înmulțirea înlănțuită a matricelor, Programarea dinamică comparată cu tehnica greedy.	3
9	Grafuri orientate și neorientate. Parcurgerea grafurilor în adâncime. Parcurgerea grafurilor în lățime. Salvarea și restaurarea arborilor binari de căutare. Colecții și mulțimi.	3
10	Tipuri abstracte de date. Tablouri. Clasa tablou. Clasa parametrică tablou<T>Stive, cozi, heap-uri Clasele stiva<T> și coada<T> Clasa heap<T>	4
11	Supraîncărcarea operatorilor. Derivare publică, funcții virtuale. Tablouri inițializate virtual. Structura TablouVI<T>	4
12	Aspecte generale privind analiza și complexitatea unui algoritm; Evaluarea complexității. Exemple de analiză a unor algoritmi. Clase de complexitate.	6
		Total = 42 ore

#### b. Aplicații:

1	Liste simplu înlănțuite.	2
2	Liste circulare simplu înlănțuite	2
3	Liste dublu înlănțuite.	2

4	Arbori. Arbori binari oarecare. Parcurgerea arborilor: preordine, inordine, postordine, lățime. Arbori binari înșiruiți;	3
5	Arbori binari de căutare (BST). Operații caracteristice. Complexitatea operațiilor pe arbori BST.	3
6	Reprezentarea și traversarea grafurilor Aspecte generale. Modalități de reprezentare a grafurilor. Modalități de parcurgere a grafurilor: în adâncime.	3
7	Algoritmi pentru prelucrarea grafurilor Tabele de dispersie.	3
8	Metode generale de elaborare a algoritmilor, Metoda Greedy, Metoda backtracking.	2
9	Metoda "Divide et Impera" Metoda programării dinamice .	3
10	Exemple de programe cu funcții pentru prelucrarea fișierelor.	3
11	Algoritmi fundamentali de sortare.	2
		Total = 28

## 5 EVALUAREA

a) Activitățile evaluate și ponderea fiecăreia (conform Regulamentului studiilor de licență) : 30% - activitatea la laborator (probleme rezolvate la laborator, teme de casă); 30% - parțial (lucrare la laborator, la jumătatea semestrului); 40% - examenul final.

- b) Cerințele minimale pentru promovare
- promovarea laboratorului (teme de casă);
  - obținerea a 50 % din punctajul verificării finale;

c) Calculul notei finale: prin rotunjirea punctajului final, în condițiile punctului b.

(de exemplu: 50-55 puncte-nota 5; 56-63-nota 6; 64-72- nota7; 73- 81- nota 8; 82-90 nota 9; 91- 100- nota 10 )

## 6 REPERE METODOLOGICE (modul de prezentare, materiale, etc.)

Cursuri prezentate pe sliduri, lucrări de laborator, susținute de specialiști cu competență, absolvenți ai Facultății de Automatică și Calculatoare, cu experiență în domeniul programării, precum și în activitatea didactică, care au elaborat lucrări în acest domeniu, pe plan național și internațional.

Laboratorul este dotat cu o rețea de 20 calculatoare ( Pentium 4), dotate cu compilatoare C/C++, sala ECS001.

## 7 BIBLIOGRAFIA

1. Vasile Lungu, Limbajul de programare C, Editura Printech, 224 pg., 2005.

2. Vasile Lungu, Programare C/ C++ , Ed. Printech, 195 pg., 2004.
3. V. Lungu, Gh. Petrescu, "Programare C/ C++. Indrumar laborator", Ed. Printech, 2002.
4. WIRTH, N.: Algorithms + Data Structures = Programs. Prentice- Hall Inc., 1976
5. Florian Moraru. Structuri de date si algoritmi. Ed.Bren, 2003

**SEF DE CATEDRA**

**Prof. Dr .Ing. Ioan Dumitrache**



**TITULAR DE DISCIPLINA**

**Conf. Dr. Ing. Gheorghe Petrescu**

